

**TRACKING AND INDEXING OF HUMAN ACTIONS IN
VIDEO IMAGE SEQUENCES**

GAMHEWAGE CHAMINDA DE SILVA
(B. Sc. (Eng.), M. Eng., Sri Lanka)

**A THESIS SUBMITTED
FOR THE DEGREE OF MASTER OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER
ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE
2003**

Acknowledgment

First of all, I would like to thank my main supervisor, Dr. Liyanage C. de Silva, for his excellent supervision during this thesis. His devotion to my development was more than what can be expected from a supervisor. I am also grateful to my co-supervisor A/P. Surendra Ranganath for his support and comments. I would like to express my gratitude to Dr. Michael J. Lyons, my supervisor during my 5 month internship at ATR Media Information Science Laboratories, Japan. Michael-san was always quick in clarifying my doubts and helping me to solve problems encountered in research work.

I would like to thank to Ms. Serene Oe and Mr. Henry Tan of the Communications Lab for their support during my two-year stay in the lab. They ensured that I could progress with my research smoothly by providing me sufficient laboratory resources.

Many thanks are due to my good friends Chathura, Lay Nwe and Mei Poh. They were always around; sharing their expertise with me, participating in the experiments, and helping me get over the hard times. They made my stay here a memorable one.

Finally, to my parents and family I am grateful for our closeness and for their constant love and support.

Table of Contents

List of Figures	vii
List of Tables	x
Summary	xi
Chapter 1: Introduction	1
1.1 Motivation.....	2
1.1.1 Smart Environments.....	2
1.1.2 Video Annotation.....	3
1.1.3 Effective Human Computer Interaction.....	3
1.1.4 Entertainment and Education.....	3
1.2 Problem Statement and Description.....	4
1.2.1 Layout of the Scene and Camera Positioning.....	4
1.2.2 Input Video Sequences to the System.....	6
1.2.3 Functional Overview of the System.....	7
1.2.4 Outputs Generated by Analyzing the Images	8
1.2.5 Issues Related to the Scene	8
1.3 Contributions.....	9
1.4 Organization of this Thesis	9
Chapter 2: Literature Review	11
2.1 Introduction.....	11
2.2 Image Segmentation.....	11
2.2.1 Background Modelling and Subtraction	11
2.2.2 Segmentation Based on Edges and Contours.....	15
2.2.3 Motion Based Segmentation	16
2.2.4 Region Based Segmentation	17

2.3	Human Detection and Modeling.....	18
2.4	Image Sequence Analysis	19
2.4.1	Tracking of Moving People	19
2.4.2	Action and Body Gesture Recognition	21
2.5	An Overview of Existing Systems.....	26
2.6	Limitations in Existing Systems	27
2.7	Summary	27
Chapter 3: Overview of the System.....		33
3.1	Introduction.....	33
3.2	System Overview	34
3.3	System Design	35
3.3.1	Functional Design	35
3.3.2	Algorithm.....	36
3.4	Scene Context	36
3.5	Extraction of Scene Context	38
3.6	Contents of Scene context.....	38
3.6.1	Background Information.....	38
3.6.2	Region-Specific Information	39
3.6.3	Camera-Specific Information.....	39
3.6.4	Geometric and Scale-Related Information.....	39
Chapter 4: Background Modelling and Foreground Extraction		40
4.1	Introduction.....	40
4.2	Background Initialization and Modeling	40
4.3	Segmentation and Background Adaptation	44

4.3.1	Foreground Segmentation.....	45
4.3.2	Background Adaptation.....	46
Chapter 5: Human Detection and Model Acquisition.....		48
5.1	Introduction.....	48
5.2	Overview of the Algorithm.....	50
5.3	Head-Shoulder Model.....	51
5.4	Human Detection.....	52
5.5	Model Initialization.....	53
5.6	Model Refinement.....	55
Chapter 6: Human Tracking and Indexing of Actions.....		58
6.1	Problems related to tracking.....	58
6.1.1	Tracking with Multiple Cameras.....	58
6.1.2	Dealing with Occlusion.....	58
6.2	Overview of the Tracking Algorithm.....	59
6.2.1	Head-Shoulder Region Extraction.....	60
6.2.2	Similarity Measures for Tracking.....	61
6.3	Recognition of Events.....	64
6.3.1	State Model.....	64
6.4	Detection of Unusual Events and Actions.....	67
6.5	Tracking Persons in the Scene.....	67
6.6	Indexing and Recording Key Frames.....	68
Chapter 7: Results and Discussion.....		71
7.1	Background Modeling and Foreground Extraction.....	71

7.1.1	Methods of Evaluation	71
7.1.2	Subjective Evaluation	74
7.1.3	Quantitative Evaluation	78
7.2	Human Detection and Body model Acquisition	80
7.2.1	Methods of Evaluation	80
7.2.2	Results of Human Detection	82
7.2.3	Subjective Evaluation of Model Acquisition.....	83
7.2.4	Quantitative Evaluation of Model Acquisition.....	84
7.3	Tracking	85
7.3.1	Quantitative Evaluation	85
7.4	Generation of the Index and Key Frames	87
7.4.1	Index of Events and Actions	87
7.4.2	Key Frames	90
7.4.3	Visualization of the path of movements	94
7.4.4	Evaluation of Event Recognition	96
7.5	Discussion	97
7.5.1	Background Modelling and Segmentation.....	98
7.5.2	Human Detection and Modelling.....	99
7.5.3	Tracking and Generation of Results	99
Chapter 8: Conclusion and Future Work.....		101
8.1	Conclusion	101
8.2	Future Directions	103
8.2.1	Incorporating Person Recognition	103
8.2.2	3D Human Body Modeling and Tracking	104
8.2.3	Improving the Recognition Capability.....	104

8.2.4	Facial Expression Recognition	104
Author's Publications		107
References		108
Appendix A: Additional Contributions		124
A.1	Overview	124
A.2	Background	124
A.3	Approach.....	126
A.3.1	Detecting and Tracking the Eyes	127
A.3.2	Detecting and Tracking the Nose Tip	128
A.3.3	Mapping Nose Tip Movement.....	130
A.3.4	Using the Mouth to Click.....	131
A.3.5	Implementation	131
A.4	Performance Evaluation.....	132
A.4.1	The ISO 9241-9 Standard	132
A.4.2	Multi-direction Tapping Task	133
A.4.3	Experimental Procedure.....	133
A.4.4	Results.....	134
A.5	Usability Assessment	136
A.6	Descriptive User Feedback	137
A.7	Hands-Free Text Entry.....	138
A.8	Drawing.....	139
A.9	Conclusion	140
A.10	Future Directions	141
A.11	References.....	141

List of Figures

1.1	System Overview	4
1.2	Layout of the Scene	5
1.3	Views from the Cameras Used	6
3.1	System Overview	34
3.2	System Data Flow Diagram	35
4.1	Background Initialization	41
4.2	Motion Segmentation and Background Adaptation	45
5.1	Silhouettes that Give the Perception of the Presence of Humans	49
5.2	Overview of Human Detection and Modeling	50
5.3	Construction of Head-shoulder Model	51
5.4	Proportions of the Human Body with Respect to the Height of the Head	54
5.5	Initial Human Body Model	55
5.6	Refined Human Body Model	57
6.1	Head-shoulder Region and Its Attributes	61
6.2	Overlapping Bounding Boxes	62
6.3	Histogram Computation and Matching for the Shoulder Region	64
6.4	State Transitions for a Human Detected in the Scene	65
6.5	Visualization of Human Tracking Results	68
7.1	False Positive and False Negative Pixels	72
7.2	Background Initialization	74
7.2	Motion Segmentation	74
7.4	Background Initialization	75
7.5	Motion Segmentation without Adaptation & Selection Map	75

7.6	Adaptive Motion Segmentation	76
7.7	Background Initialization	77
7.8	Adaptive Motion Segmentation	77
7.9	Adaptive Motion Segmentation in Multimodal Regions	77
7.10	Variation of Background Accuracy over Time	79
7.11	Results of Human Detection	83
7.12	Images Used for Subjective Evaluation	84
7.13	Key Frame Showing a Person Entering the Room	90
7.14	Key Frame Showing a Person Leaving the Room	91
7.15	Key Frame Showing a Standing Person	91
7.16	Key Frames Showing a Person Sitting	92
7.17	Key Frame Showing a Person Using a Computer	92
7.18	Key Frame Showing a Person Placing an Object	93
7.19	Key Frame Showing a Person Removing an Object	93
7.20	Key Frame Showing an Unusual Event	94
7.21	Sample Frame from an Image Sequence Showing a Single Person	94
7.22	Visualization of Motion Path for a Single Person	95
7.23	Sample Frame from an Image Sequence Showing Two Persons	95
7.24	Visualization of Motion Path for Two Persons	96
A.1	Schematic of the Face Tracking Interface.	127
A.2	Nose Tip Search Area Relative to the Eyes.	129
A.3	Detecting and Tracking Points Corresponding to Between-the-eyes, Eyes, and Nose Tip	129
A.4	Multi-directional Tapping Task.	133
A.5	Sample Trajectories for the ISO Standard Multi-directional Tapping Task.	135

A.6	Learning Curves.	135
A.7	Average Movement Time (sec) Versus Orientation (deg) for the ISO Standard	136
A.8	Operating the Dasher Text Entry Interface with Head Movements.	139
A.9	A Drawing Created Using Head Movements	140

List of Tables

2.1	A Summary of Research on Segmentation	29
2.2	A Summary of Research on Human Detection, Modeling and Tracking	30
2.3	A Summary of Research on Human Action and Body Gesture Recognition	31
4.1	Contents of the Selection Map	44
5.1	Specification of the Initial Body Model	54
6.1	Set of Rules for State Transitions	66
6.2	Sample Entries of the Scene Index	69
7.1	Results for Non-adaptive Foreground Segmentation	78
7.2	Results for Adaptive Foreground Segmentation	78
7.3	Results of Human Detection	82
7.4	Accuracy of Body Model Acquisition	85
7.5	Results of Evaluation for Human Tracking	86
7.6	Sample Index after Image Sequence Analysis	88
7.7	Accuracy of Action and Event Recognition	97
A.1	Summary of Responses to the Usability Assessment Questionnaire	136

Summary

Video surveillance using recorded video captured by Closed Circuit Television (CCTV) cameras is a common means of increasing the security of a given environment. However, tracing back an incident using such video is a tedious task as the amount of video images to be searched is quite large.

Recent advances in Computer and Video Technology have resulted in the availability of powerful hardware for acquisition and processing of images and video, at a fairly low cost. However, the current state of the art in computer vision algorithms is not mature enough to be used in a system for fully automated surveillance. Hence the approach we have taken is to analyze videos and provide an index which will enable manual search time to be reduced significantly.

If the videos can be analyzed automatically and indexed according to events such as a person entering the scene, the search time can be reduced significantly. The index, which is much shorter than the video sequences, can now be searched for the related events and the appropriate portions of the video can be displayed.

In this thesis, we propose a system that analyses image sequences acquired from a particular scene to detect humans and their actions, such as entering/leaving the scene, walking, using a computer etc, and index the sequence according to these actions. Images from multiple stationary cameras mounted in the scene are used to acquire video image sequences. An innovative approach for background modeling and adaptation is used to identify image features corresponding to humans and foreground objects in the scene. A novel method for human detection is used to detect

humans present in the scene and acquire human model parameters. This method is capable of detecting humans from incomplete views and modeling them accurately. Also, the method is not dependent on the skin color or the size that the humans appear in images. The detected humans are tracked and the recorded model parameters are validated against a set of rules and a state machine to recognize actions and events.

The image sequence is indexed using the results for faster searching. *Key frames* are extracted from the image sequences for each entry in the index, to facilitate visual inspection without browsing the image sequence. In addition to the index, visualizations of motion paths for humans in the scene are created to provide a faster way of tracking human movements in the scene.

Different functional components of the system have been tested using a number of images and image sequences. Both subjective and quantitative evaluations have been carried out, defining measures for evaluation where necessary. According to the test results, the average overall accuracy of the system is 90.5%.

Introduction

Video surveillance using Closed Circuit Television (CCTV) cameras is a common means of increasing the security of a given environment. Often, instead of monitoring the video online, the videos are recorded and archived continuously, so that an incident can be traced back. However, this process of tracing is manual and can be a tedious task. The amount of video information that has to be searched can be extremely large, depending on the number of cameras in the scene and the timescale of tracing. In most cases video information has to be ‘searched’ sequentially to find out when a particular *event* (for example, a person coming in to a particular location of the scene and taking some object away).

Recent advances in Computer and Video Technology have resulted in the availability of powerful hardware for acquisition and processing of images and video, at a fairly low cost. Computer Vision has been a very active area of research during the past couple of decades. If the videos can be analyzed automatically and indexed according to events taking place, the search time can be reduced significantly. The index, which is much shorter than the video sequences, can now be searched for the related events and the appropriate portions of the video can be examined manually. In the above example, the search domain may now be reduced to the number of situations where a person was present in that location, instead of the entire video sequence. In this thesis, we propose a system that analyses image sequences acquired from a particular scene to detect humans and their actions (such as walking, sitting, using a computer etc.), and index the sequence according to these actions.

1.1 Motivation

The work presented in this thesis is motivated by its immediate applications in the area of automated surveillance. By incorporating real-time processing capability to this work, active security and surveillance systems can be developed. Action recognition can play a very important role in active surveillance, serving as the initiating stage of automatic indexing.

The following are some of the other application areas where the outcomes of this research are useful:

1.1.1 Smart Environments

Humans interact with the environment that surrounds them in a number of ways. Living beings, present in an environment, perceive the environmental conditions and act, react or adjust accordingly. If the environment itself can behave in the same way, there can be several advantages. It can facilitate the people in the environment by reacting to their actions, verbal expressions, body gestures and facial expressions. They can be used to give an interpretation of the status of and occurrences in the environment to interested parties outside the room, as in the case of distance learning and surveillance. Such an environment is called a *Smart Environment*. Our work can be directly applied to the task of processing video inputs in a smart environment. The Smart Rooms [1] and the *Kids' Room* [2] designed and implemented by the *MIT Media Lab* [3] are two examples for existing smart environments.

1.1.2 Video Annotation

Automatic annotation of video based on content is one very important application of this work. The amount of digital content that is available to us is very large, and is increasing at a very high rate. Automatic annotation of this large amount of content based on human actions will be very useful for various applications.

1.1.3 Effective Human Computer Interaction

If human body gestures can be robustly recognized by a computer using real-time video, communication with the computer can be enhanced. A computer capable of such recognition can interact more effectively with the user, as reliance on the conventional input devices can be minimized. The *Mouthesizer* interface for vision-based analog input [4] and vision-based cursor control systems using facial gestures [5][6] are some recent research outcomes in this area.

1.1.4 Entertainment and Education

Applications in the entertainment and the education areas are growing very rapidly. Computer vision methods for action and body gesture recognition can have a considerable impact in this area. An intelligent tutor for training body gesture related activities such as aerobics is one prospective application. Computer games can directly use this work. One example is substituting pressure plates used for the recognition of dance steps in dance games [7], by using a vision based interface.

1.2 Problem Statement and Description

The tasks as described in Section 1.1 are quite ambiguous in several ways. The nature and positioning of cameras and the quality of images can vary. Since video input contains a lot of information from the scene, it is necessary to identify clearly what is expected to be detected and identified. This section describes the specific problem that we are trying to solve with respect to these issues.

Figure 1.1 illustrates the basic functionality of the proposed system and how it is interfaced to its inputs and outputs.

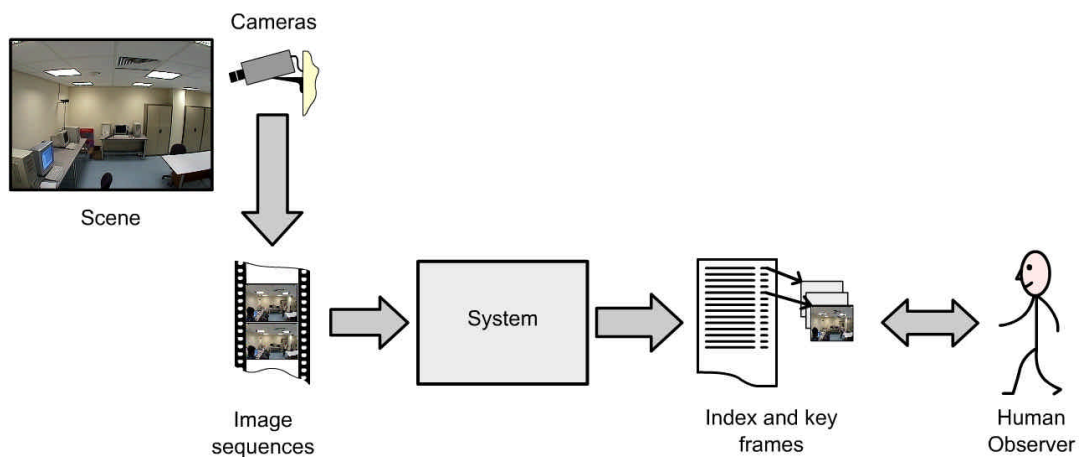


Figure 1.1: System Overview.

The following subsections describe the scene used for acquiring video sequences, the nature and the format of the images, the main functions of the system and its outputs.

1.2.1 Layout of the Scene and Camera Positioning

In this research a room at a research laboratory is selected as the environment for implementing and evaluating the system. This is a closed room with no windows

and has a single entrance. The room is furnished with tables, chairs and computers, objects that can be found in a typical office environment. Three cameras are mounted on the walls, for image acquisition. . Figure 1.2 illustrates the layout of the room.

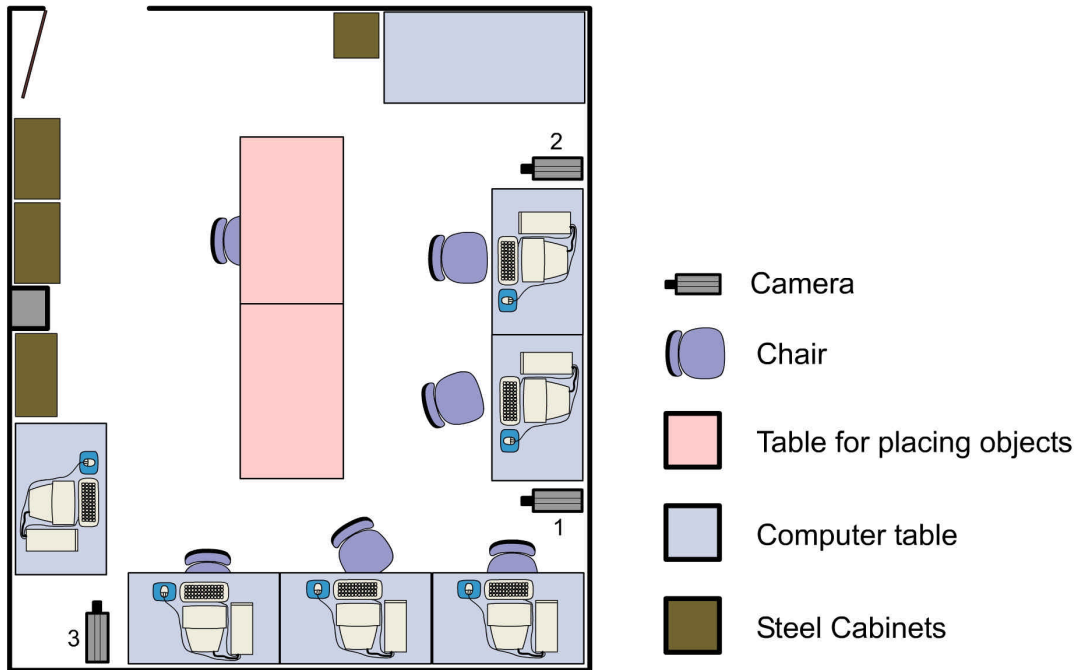


Figure 1.2: Layout of the Scene.

For the work presented in this thesis, only video sequences from cameras 1 and 2 are used. Figure 1.3 shows images acquired using these two cameras. The main reason for this is that camera 3 has been fitted with motorized zoom and focus controls and used for other research projects on face detection and recognition. Therefore it is difficult to readjust this camera and use in this work.



Figure 1.3: Views from the Cameras Used.

However, it should be noted that by selecting this particular location the applicability of the system is not restricted. Our objective is to develop a system that is usable in similar environments without major changes. More details will be discussed in Section 1.2.5.

1.2.2 Input Video Sequences to the System

Two stationary cameras (numbered 1 and 2 in Figure 1.2), mounted on the walls of the room are used to obtain synchronized color images at a maximum rate of 25 frames per second (this can vary according to the CPU load on other processes running on the server handling the cameras). These images have a resolution of 768×576 pixels. The images are in uncompressed RGB format, where 8 bits are used to represent the intensity of each color component. Once acquired, the images are stored in a hard disk for processing.

1.2.3 Functional Overview of the System

The system processes the image sequences offline to perform the following tasks:

- Recognition of known events
 - Detect persons entering/leaving the room (recognition of person identity is out of the scope of this work, but it is possible to incorporate this if it can be obtained by any other means).
 - Introduction/removal of objects to/from a table in the room.
- Track persons in the scene:
 - Location: The terms *Tracking* and *Location* do not refer to the real world coordinates of the room. Rather, an attempt is made to track a person's position in the room in relation to the stationary objects in the scene. In our work, the computers in the room (under the assumption that they are not moved to other locations) are used as a reference for tracking. In addition, a simple visualization of the movements of each person in the room is generated by sketching the path of his/her movement on an image showing only the static background.
- Actions: The term “actions” here refers to the three body gestures, standing, sitting and walking.
- Detect *unusual* events and actions
 - If a large change in the scene, such as one caused by moving the camera, is detected, the scenario is referred to as an unusual event. For such scenarios an index entry is created to facilitate manual observation.

1.2.4 Outputs Generated by Analyzing the Images

An index containing the recognized and detected events and pointing to their locations in the sequences is generated as output. In addition to keeping an index, a *key frame* is extracted from the sequence for each entry in the index. The key frame is an image frame, which is supposed to contain visual information for the particular entry. For example, the key frame for the event of a person entering the room is a frame showing that person entering the room. Providing key frames makes manual checking of the index easier.

1.2.5 Issues Related to the Scene

The scene that we have selected is a room in a research laboratory that is being used by students for their regular work, not a scene specifically restricted for our work. This has a few implications. There can be considerable movements of the objects such as chairs in the room, slight adjustments in the positioning of some objects such as tables, objects like bags, stationery and measuring instruments can be introduced to or removed from tables.

Another set of issues is posed by camera positioning and illumination. In most of the places in the room, a full view of the body of the person cannot be acquired using the cameras. Therefore the human detection and tracking algorithms dependent on a full body view cannot be employed. The room is illuminated with ordinary fluorescent lamps. This suggests that the algorithms used in the early stages of the system have to cope with flicker.

1.3 Contributions

The following summarizes the contributions presented in this thesis:

- An innovative technique for accurate background modeling, segmentation and background adaptation is presented.
- A novel approach for human detection and body model extraction is presented. This approach is different from the existing techniques (described in Section 2.3), as it is based on the minimal amount of information from the head-shoulder region of a human in an image. This facilitates defining an initial body model despite the presence of occlusion, which is subsequently refined to acquire a complete parameterized body model.
- An innovative method of integrating multi-camera data has been developed for accurate tracking using uncalibrated cameras.
- A centralized knowledge base containing *context data* related to the scene has been designed and used for improving the system performance.
- An additional contribution resulting from this research is a Facial Gesture Interface for Vision based cursor control.

1.4 Organization of this Thesis

The remainder of this thesis is organized as follows:

Chapter 2 presents the literature review of related work, in the areas of computer vision, image sequence analysis and object tracking. It also briefly describes some existing systems and ongoing research in this area.

Chapter 3 gives an overview of our approach. A detailed description of the incorporation of the context data related to the scene is included here.

Chapter 4 describes the algorithms that we use for background modelling of the scene and foreground extraction. Foreground extraction is combined with adaptation of the background model to achieve good performance under changes in the background.

Chapter 5 explains how human detection and modelling is performed after extracting foreground. A human head-shoulder model is created and used for detecting human presence in images. After detection, the parameters of a 2 dimensional human body model are extracted for each human detected.

Chapter 6 discusses how the index of events is generated and key frames extracted using the human models detected. The issues in tracking using multiple uncalibrated cameras and in the presence of occlusion are discussed here. The state machine based approach for tracking humans is described subsequently.

Chapter 7 evaluates the results obtained by testing the system with a number of image sequences. Both quantitative and qualitative evaluations are performed on intermediate and final results. A discussion of these results is also included.

Chapter 8 contains the conclusion, and a brief discussion of possible future directions.

Literature Review

2.1 Introduction

The problem addressed in our research is in the area of Computer Vision and sub-topics such as image segmentation, human detection, human/object modeling, and image sequence analysis. The remainder of this chapter contains a review of recent research in these areas. Also included in this chapter is a review of systems with similar functionality.

2.2 Image Segmentation

Foreground segmentation is necessary to identify the regions in the images that correspond to persons/objects appearing in the images. There are several approaches for image segmentation. We review a few approaches that are applicable to our work, in the following sub-sections.

2.2.1 Background Modelling and Subtraction

Since we are using stationary cameras, it is possible to segment foreground by looking for a significant difference in an acquired image from a representation of the static scene (hereafter referred to as background) seen by the particular camera. This technique is also known as *backgrounding* [8].

Backgrounding is a popular technique for foreground extraction due to its simplicity. Assuming that the camera is stationary and the background changes slowly

when compared with the moving foreground, it provides an easy way to segment the foreground. Adaptive background modeling can be used to dynamically remodel the background to ensure good results in the presence of background changes. Backgrounding is versatile in the sense that it is applicable to monochrome images, and color images in different color spaces. Although background subtraction will not produce extremely accurate motion segmentation in all situations, it provides sufficient amount of information to the intermediate levels of the system, where further processing is performed [8].

However, it is desirable to monitor background changes, such as introduction of a new static object. Changes in illumination are possible, due to opening of doors and power fluctuations. The backgrounding algorithm should be robust to these effects. Therefore, it is necessary to use an adaptive backgrounding technique which is capable of robust motion segmentation. At the same time, since motion segmentation is only the initial stage of the system, it is desirable to use computationally less intensive techniques.

The most common method of creating a background model for a scene is averaging a selected image sequence, of a short duration, over time. The background model, in this case, consists of only the resulting average image. The entire sequence should contain only the background, for acceptable results. This process is sometimes referred to as “background initialization”. The difference between a frame and the background image is thresholded using a scalar value to segment moving objects. However, this simple scheme has been found to be inadequate for effective motion segmentation in the presence of illumination changes, flickering in the illumination

source and noise. Therefore, several modifications have been incorporated into this basic method by various researchers.

One popular approach in recent research is to model the background as a texture surface. On a texture surface, each point is associated with a mean color/gray value and a variance. This was first suggested by Pentland et al.[9] and successfully implemented in their *Pfinder* system for tracking a single person in a slowly varying background. The mean and the variance of the pixel values of the background image were used as input to an expectation maximization algorithm for motion segmentation. Modified forms of the above approach were adopted with minor changes by other researchers for similar applications [8][10][11][12][13].

A few other techniques have also been successfully employed for background modeling. Haritaoglu et al. [12], in their *W4* system, use minimum and maximum pixel values in background image frames together with the maximum temporal derivative at each pixel. Utsumi et al. [14] use a distance transform together with the average background. Riddler [15] modeled the background using Kalman filters to handle illumination variations. Stauffer and Grimson [16] suggested that each pixel in a background image can be modeled with a mixture of Gaussians. O'Malley et al. [17] use squared Mahalanobis distance between a pixel value and the corresponding background pixel of segmentation. However, this results in an algorithm that is computationally intensive, making the algorithm less suitable for real-time applications. Khan and Shah [18] use the same calculation, in the YUV color space, with a few simplifications. However, the performance in terms of processing time has not been reported.

There are situations where it is not possible to find images with only the background, for background initialization, e.g. when using video of an expressway to measure vehicular traffic flows. Averaging over time can be applied in such situations, but the results will be poor. De Silva [19] has used statistical mode of pixels in an image sequence for successful modeling of the background in traffic image analysis.

It is not possible to segment motion properly using a static background model as discussed above, if the scene is subject to variations in illumination, and the introduction and removal of static objects. This problem can be eliminated by periodically updating the background model created in initialization. However, it is essential to update the model using only the pixel values corresponding to the background or newly placed static objects, not those corresponding to moving objects. To enable this, *binary support maps* consisting of segmented blobs are used [8][9][10][12][16][20]. A less computationally intensive approach is to refrain from updating the model where the bounding boxes corresponding to moving objects are located [11]. Haritouglu and Flickner [21] use a *decision support map* constructed by studying the temporal variations in the binary support map, to improve segmentation. Background subtraction is usually followed by a couple of morphological operations to group blobs that are in proximity, and remove small blobs. After this, the results of background subtraction are used as input to another stage that identifies motion parameters and performs other functions as necessary. Any errors in background subtraction get passed over to this stage. The parameters of object models can be used to remove the errors and produce accurate outputs.

2.2.2 Segmentation Based on Edges and Contours

Background modeling and subtraction cannot be employed in situations where the cameras are not stationary or the background is changing too fast for the adaptation algorithms to remodel the background properly. Rosenberg and Wermon [22] used image registration with the known background to handle this problem. Alternatively, for such situations, the edges or contours corresponding to the foreground can be used to assist the process of segmentation. Moreover, information about edges can be used to improve the performance of backgrounding [12].

Hyeon et al. [23] segment the upper body regions of humans in images by comparing the edges in the image with a predefined curvature model. Jabri et al. [25] combine an edge model, a background model and a confidence map for improved segmentation for detection and location of people in video images. Sminchisescu [25] uses a combination of intensity edge energy and horizontal flow field of motion boundaries for segmentation for monocular 3D body tracking. *Snakes* (active contours) can also be used for image segmentation. Tabb et al. [24] use active contour models to detect human objects in an image. However, this work needs user initialization. Schoepin and Chalana [26] use snakes to segment non-rigid objects for tracking objects that can be represented with a single closed curve.

2.2.3 Motion Based Segmentation

Another approach to segmentation is to extract image features corresponding to object motion in the scene by using a pair of consecutive image frames of the sequence simultaneously as input. The simplest form of this approach is to calculate the difference between two consecutive image frames. This was first suggested by Jain [94] for image sequence analysis. Lee [28] uses difference images for segmentation of humans in cluttered indoor scenes. This is based on the fact that humans generally make at least a small movement between consecutive image frames whereas stationary foreground objects do not.

Optical flow can be used as a means of segmenting moving objects in image sequences. A survey of the state-of-the-art for the computation of the optical flow can be found in [84]. Optical flow is classically extracted by assuming conservation of intensity between two consecutive frames. This problem, being ill-conditioned, requires regularization techniques [85][86]. Zhao and Nevatia [65] use optical flow for segmentation of moving human limbs for tracking human locomotion. However, the calculation of optical flow is computationally intensive, making it unsuitable for real-time applications constrained by limited processing power.

It is possible to calculate motion parameters by collectively processing a number of image frames corresponding to a known time interval, instead of calculating optical flow between each pair of frames. Motion history and motion energy are two measures derived in this manner. Rosales [90] uses Motion Energy Image (MEI) and Motion History image (MHI) to segment human actions. The work

of Davis and Bobick [91] is similar, except for the fact that they use MHI and MEI to classify a large set of action using low resolution images.

Since recently, there has been a growth in the area of processing MPEG image sequences, due to the advantages achieved by data compression. For motion based segmentation, MPEG sequences present an advantage in the sense that they already contain an encoded representation of motion in the image, in terms of *motion vectors*. Ozer et al. [93][94] use MPEG motion vectors to segment human motion in MPEG sequences.

2.2.4 Region Based Segmentation

Another approach for segmentation is based on information from image regions. Rao and Shah [87][92] use skin detection together with connected component algorithms for segmentation of human hands in images. Most face detection systems are based on this approach. However, skin detection causes problems when background regions have colors within the range of skin tones, resulting in false positives. Moreover, the performance of skin detection algorithms also depends on illumination color. Therefore, additional constraints are often required. Stauffer and Grimson [89] use color similarity of random background patches for segmentation of pedestrians from street images. However, this technique performs well only when the background color distribution is limited, such as that of a road surface, rendering it unsuitable for cluttered backgrounds.

2.3 Human Detection and Modeling

Detecting the presence of a human in a scene and acquiring parameters of a predefined body model are essential steps in many computer vision systems. The applications include intruder detection, human tracking and action recognition. There are numerous research works on these topics. However, most of them are based on some assumptions that make them difficult to be used in typical scenes where human detection and modeling is necessary. For human detection, methods based on face/skin color detection [9][74] can only handle a limited number of head poses [28] and result in false positives in the presence of objects with similar tones. Edge based methods such as [23] are not suitable for cluttered scenes. Contour based methods are computationally intensive making them not suitable for real-time implementation [75]. A common approach is to perform background subtraction on the image and detect foreground blobs as corresponding to humans, based on size, aspect ratio, shape and orientation constraints [76][57]. However, for monocular images, this method fails in the presence of occlusion as the said constraints are based on a complete view of the human body.

After human detection, the image features corresponding to a human in the scene are used to extract parameters for the human body model. There is a wide variety of the models used [63][64]. Some of the existing researches use multiple cameras to obtain a full view of the human body so that the model parameters can be specified completely. Where monocular images are employed, using markers for identifying different parts or joints of the body is common. In most systems based on monocular images, it is assumed that the human body is occluded only by itself.

Marker-less automated human body model acquisition using monocular video in the presence of occlusion is still a challenging task.

There has been a substantial amount of research on human detection and modeling during the past few years. Petkovic et al. [36] have employed the above mentioned features and masks constructed in the shape of a person to identify players in tennis videos. Pentland's *Pfinder* system [9] uses maximum likelihood based on the same features to identify moving persons. Hua et al. [37] use skin color to detect human faces in images. Utsumi et al. [14] employ a cylindrical model of a human to identify humans in images. Ju et al. [66] used a cardboard model to parameterize human limb motion in video. Zhao and Nevatia [13] show that an ellipsoidal 2D model is sufficient for tracking people, but use a 2d skeleton model for tracking human locomotion with higher accuracy [65]. Kakadiaris and Metaxas have developed a system for 3D human body model acquisition [67] and tracking [68] using three cameras placed in a mutually orthogonal configuration. In one of the techniques, the person under observation is requested to perform a set of movements according to a protocol that incrementally reveals the structure of the human body. Once the model has been acquired, the tracking is performed using a physics-based framework [69].

2.4 Image Sequence Analysis

2.4.1 Tracking of Moving People

The most common method used for tracking moving people is to match the blobs in one image frame to the subsequent frames in the sequence. If the frame rate

is reasonably high, or if the movements are slow, *similar* blobs in consecutive frames will be produced by a moving person. By continuously matching the blobs and recording the position information, the person/object can be tracked. The measure of similarity is usually a combination of attributes such as the size, color and position. However, the position is sufficient for matching objects in most cases, as shown by McKenna et al. [8]. Wren uses the constraints on human movement to model human motion, and thereby track moving humans [38]. Histogram matching is the most common approach for tracking humans [50][13][62][21]. Expectation maximization algorithms have been employed for most model-based tracking systems [57][18]. O'Malley et al. use a color model with a mixture of Gaussians for tracking humans in a wide area [17]. Techniques based on Kalman filtering have been used to improve the accuracy of the estimated position [10][13][51][63][64]. An algorithm using products of exponential maps and twist motions to describe the connectedness of body parts and relative motion of parts connected by joints is described by Bregler in [70]. Measurements are based on optical flow. This method assumes that the body model and initial pose are known. However, hand initialization is required, making it unsuitable for most applications. Pavlovic et al. used a Dynamic Bayesian Network models for tracking. Again, the templates for body parts are initialized manually [63].

Deutscher et al. [71][72] develop a system based on the CONDENSATION algorithm [73], which uses non-parametric densities represented by a set of samples for observation process and posterior density. Two image features are used in combination for tracking: edges and foreground silhouettes. Good tracking results are achieved using this approach [63].

Tracking multiple humans is a much more complex task compared to tracking a single human. This is due to occlusions and the increased number of possible matches. Reasoning based on predictions can be used to handle inconsistencies in matching caused by occlusion. Khan and Shah [18] use maximum a-posteriori probability of similarity of colors to handle occlusion implicitly. Zhao and Nevatia [13] explicitly handle occlusion using both the human body models and walking speeds to separate occluded humans. Haritaoglu et al. [21] use temporal segmentation of foreground based on geometry and motion cues to separate foreground segments of multiple people into individuals. This method is more suitable for our work as it can be applied directly on the results of foreground segmentation.

2.4.2 Action and Body Gesture Recognition

There has been a growing amount of research on image sequence analysis for hand/body gesture recognition. There have been two main approaches to which this work can be categorized, namely configuration-based recognition and motion-based recognition.

(a) Configuration-based tracking and recognition

The basic idea in configuration based tracking and recognition is to identify the structural appearance of human body in each image frame and observe how it changes within the sequence of frames during a specified time interval. For instance, if frame k shows a standing person, subsequent frames up to frame $k+m$ shows the person's body gradually reaching a seated posture (where the value of m depends on the frame rate), and frame $k+m+1$ shows the person seated, it can be recognized that the image sequence contained the action "sitting" of a person. There are three main

tasks involved in this approach. The first task is to detect the presence of a human in a given image frame. This includes recognition of the initial posture and fitting the data into a structural model representing a human. The second task is to track the detected human along the sequence of frames. The third task is to recognize the gesture or action using the spatio-temporal data thus gathered. This is a pattern recognition task where the difficulty is governed by the number of gestures to be distinguished, the nature of the data and the amount of noise present in the data.

Several researchers have followed this approach using different techniques to perform each of the above tasks. . Rehg and Kanade [45] used a 27-degree of freedom model of a human hand in their “DigitEyes” system for hand gesture recognition. Hogg [46] and Rohr [47] use a cylindrical model for the full human body for tracking a walking human in natural scenes. Gavrilla and Davis [48] use a full body model with 22 degrees of freedom for tracking human motion. However, this system requires the humans to wear tight fitting body suits with contrasting limb colors. All these techniques involve a 3 dimensional model of the human body. Davis states that a 3 dimensional model is necessary and sufficient for understanding action [49]. Roberts and McKenna [50] map image data to the surface of a 3D body model for tracking highly textured human subjects in a cluttered indoor scene. Zhao, Nevatia and Fengjun [13] use a 3D ellipsoidal model for segmentation and tracking of multiple humans in complex situations. Mikic et al. [51] use multi-camera voxel data to acquire a 3d human body model consisting of cylinders and ellipsoids. Sminchisescu and Triggs [52] extract a 30-joint 3D body model using edge and intensity data using monocular image sequences.

Some researchers have attempted to use only the 2 dimensional appearances of actions for action recognition. Although this has the advantage of being simpler than fitting a 3 dimensional model, it can make recognition more difficult as actions that include the appearance of the total body are not as visually consistent across different people due to natural variations and different clothing [49]. Yamato et al. uses low-level body silhouettes of human actions together with Hidden Markov Models [53]. Akita [54] uses body contours and edges together with some knowledge about the human body structure. Rosales and Sclaroff [55] use silhouettes and trajectory guided recognition for adaptive classification of action. Darrel et al. [56] construct the *visual hull* of a foreground object for human detection, using multiple cameras. The visual hull, being the maximum volume that creates all the given silhouettes of an object, is a 3D entity. However, projections of the visual hull are used for creating the 2D model. Rosales et al. [57] use a 2D model that consists of body joint locations for estimating 3D body pose using multiple, uncalibrated cameras.

(b) Motion-based Recognition

Motion based approaches attempt to characterize the motion itself without referring to the underlying static poses of the body. Two main directions within this approach are treating the entire body region as a single blob-like entity and the tracking of predefined body regions using motion instead of structural features. Polana and Nelson [58], follow a blob-based approach to recognize cyclic walking motions. They use periodicity measures together with a feature vector describing optical flow magnitudes on blobs. Shavit and Jepson [59] model the motion of the person in to that of an ellipsoidal body model. Little and Boyd [60] recognize people walking by analyzing the motion associated with two ellipsoids fitted to the detected

human body. Rao and Shah [61] use spatio-temporal curvature of trajectory to achieve view invariant action recognition. O'Malley et al. [17] employ position based data association based on color models of mixtures of Gaussians to track human activity for wide area surveillance.

(c) Mixed approaches

It has been demonstrated that both configuration based and motion based approaches can be used to recognize human actions and body/hand gestures. Configuration based approaches facilitate recognition of a large number of gestures, but are computationally intensive due to the high amount of processing involved in modeling and matching. Mikic [51][63] and Roberts et al. [50] emphasize the need of computationally efficient implementations of configuration-based algorithms. Improved human body models have been suggested to facilitate more accurate recognition [13][51][52][64]. Motion-based techniques are faster, but motion data may be more difficult to classify compared to structural data fitted to a highly constrained body model. Refinement of available motion data is suggested by Zhao et al. [65].

A mixed approach, where both configuration-based information and motion-based information are used can be suggested as a means of improving recognition. One such approach for view based human activity recognition has been taken by Ben-Arie et al. [62]. They represented activity by a set of pose and velocity vectors, thereby combining both approaches. Recognition was performed by searching within a multi-dimensional hash table containing these vectors. However, processing speed is an additional issue to be dealt with in this approach.

Most of the work deals with off-line image sequences due to the inability to perform the large amount of processing involved in real time. Most of the systems function only on selected sets of image sequences. When the systems used in smart environments are considered, it is evident that the main focus has been hand gesture recognition, not body gesture recognition. Therefore, there is sufficient room for improvement.

In environments where background is cluttered and more than one person is present at a given time, we have to deal with additional problems like partial occlusions of objects. Most of the current research is focused on body gesture recognition using images of a single person [50][51][56][57][61][62]. In cases of monocular images, it has been ensured that only self-occlusion takes place [52]. Even where multiple cameras are used, all cameras see a complete view of the human, not occluded by any object [50][51][56][57][61], other than for the approach by Zhao et al. [13]. Both these are simpler situations compared to a situation with occlusions and multiple humans. Therefore, this can be considered as a good situation to look for a mixed, modified or novel approach.

Recent research has resulted in systems that can recognize body pose in general, and specific body gestures such as standing, sitting, walking, jumping, kneeling etc. [50][56][57][62][63][64]. However, all these systems require the view of the full human body in image sequences.

2.5 An Overview of Existing Systems

Several types of vision-based systems for surveillance and monitoring of closed environments have been described and built over the past 20 years [1][2][3]. *Smart environments* are an immediate application of this work. Alex Pentland's research group at MIT Media Laboratory designed a *smart room* in 1991 [1]. This has evolved from its initial design to its current state of five networked smart rooms in the United States, Japan and the United Kingdom. These rooms use several machines, none more powerful than a personal computer, to identify the following:

- Location of a person in the room
- Identity of each person
- Facial expression
- Hand gestures (in American Sign Language)

The college of computing, Georgia Institute of Technology, has constructed several smart classrooms [27]. These rooms are equipped with multiple data projectors, cameras and active white boards, to facilitate capturing of lectures for later review by students. The classrooms are also equipped with stylus based tablets for the use of the students.

Xerox PARC uses infrared beacons to provide improved user interfaces for smart rooms [30]. A graphical user interface is used to control equipment in the room, with the aid of these beacons.

2.6 Limitations in Existing Systems

The smart rooms, at their current state, can perform accurately only in simple situations. For instance, a situation where there is only a single person in the room. There is a lot of room for improvement, especially in the areas of action/body gesture recognition and recognition of collective human behavior.

Another important issue is the design of a data model for the interpretations obtained from a smart environment. Most of the existing researches look at providing results to be used directly by persons in the room or observers monitoring the results from outside the smart room. Our idea is to store the results in a relational data base to facilitate content based indexing of the acquired image/voice inputs, and also querying the results to obtain relevant information; for example, when a person has been walking in the room during the last ten days.

2.7 Summary

The main research areas and the related work to this thesis, were reviewed in this chapter. We first examined techniques for image segmentation. Approaches based on background modeling and subtraction, edge/contour based segmentation, motion-based segmentation and region-based segmentation were discussed. The other areas reviewed are human detection, human modeling, and image sequence analysis.

The following tables summarize the approaches and techniques reviewed in this chapter. Table 2.1 outlines the techniques used for foreground segmentation. Table 2.2 presents approaches for human detection and tracking in image sequences.

A summary of research work on human action and body gesture recognition is contained in Table 2.3.

Table 2.1: A summary of research on segmentation

Authors	Application	Scene	No. of Cameras	Segmentation technique	Adaptive technique?
Haritaoglu et al. [12]	Real-time Visual surveillance	Outdoor environment	1	Background subtraction	Yes
Pentland et al. [9]	Real-time tracking of human body	Indoor scene	1	Background subtraction	Yes
Utsumi[14]	Tracking multiple humans	Indoor scene	>3	Background subtraction	Yes
de Silva[19]	Traffic image sequence analysis	Road scene	1	Background subtraction	No
Hyeon et al. [23]	Human detection in still images	General	1	Edge-based	Yes
Jabri et al. [25]	Finding people in video images	general	1	Edge-based	Yes
Sminchicescu [64]	3D human body modeling and motion reconstruction	Controlled background	1	Edge-based	No
Tabb et al. [21]	Detecting partial occlusion of humans in video	General	1	Contour-based	No
Lee [28]	Detecting people	Indoor scene	1	Motion based	No
Zhao and Nevatia [65]	3D tracking of human motion	Outdoor scene	1	Motion based	No
Rosalez [90]	Human action recognition	Controlled background	1	Motion based	No
Stauffer and Grimson [89]	Pedestrian detection	Road scene	1	Region based	No

Table 2.2: A summary of research on Human detection, modeling and tracking

Authors	Application	Human model	Incomplete view	Method of Tracking
Utsumi [14]	Tracking multiple humans	Elliptic pillar	-	Matching Center of Gravity
Hyeon et al. [23]	Human detection in still images	Curvature model of upper-body region	Yes	-
Jabri et al. [25]	Finding people in video images	-	No	Matching silhouette features
Sminchicescu [64]	3D human body modeling and motion reconstruction	3D	No	Eigen vector tracking
Tabb et al. [21]	Detecting partial occlusion of humans in video	Spline	No	Neural networks
Lee [28]	Detecting people in images	Polygonal Head shoulder model	-	-
Stauffer and Grimson [89]	Pedestrian detection	Template	No	Matching conditional color models
Pentland et al. [9]	Real-time tracking of human body	Statistical color and shape model	Yes	Contour & shape analysis
Ju et al. [66]	Modeling articulated human motion	2D Rectangular (cardboard)model	No	Corner matching
Mikic [63]	Human body model acquisition and tracking	3D ellipsoidal model	No	Extended Kalman filtering
O'malley et al. [17]	Wide area surveillance	Silhouette features	No	Matching silhouette features
Khan and Shah [18]	Tracking people using video	Silhouette features	No	Matching color information
Haritaoglu et al. [21]	Tracking shopping groups in stores	Silhouette regions	No	Matching color and shape information

Table 2.3: A summary of research on Human action and body gesture recognition

Authors	Application	Scene	Segmentation technique	Human Model	Tracking method	Recognized Gestures	Recognition with incomplete view?
Haritaoglu et al. [12]	Real-time Visual surveillance	Outdoor environment	Background subtraction	2D rectangular (cardboard) model	Matching torso regions in consecutive frames	<ul style="list-style-type: none"> ▪ Walking ▪ Standing 	No
Zhao and Nevatia [65]	3D tracking of human motion	Outdoor scene	Motion based	Motion template	Maximum likelihood estimation of motion parameters	<ul style="list-style-type: none"> ▪ Standing ▪ Walking ▪ Running 	No
Rosalez [90]	Human action recognition	Controlled background	Motion based	None	Maximum likelihood estimation of motion parameters	<ul style="list-style-type: none"> ▪ Crouching-down ▪ Jumping ▪ Arm waving ▪ Kicking ▪ Leaning over ▪ Sitting ▪ walking 	No
Ayers and Shah [31]	Monitoring Human behavior using video	Office	Skin detection and background subtraction	None	Matching color information from silhouette regions	Actions related to the scene, including Entering, leaving, Standing up, etc.	No

After reviewing these research directions, we observed that there is still room for improvement in the existing approaches and algorithms related to our work. In the light of the above survey, we propose our approach for solving the problem defined in Section 1.2, in Chapter 3.

Overview of the System

3.1 Introduction

The objective of this research is to detect humans, track them, recognize their actions and index image sequences based on this information. In order to achieve improved performance compared to those described in the previous chapter, we suggest a context-based approach. Here, we use the scene context (knowledge about the scene) extensively to perform this task. We divide the main task of achieving the objective into a number of sub tasks that depend on the same scene context. A mixture of the approaches described in Chapter 2 will be used in these sub tasks, while employing novel or modified techniques where necessary.

The remaining sections of this chapter describe design of the system briefly, while subsequent chapters provide a detailed description of the main functional components.

3.2 System Overview

Figure 3.1 outlines the functionality of the proposed system, showing its inputs and outputs.

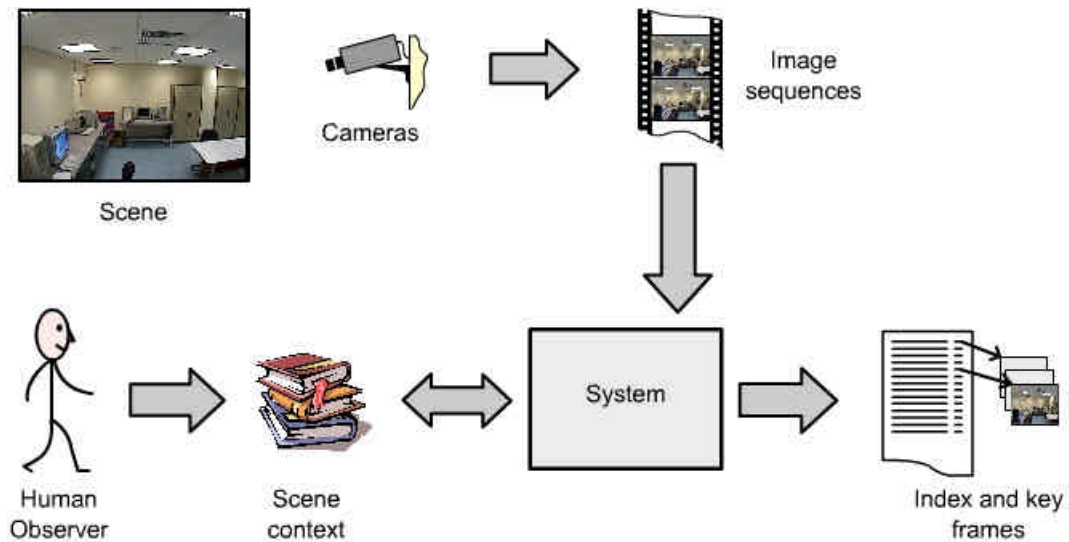


Figure 3.1: System Overview.

Image sequences acquired using two wall-mounted cameras are the main input to the system. The scene context contains knowledge about the scene, constructed both from human observers and the system itself. This acts as an auxiliary input. The output of the system consists of an index to the image sequences, and key frames extracted from the same. The specifications of the inputs, system and outputs are as stated in Section 1.2. A detailed specification of the scene context is found in Section 3.4.

3.3 System Design

3.3.1 Functional Design

The system consists of 5 functional modules performing separate tasks. Figure 3.2 illustrates the functional model of the system according to *Yourdon* notation for functional modeling [83].

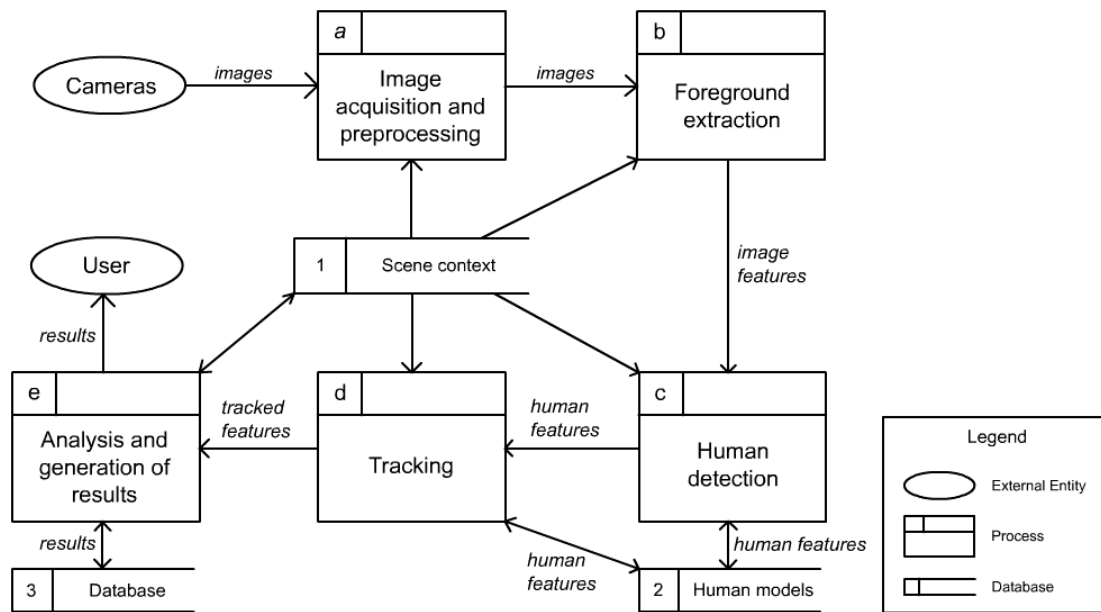


Figure 3.2: System Data Flow Diagram.

3.3.2 Algorithm

We use the following algorithm to obtain the desired results using the image sequences:

1. **Foreground extraction:**

Background is subtracted from the images to obtain binary images containing only the features corresponding to foreground objects and humans.

2. **Human detection and modeling:**

Humans present in the images are detected and model parameters corresponding to each human are acquired.

3. **Tracking and generation of results:**

Using the information from image sequences from each camera, humans are tracked. After tracking, information from multiple cameras is integrated. The results are then analyzed to generate the index and acquire the key frames from the sequence.

The main steps in the above algorithm are described in detail in Chapters 4, 5 and 6.

3.4 Scene Context

Images captured from the environment specified in Section 1.2 provide only a limited amount of information, namely visual information, to the system. For example, a lot of important information such as depth information is lost in the process of creating an image.

However, a human looking at a particular scene can make quite correct interpretations. The reason is that the interpretations are not solely based on visual information. A human combines his knowledge about the scene and the objects in the scene to the visual information, and thereby completes the partial information. This phenomenon is known as visual completion. A simple example of visual completion is the ability to identify a chair that is almost completely occluded by a table, by simply seeing its top part above the table.

Since a smart environment is not a scene where arbitrary changes and events take place, it is possible to provide a lot of information about the room, its contents and the possible events. Such knowledge is referred to as *scene context* in literature related to smart environment research [31][96]. We propose the use of scene context to facilitate more accurate body gesture recognition in a smart environment.

Most of the existing systems make use of scene context by incorporating it into the algorithms. However, there is a disadvantage in this approach. If such specific information is hard coded in the algorithms, a system that works for a particular environment will be unusable or difficult to customize when it is used in a different environment or changes have been made to the environment.

Our idea is to store the scene context in a centralized knowledge base. Some of the contents of the knowledge base can be manually created, whereas others can be created automatically. This knowledge base is centralized in the sense that all components of the system use the information in the knowledge base for their tasks, instead of embedding scene context in the components. Thus, the system can be

configured to be used in a new or changed environment by simply modifying the knowledge base accordingly. The following sub sections describe the information stored in the knowledge base.

3.5 Extraction of Scene Context

Some of the information in the scene context can be extracted automatically. For example, probability distributions of the pixel intensities of the background can be constructed while modelling the background. Another approach to extract scene context is unsupervised learning. An example is training a system with several image sequences showing people entering a room, so that the system can extract information about the image regions corresponding to entrances of the scene. However, in most cases, it is simpler to provide these parameters manually.

3.6 Contents of Scene context

Different types of information are combined from the acquired images to form the scene context. The following sub-sections briefly describe the content of the scene context that we use.

3.6.1 Background Information

This information is useful for performing motion segmentation on the image frames in the captured image sequences. A detailed description about these items is presented in Chapter 4.

3.6.2 Region-Specific Information

The information from particular regions can be more significant than those from other regions, when it comes to the detection of particular events. These are regions where the system has to perform processing in order to detect entry or leaving of a person. By specifying these regions, processing for them in the entire image can be avoided.

3.6.3 Camera-Specific Information

Since we are designing a multi-camera system with overlapping views, the parameters for different cameras may not be the same. The image resolution and frame rate are stored for each camera. Since the cameras are overlapping, the region of overlap is specified and stored in the scene context to facilitate accurate analysis of human features. This will be discussed in detail in Chapter 6.

3.6.4 Geometric and Scale-Related Information

Although we are not planning to calibrate the cameras in the smart room, it is useful to store some information regarding the relation between image sizes and actual object sizes. For example, size information related to humans as seen by each camera can be used to discard small blobs in human detection, as described later in Chapter 5.

Background Modelling and Foreground Extraction

4.1 Introduction

The first step in our approach is to detect image regions that correspond to foreground objects and humans in the scene using background subtraction. Our objective is to design a technique for background modeling and adaptation that is suitable mainly for an indoor scene as specified in Section 1.2. Scene context is incorporated into the technique to achieve better performance.

Our technique consists of two phases. Background initialization is performed in the first phase. In the second phase, both object segmentation and background model adaptation take place. These two phases, and experimental results, are described separately in the following sections.

4.2 Background Initialization and Modeling

In this phase, an initial background model is created by analyzing an image sequence with no foreground objects. An outline of the background initialization phase is illustrated in Figure 4.1.

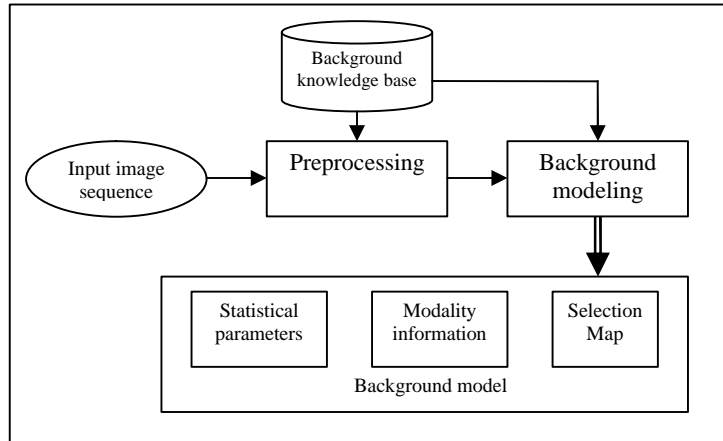


Figure 4.1: Background initialization.

The input video sequence is at least 6 seconds in duration, acquired at 10 frames per second. This number of frames and duration has been selected so that there are sufficient frames for estimation of the parameters of the background model accurately, and also conforms to the number of frames used in similar work [9].

The objective of preprocessing is to perform basic tasks like noise reduction and color space conversion. We employ Gaussian filtering for noise reduction, and convert to the 24-bit RGB color space if the images are not in this format already. This choice of color space has been made mainly for simplicity. The algorithms are applicable for other color spaces with only minor changes.

The knowledge base, which is part of the scene context, contains additional knowledge about the scene that can be utilized in both background modeling and adaptation. Most of the existing systems attempt to improve performance by hard coding such knowledge into the algorithms. Presenting both the knowledge and the images as an input enables the system to be optimized while preserving flexibility to adapt for different scenes. Our knowledge base consists of the following information:

(1) Regions in the image where specific events take place:

Since the events taking place in the scene are usually not arbitrary, the process of image analysis need not seek for the events in an arbitrary manner. For example, if a person enters a room that is currently empty, the entry can be detected by examining only the regions in each frame corresponding to the neighborhoods of entrances.

(2) Expected pattern of modality of the background model:

For a background region that is not changing, the corresponding pixels in the background image show a Gaussian normal distribution over time, due to the presence of flickering of illumination and camera noise etc. But there can be other regions with bimodal distributions, for instance a region corresponding to a window that can be open or closed. In general, there may be regions in the background image where pixels exhibit a multimodal distribution.

(3) Regions in the image corresponding to locations in the scene where foreground objects cannot be present due to physical constraints:

These regions can be disregarded in the processes of background initialization and motion segmentation, to save processing time and system resources.

Let the image sequence in consideration be I where I consist of 3 components $R(x, y, t)$, $G(x, y, t)$ and $B(x, y, t)$ corresponding to the red, green and blue components of the image sequence respectively and

$$x = 1, 2 \dots w$$

$$y = 1, 2 \dots h$$

$$t = 1, 2 \dots N,$$

where

$$w = \text{image width}$$

$$h = \text{image height}$$

$$N = \text{number of frames in the sequence}$$

The background model consists of the following parameters:

(1) Mean of the pixel values, $\mathbf{m}(x, y)$ given by

$$\mathbf{m}_R(x, y) = \frac{\sum_t R(x, y, t)}{N} \quad (4.1)$$

$$\mathbf{m}_G(x, y) = \frac{\sum_t G(x, y, t)}{N} \quad (4.2)$$

$$\mathbf{m}_B(x, y) = \frac{\sum_t B(x, y, t)}{N} \quad (4.3)$$

(2) Standard deviation of the pixel values, $\mathbf{s}(x, y)$ given by

$$\mathbf{s}_R(x, y) = \frac{\sqrt{\sum_t \{R(x, y, t) - \mathbf{m}_R(x, y)\}^2}}{N} \quad (4.4)$$

$$\mathbf{s}_G(x, y) = \frac{\sqrt{\sum_t \{G(x, y, t) - \mathbf{m}_G(x, y)\}^2}}{N} \quad (4.5)$$

$$s_B(x, y) = \sqrt{\frac{\sum_t \{B(x, y, t) - m_B(x, y)\}^2}{N}} \quad (4.6)$$

(3) Statistical mode of the $L(x, y, t)$, brightness component of the pixel values, defined by

$$L(x, y, t) = \frac{R(x, y, t) + G(x, y, t) + B(x, y, t)}{3} \quad (4.7)$$

(4) Selection map of the background, $S(x, y)$.

The following table describes the encoding of $S(x, y)$ according to the properties of pixel (x, y) :

Table 4.1: Contents of the Selection Map.

$S(x, y)$	Description
0	Not part of background model
1	Uni-modal distribution with low variance
2	Multimodal distribution

Depending on the problem and the scene, more information can be encoded in $S(x, y)$.

4.3 Segmentation and Background Adaptation

In this phase, the background model created in the previous phase is used for segmenting the foreground. At the same time, the background model is updated to achieve accurate segmentation in the presence of slow changes in the background. An outline of this phase is shown in Figure 4.2.

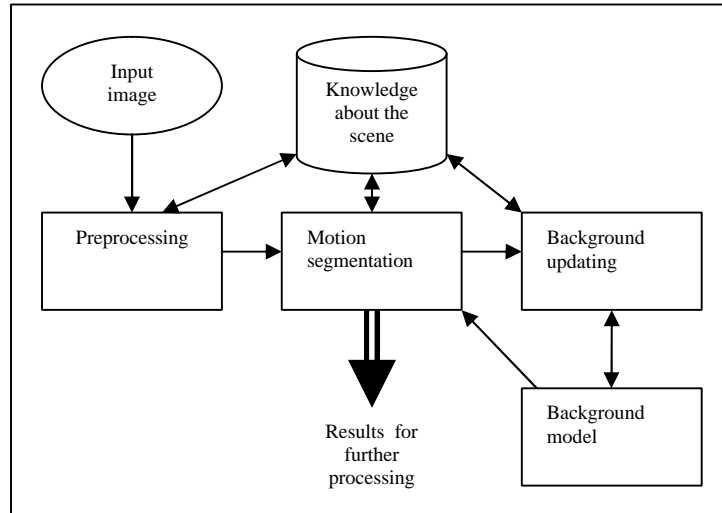


Figure 4.2: Foreground Segmentation and Background Adaptation.

4.3.1 Foreground Segmentation

The process of motion segmentation is outlined briefly, to demonstrate how it makes use of the background model. From each color component of the current frame, the mean background component is subtracted. The magnitude of the resulting difference image is then thresholded with 3 times the standard deviation of that component. This allows different thresholds for different pixels. This results in three separate binary images, corresponding to R, G and B channels of the input frame, which are then combined into a single binary image. For the pixels with a uni-modal distribution and low variance, a logical OR operation is adequate. But for pixels with high variance and multimodal distributions, better results can be obtained by looking for presence of foreground in at least two images.

A series of morphological operations are employed to clean the isolated noise pixels and to fill small holes within blobs. Then the information in the selection map

can be utilized to set pixels in the regions that need not be segmented to zero. Labeling the binary image now can yield the blobs corresponding to objects.

4.3.2 Background Adaptation

Unlike background modeling, background adaptation takes place when the system is actually running and performing its tasks. These include calculation of motion parameters, recognition of moving objects/persons, recognition of actions/events etc. These tasks are computationally intensive, and tend to take a substantial amount of processing time and system resources. Therefore, the background adaptation process has to be sufficiently lightweight, not involving a large amount of processing and storage. However, at the same time, it should either improve or at least maintain performance under varying background conditions. For these reasons, the results of segmentation have been used as the primary inputs for this phase. Since the intermediate results of segmentation are available without any additional computations, they are also utilized.

In the current system, background update is performed only on $\mu(x,y)$ and $s(x,y)$. However, it is possible to extend this to update the selection map $S(x,y)$ as well, if necessary. The updates are not performed on regions that are not subject to background modeling (as specified in the selection map) and the regions covered by the segmented blobs. For the other pixels, the selection of update algorithm is based on the entries in the selection map. At present, the system does not update the pixels with a multimodal distribution. For the others, the update formulae are

$$\mu'(x,y) = af(x,y) + (1 - a) \mu(x,y) \quad (4.8)$$

$$s'(x,y) = \sqrt{\beta \{f(x,y) - \mathbf{m}(x,y)\}^2 + (1-\mathbf{b})s^2(x,y)} \quad (4.9)$$

where

$\mu'(x,y)$ = mean of the pixel values after updating

$s'(x,y)$ = Standard deviation of the pixel values after updating

The values of a and β are small, and in the range 0 to 1. These formulae are widely used in updating background models [8][9][11].

The system was tested on a number of image sequences. The sequences were selected to contain different combinations of inconsistencies that call for dynamic background adaptation. Chapter 7 contains a detailed description of test image sequences and the results.

Human Detection and model Acquisition

After segmentation, we have a binary image with blobs corresponding to foreground objects and humans. In addition to these, it is possible that there are some blobs corresponding to scene features that we are not interested in. For example, a chair that has been moved can create a blob of considerable size, depending on its position in relation to the camera. An additional problem that has to be tackled is the presence of occlusion. A human in the scene can be partially occluded by stationary objects such as tables. Therefore, at this stage, it is necessary to use a technique for accurate human detection under these conditions and for modeling detected humans so that they can be tracked in subsequent frames. The remainder of this chapter describes the algorithms that we use for human detection and modeling.

5.1 Introduction

Our objective in this research is to detect humans and acquire body model parameters despite the presence of occlusion and absence of predefined markers, using monocular images. To achieve this we attempt to use the minimum possible amount of image features for human detection. The neck is the region that has the least diameter when this region is considered, whereas the shoulders form the broadest region in the human body. This variation of breadth can be used for human detection with ease. Because of this variation in breadth, the silhouette of the head and shoulder region of a human has sufficient features so as to be recognized by the human eye, as illustrated in Figure 5.1. Moreover, these portions of the body are less

likely to be occluded by objects placed on the floor, assuming that the cameras are located at or above the eye level. This assumption is valid and realistic for most practical situations. We employ a predefined model of human head and shoulder silhouettes for view invariant human detection. This is similar to the idea used in [28]. However, rather than using a simple model merely for detecting the presence of a human, we use a more general and advanced model that enables us to extract the height of the head (in pixels) and the angle of the human body with respect to the image plane, results that are useful for accurate human body model acquisition.



Figure 5.1: Silhouettes that Give the Perception of the Presence of Humans.

Instead of acquiring parameters of a complex geometric model of a human, we use a two-stage approach. Artists have long observed that the proportions of a human body can be specified completely to a high accuracy using the height of the human head [77][78][79]. We use this property to define an initial body model. Thereafter this model is refined using the image features and geometric constraints. Where portions of the body are found to be occluded, parameters of the initial body model are used to determine parameters of the final model.

5.2 Overview of the Algorithm

Figure 5.2 illustrates an overview of our approach for human detection and modeling. First, the images are segmented by subtracting the background, leaving only the humans and objects in the scene. Human detection is performed on the resulting foreground based on the head-shoulder model, and a coarse initial model is created for each human detected. This model is then refined to achieve a complete human body model. The following subsections describe each of these functions and models in detail.

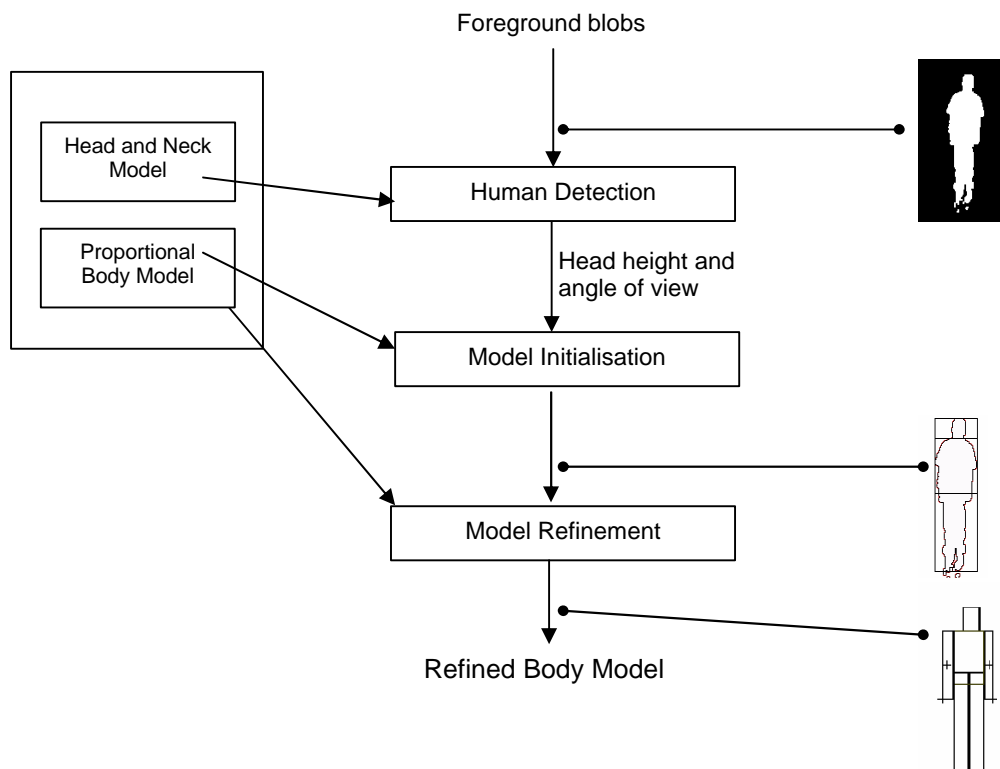


Figure 5.2: Overview of Human Detection and Modeling.

5.3 Head-Shoulder Model

In order to achieve scale independent human detection, we have constructed a head-shoulder model using 300 head-shoulder images. Figure 5.3 shows an overview of the construction of this model. Since the head-shoulder region is seen in different shapes from different angles with respect to the direction the human is facing, the images were categorized and averaged to form 3 different templates, T_1 , T_2 and T_3 . The angles corresponding to the different templates are marked in Figure 5.3. Projections P_1 , P_2 and P_3 were then created by projecting T_1 , T_2 and T_3 , respectively onto the vertical axis. These projections were normalized by obtaining samples by dividing the projection into 100 equal-sized intervals along the vertical axis. Amplitude of the projections was normalized by dividing by the maximum value of the projection.

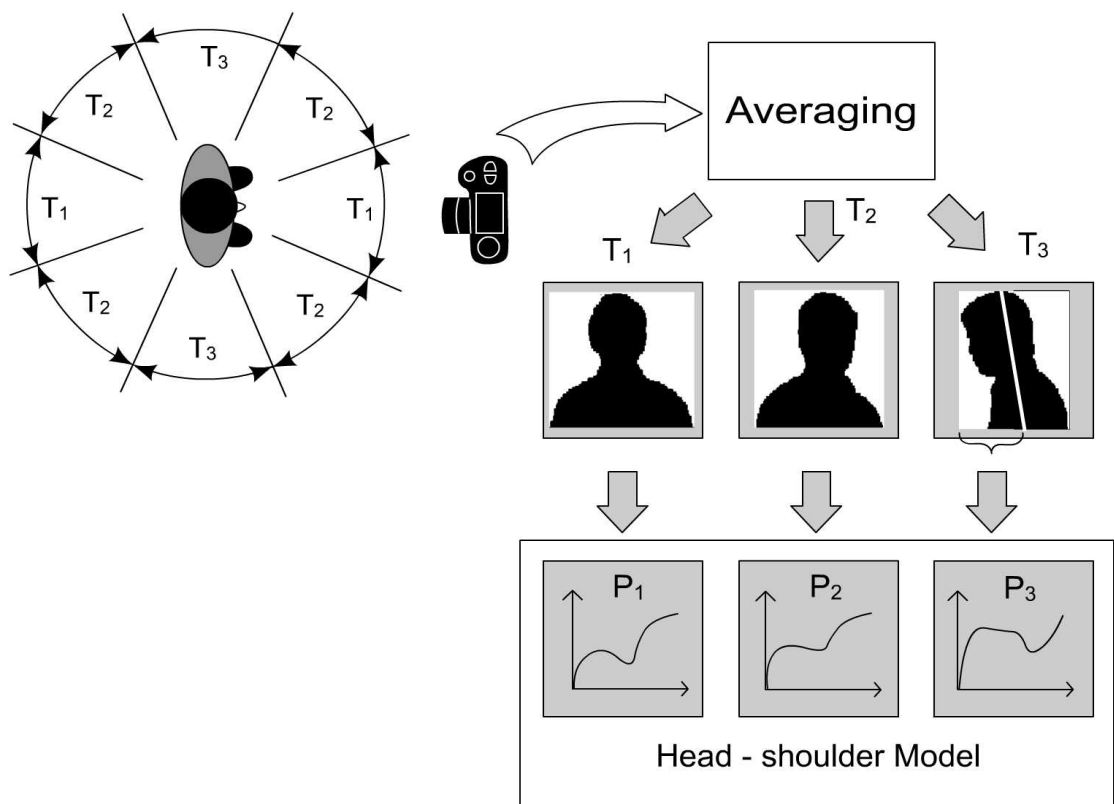


Figure 5.3: Construction of Head-Shoulder Model.

It was observed that the P_1 and P_2 have a distinct shape that can be used for human detection. However, it was evident that the variation in P_3 was not as prominent as in the other two templates. Moreover, different hairstyles can make the silhouette wider in the neck region and make detection less accurate. Therefore we have made a refinement when P_3 is created. The portion corresponding to the back of the head –shoulder region of T_3 is removed before projection. It is evident that a more prominent variation is present in the new template P_3 after this modification.

5.4 Human Detection

After segmentation using the algorithm described in Chapter 4, we project each blob onto the vertical axis (under the assumption that the human is not stooping forward substantially). The next step is to match this projection with the three projections in the head-shoulder model. For this it is necessary to use a matching algorithm that is invariant to scale as both the size of the blob and the degree of occlusion can vary. Therefore, we use the following approach.

Since all three projections in the head-shoulder model contain a sharp local minimum, we look for local minima along the projection of the blob from top to bottom. When we find a local minimum, we obtain three sub-projections by sampling the blob projection to the same dimension of the model projections such that for each sub-projection, the local minimum in the blob projection coincides with the global minimum in the corresponding model projection. Three sub-projections are required as P_1 , P_2 and P_3 have local minima at different positions. These sub-projections are now matched with the P_1 , P_2 and P_3 to identify a strong match (90% normalized correlation). However, for matching with P_3 , the blob is split along the vertical axis

and two sub-projections have to be created as the person appearing in the image can be looking either left or right. If there is a strong match, a human is detected. The height of the human head in the image (in pixels) can be determined using the position of the local minimum that results in the match. By finding the best matching template, we can get a rough idea of the angle of view.

At this point, it is possible to validate the result with scene context to avoid false detections. Since the room is a closed space, the humans seen in images cannot appear arbitrarily small. The minimum possible head height in pixels, for each camera, is stored in the scene context. The detected head height can be validated against this before further processing.

5.5 Model Initialization

The results of matching are the human head height and the angle of view. The basis of the body model acquisition is that the proportions of a human body can be specified to a high degree of accuracy using the height of the head. Figure 5.4 illustrates the proportional model that we are using. This model, referred to as “*the eight-head model*”, is widely used in life drawing by artists [82].

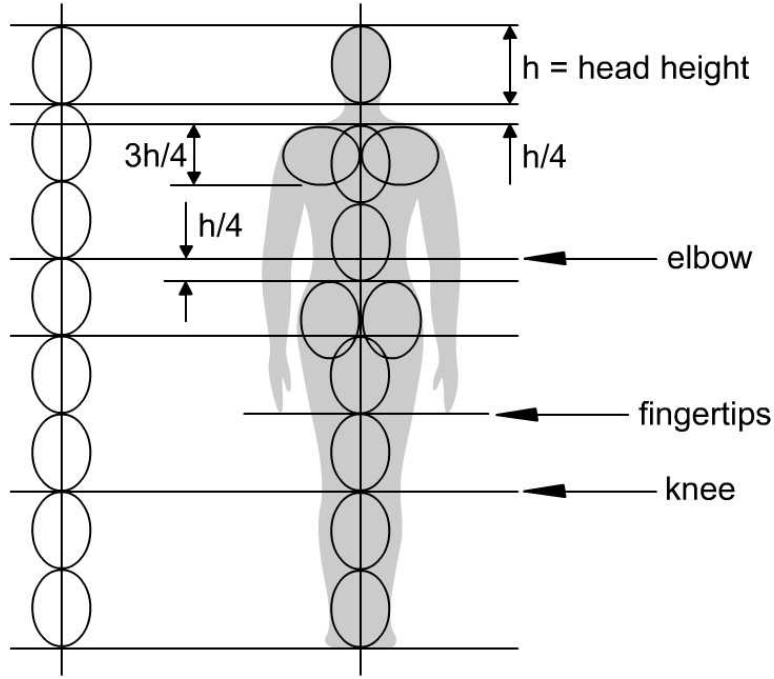


Figure 5.4: Proportions of the Human Body With Respect to the Height of the Head.

However, this is not a universal model. The overall height of a human can vary between 7-8 heads, depending on the gender, age and ethnic group [78]. Moreover, any small error in head height estimate can result in a larger error in estimating the full body. Therefore, only an initial model can be created at this stage and it needs to be refined subsequently. Because of this, we create an initial model with the blob and the bounding rectangles for 3 body regions. For a blob B , bounded by the top-left and bottom-right coordinates (T_{Left}^B, T_{Top}^B) and $(T_{Right}^B, T_{Bottom}^B)$ respectively on the X - Y plane, and a head-height of h , these body regions are specified as shown in Table 5.1.

Table 5.1: Specification of the Initial Body Model.

Rectangle	Left	Top	Width	Height
Head	T_{Left}^B	T_{Top}^B	$T_{Right}^B - T_{Left}^B$	h
Torso	T_{Left}^B	$T_{Top}^B + h$	$T_{Right}^B - T_{Left}^B$	$3 h$
Legs	T_{Left}^B	$T_{Top}^B + 4h$	$T_{Right}^B - T_{Left}^B$	$4 h$

This model, superimposed on an example blob, is shown in Figure 5.5. It is evident that only the upper part of the leg region of the initial model has pixel information due to occlusion. Also, it should be noted that some regions of the initial model can be out of the pixel bounds of the image. The next step is to refine the model and acquire a full body model, while tackling with such situations.

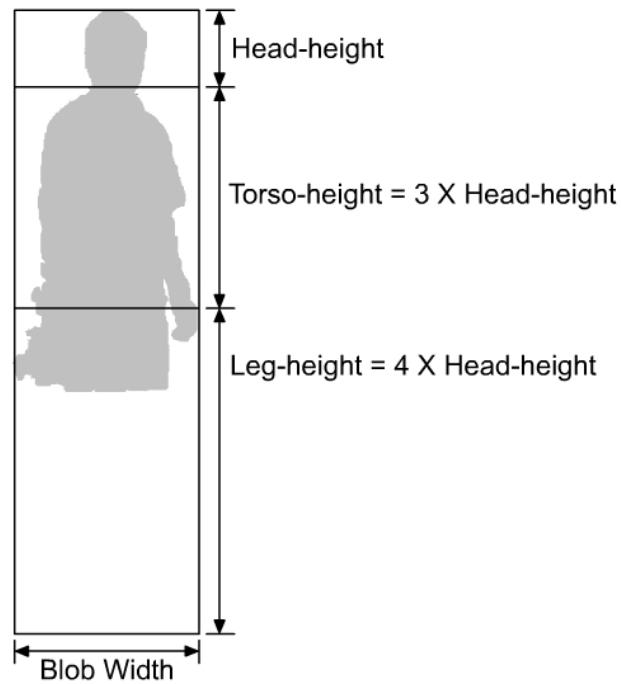


Figure 5.5: Initial Human Body Model.

5.6 Model Refinement

This stage has two main functions. Firstly, it deals with errors in approximating the initial model caused by inaccurate segmentation. Secondly, it uses the proportional model together with the available blob features to acquire the parameters of the complete body model.

The initial model contains only the height and position details of each region. First, the width and centroid of the head are calculated. The height of the head is refined by validating it with the aspect ratio of a human head when viewed in different angles. Refinement of the head region is relatively simple as it is assumed that the entire head region is present in the image, for a successful detection. However, this may or may not be the case for other regions.

For the torso region, the width is calculated using the head height, as shown in Figure 5.4. The axis of the torso region is calculated by joining the centroids of the head region and the upper part of the torso region.

The arms are modeled by removing the refined torso region from the torso region of the initial model. Length constraints of the arms, as imposed by the proportional model, are combined with the pixel information of the arm region to identify the elbow and forearm. The width of an arm region is calculated using the dimensions of the initial model. If a region corresponding to an arm is not found or is narrower than half of the head-height, it is assumed that the arm is occluded by the torso. The position of the arm is assumed to be straight and lowered.

The refinement of the leg regions is similar, other than for the position and length constraints. Possible shadows segmented as foreground can be eliminated during this step.

After the refinements, the final model can be as in Figure 5.6. If the blob is not covering the torso and leg regions, it is assumed that the parts of the body are occluded.

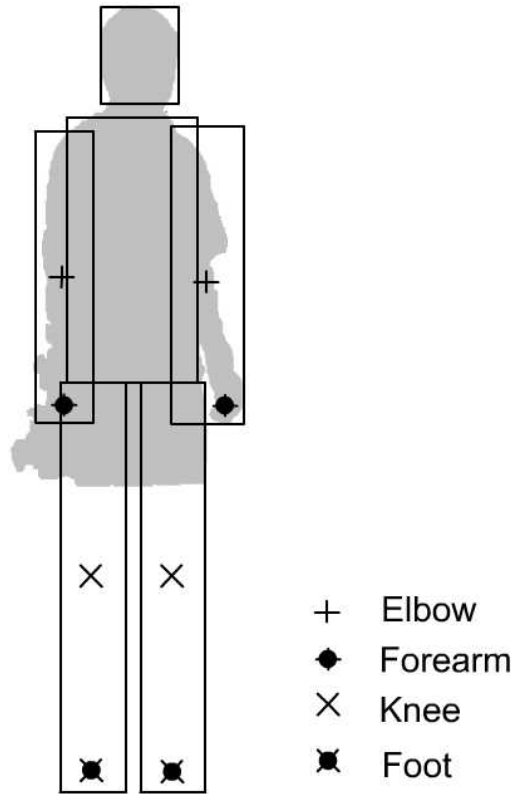


Figure 5.6: Refined Human Body Model.

The ability to detect humans in images was tested by using several images with a number of human subjects appearing in them, both alone and together. Subjects with different hairstyles and attire were selected to identify weaknesses in detection. The results of testing are presented in Chapter 7.

Human Tracking and Indexing of Actions

After human detection and modeling, a set of instances of human models is available. The next step is to keep track of these model instances along the frame sequences and create an index of the selected actions. The remaining sections of this chapter discuss how we perform this task.

6.1 Problems related to tracking

6.1.1 Tracking with Multiple Cameras

The room is equipped with two cameras, having an overlapped view. A person inside the room can be seen in different scales and positions in the images acquired by the two cameras. Since the cameras are not calibrated, and more than one person can occupy the room at the same time, it is important that accurate tracking has to be performed using the information from the images themselves, to prevent inconsistencies such as tracking one person as two persons.

6.1.2 Dealing with Occlusion

There are two types of occlusions that have to be dealt with in human tracking in the selected environment. As a person moves within the room, parts of his body may be occluded by the objects in the room. An example is the occlusion by the table in the middle of the room. The other possible type of occlusion is when multiple humans occlude each other. Since multiple cameras are used in the scene, it is

possible that at least one camera can provide an unoccluded view of each human. However, this cannot be guaranteed in most practical situations.

6.2 Overview of the Tracking Algorithm

The following is an outline of the tracking algorithm used in our work:

1. Start with empty tracking database
2. For each human model acquired from the current image:
 - a. Extract head-shoulder region
 - b. Acquire the following attributes of this region
 - i. Head height
 - ii. Bounding rectangle
 - iii. Centroid
 - iv. Color histogram of the shoulder region
3. For each human model in the tracking database
 - a. Match with the attributes obtained in step 2
 - b. Record the degree of matching for each attribute
4. Find the best match and record the current set of attributes
5. For any human models not matched to the models in the tracking database,
 - a. Add a new model to the tracking database
6. Repeat steps 2-6 until the last image frame has been visited.

The following sub-sections will describe each of the main steps stated above in detail.

6.2.1 Head-Shoulder Region Extraction

We use only the head-shoulder region of the complete body model for tracking, due to a few reasons. Firstly, this region is least likely to be occluded by static objects in the background, such as tables. Secondly, changes of attributes due to limb motion are relatively low for this region. For example, the attributes of the bounding box of a full human body will change drastically with limb motion such as walking. An additional advantage is that chances of this region containing objects carried by a person are also low for the type of office environment we consider.

The head-shoulder region is extracted from the human model using head height to calculate sizes and positions as specified in Chapter 5. The region is resampled to a resolution of 100×100 , to achieve scale invariance. The following attributes for this region are extracted and/or calculated:

1. Head height, in pixels
2. Centroid of the region
3. Bounding box of the region
4. Histogram of the normalized shoulder region.

Figure 6.1 illustrates the head-shoulder region together with these attributes.

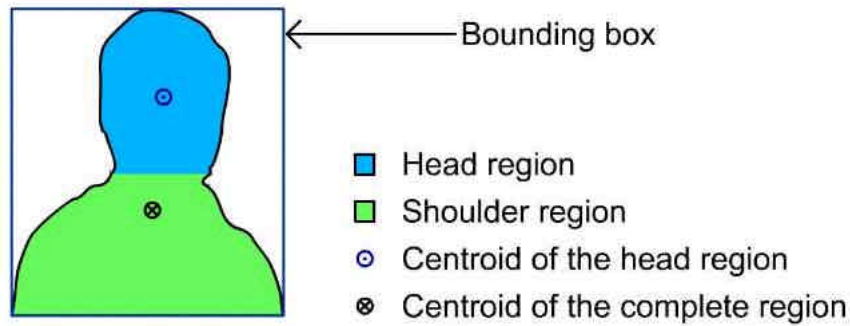


Figure 6.1: Head-Shoulder Region and its Attributes.

6.2.2 Similarity Measures for Tracking

Tracking a human/object between a pair of frames is performed by finding corresponding features in the frames. The process can be repeated to track within a sequence of images. The following measures are used to evaluate similarity between the features that we have selected:

1. Overlapping bounding boxes:

Given that the duration between image frames is close to 25 frames per second, and the normal speeds of human movement within a closed environment, the bounding boxes corresponding to the head-shoulder regions of the same human in two consecutive frames should have a considerable overlap. This is illustrated in Figure 6.2. For our work, we consider above 75% of the area of the bounding box from the earlier frame as the threshold for matching.

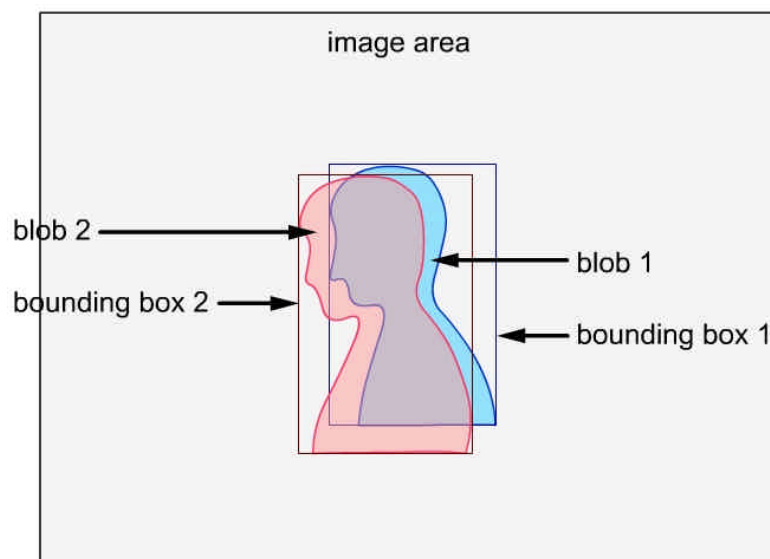


Figure 6.2: Overlapping Bounding Boxes.

2. Nearly equal head height

The head heights for head-shoulder regions corresponding to the same human in two consecutive frames may not be the same, as the human can be moving either towards or away from the camera. The presence of noise also can contribute to some difference. However, the difference, in any of these cases, cannot be significant owing to the high frame rate. We consider a difference less than 10% as the threshold for matching.

3. Head centroids located closely

This condition can be used as an alternative to the first condition, due to its simplicity of computation. However, the weakness in this approach is that the distance between centroids becomes larger when the humans are closer to the camera. The first method provides automatic normalisation for distance, as the percentage of overlap area is independent of blob size.

4. Similar histograms for the shoulder region

The previous conditions can be used for tracking only when the two consecutive images are from the same camera, and the humans/objects are moving at low speeds compared to the frame rate. However, when these conditions are not satisfied, there should be a method for accurate tracking in image sequences.

One common approach is to match the pixel distribution of the blobs in the two consecutive frames. For this, histograms of the foreground regions corresponding to the blobs are constructed and matched. However, the method is not robust in the presence of occlusion, as the histogram can change drastically if a region containing pixels with values close to the statistical mode of the distribution is occluded. We minimize this problem by using only the head-shoulder region. However, head rotations can result in significant changes in the histogram of the entire head-shoulder region, as the proportions of skin colour and hair colour change drastically with head rotations. Therefore we select only the chest region for histogram matching. Another advantage of this method is that it is possible to match views of a human from two cameras using this method, for most types of clothing and hair styles. However, the method cannot be employed alone in the presence of multiple humans with similar clothing. In such cases, additional information is required for accurate matching. Figure 6.3 illustrates the process of histogram computation and matching for the chest region. For this work we consider 80% correlation as the threshold for matching histograms.

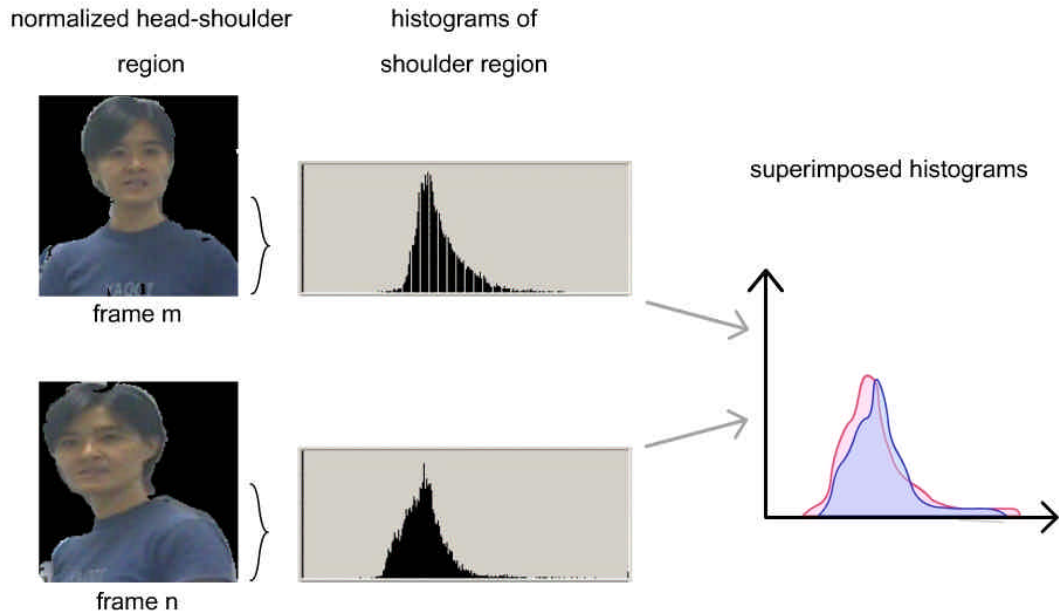


Figure 6.3: Histogram Computation and Matching for the Chest Region.

6.3 Recognition of Events

We use a state-based approach to recognize events and actions. Each human detected in the room is considered to be in one of the states of a predefined state machine. According to actions and events detected, the state of a human is changed.

6.3.1 State Model

The state diagram in Figure 6.4 shows the transitions between states defined for a tracked human in the image sequence.

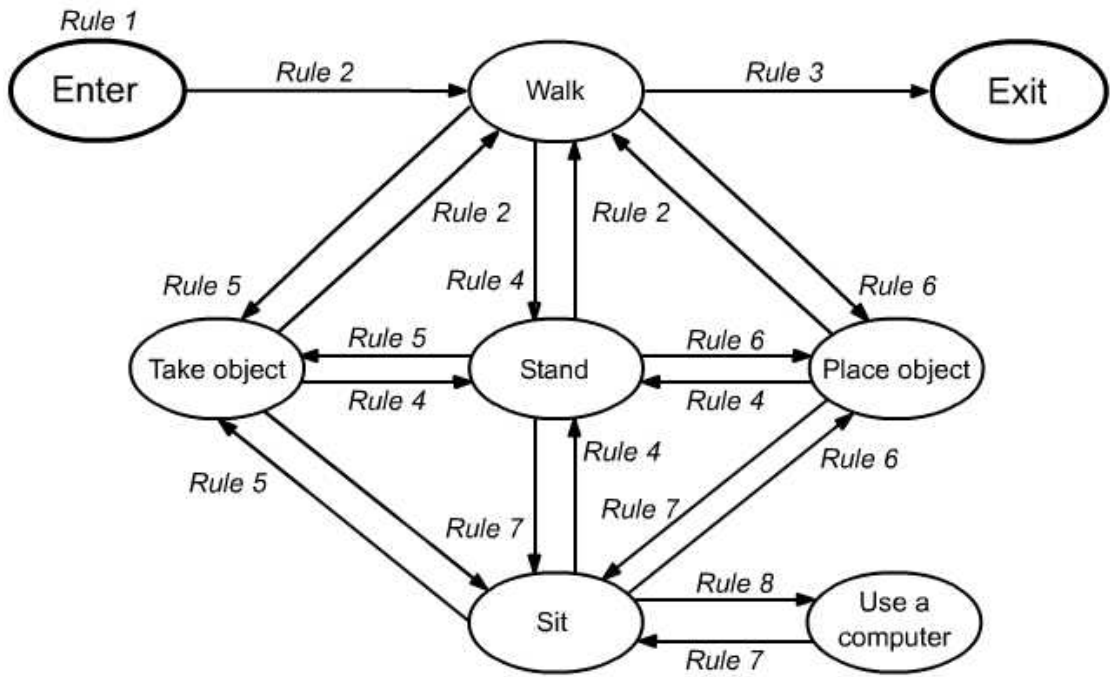


Figure 6.4 State Transitions for a Human Detected in the Scene.

The rules governing the state transitions in the above diagram are specified in Table 6.1.

Table 6.1: Set of Rules for State Transitions.

Rule number	State moved to	Specification
1	Enter (initial state)	New human model instance detected near the entrance region for 10 frames
2	Walk	coordinates of the centroid of the head-shoulder region changes gradually over 20 frames
3	Exit (final state)	Human model instance located near the entrance region for more than 10 frames, and could not be tracked thereafter
4	Stand	Y coordinate of the head centroid increases by more than 2 head-heights, with less than 10% change in head-height
5	Take object	Location of object changes together with the location of the person
6	Place object	New object detected on the table region for more than 15 frames
7	Sit	Y coordinate of the head centroid decreases by more than 2 head-heights, with less than 10% change in head-height
8	Use a computer	Person sitting near computer with only small movements below shoulder region

6.4 Detection of Unusual Events and Actions

The rules specified in Table 6.1 correspond to the most common events that could take place in a scene we have selected. However, it is possible that other actions or events that are important to be recorded take place. For example, there can be a situation where a person trying to block the camera. We keep an index to such an action as “unrecognized” to facilitate human observation to recognize the action. If the amount of scene change occurring between two frames is substantial and the action/event cannot be recognized, the scenario is identified as an unrecognized event. Key frames showing the scene change are extracted from the image sequence.

6.5 Tracking Persons in the Scene

Since only two uncalibrated cameras are employed for image acquisition, it is impossible to determine the location of a human or object in the room using the available images. Instead, a person in the scene is tracked by recording the path of the human on an image of the static background. Once a human enters the scene, a new background image is used to record his position. The resulting track images contain only one person per image even when there are multiple humans in the scene. Figure 6.5 illustrates how the track of a human is visualized.



Figure 6.5: Visualization of Human Tracking Results.

The regions corresponding to the human in the sequence of frames are superimposed on the background with time, resulting in a trail as shown in Figure 6.5 (a). Since this image can be cluttered for a large amount of movement close to the camera, another image is created by recording only the position of the centroid of the head region. The color of the centroid varies from green to red, giving an indication of the direction of movement and the time spent in the room. Again, trails for different persons are plotted on different frames. This is shown in of Figure 6.5 (b).




6.6 Indexing and Recording Key Frames

For each action/event detected, we record an index entry with the following attributes:

1. Frame number (can be converted to time for more convenient tracing)
2. Event type
3. Key frame/s

The entries can be indexed both by the frame number and the type of the event, facilitating fast searching. The key frames assist human verification without browsing the image sequence. Table 6.2 contains some possible index entries and key frames.

Table 6.2: Sample Entries of the Scene Index.

Frame number	Event type	Key frame/s
118	Entering person	
223	Standing person	
432	Sitting person	

For a sitting person and standing person, a pair of key frames is stored, showing both the seated and standing person. For a walking person, key frames are not saved.

By looking up the index and key frames, the image sequence can be browsed quickly without tracing sequentially for actions and events. Chapter 7 describes in detail the results obtained by evaluating the system using a number of image sequences.

Results and Discussion

This chapter presents the evaluation of the performance of the algorithms designed and implemented in previous chapters, and the results obtained. The results are categorized by the stages of the functional model. Also presented in this chapter is a critical discussion of these results.

7.1 Background Modeling and Foreground Extraction

7.1.1 Methods of Evaluation

There is no standard methodology for evaluating the performance of a technique for background modeling and foreground extraction. In some researches the accuracy of foreground segmentation as a separate stage is not evaluated, as it is an intermediate result. However, it is important to ensure that foreground extraction performs well, as subsequent processing is dependent on its output.

In most of the related research, the accuracy of background modeling and motion segmentation is demonstrated using images. Background image and other components of the model are shown to visualize background features. Segmented foreground is shown together with the background images and/or acquired images, to demonstrate how well segmentation has been performed [8][9][10][12][14][49][51]. This is a relatively convenient method, and the strong points and limitations of algorithms can be identified easily if sufficient information about the scene is available. However, this method of evaluation is subjective. Moreover, visual completion can make foreground appear more complete than it actually is.

If quantitative evaluation of segmentation algorithms is possible, it is possible to compare their performances more accurately and identify which algorithms are suitable for particular applications. The number of pixels that have been segmented incorrectly can be used as a metric of performance. However, this requires human intervention to determine ‘correct foreground’. De Silva [19] uses the percentage of inaccurately segmented pixels in the image to compare three techniques of background image construction and foreground segmentation.

We have used two methods for evaluating the performance of our algorithm. The first method is qualitative evaluation using images and foreground extracted from different image sequences. The second method is to measure the amounts of incorrectly segmented pixels of the following two categories:

- (a) False positive pixels: pixels that have been segmented as foreground, but correspond to background when segmented manually
- (b) False negative pixels: pixels that have been segmented as background, but correspond to foreground when segmented manually

Figure 7.1 illustrates these two types of pixels by showing them on a segmented binary image masked with the foreground.

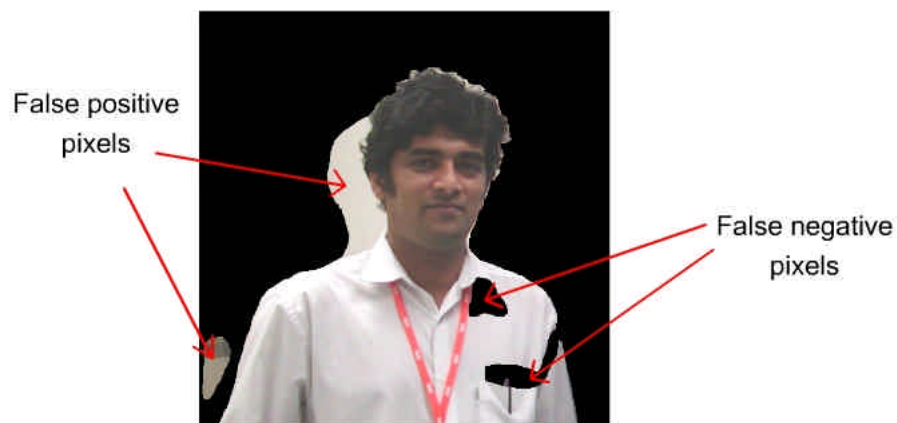


Figure 7.1: False Positive and False Negative Pixels.

We use the following formulae to calculate *Accuracy* and *Pixel Ratio* for the above types of pixels:

$$\text{Background accuracy} = \frac{N_b}{N_{fp} + N_b} \times 100 \quad (7.1)$$

$$\text{False positive pixel ratio} = 10 \text{Log}_{10} \left[\frac{N_b}{N_{fp}} \right] \quad (7.2)$$

$$\text{Foreground accuracy} = \frac{N_f}{N_{fn} + N_f} \times 100 \quad (7.3)$$

$$\text{False negative pixel ratio} = 10 \text{Log}_{10} \left[\frac{N_f}{N_{fn}} \right] \quad (7.4)$$

$$\text{Overall accuracy} = \frac{N_b + N_f - (N_{fn} + N_{fp})}{N_b + N_f} \times 100 \quad (7.5)$$

$$\text{Overall pixel ratio} = 10 \text{Log}_{10} \left[\frac{N_f + N_b}{N_{fn} + N_{fp}} \right] \quad (7.6)$$

Where

N_f = No. of pixels that correspond to the foreground

N_b = No. of pixels that correspond to the background

N_{fp} = No. of false positive pixels

N_{fn} = No. of false negative pixels

A logarithmic scale is used in equations (7.2), (7.4) and (7.6) since the ratios can have a wide range of values. Sections 7.1.2 and 7.1.3 describe the evaluation of foreground segmentation using these two methods.

7.1.2 Subjective Evaluation

Sequence 1: Stable scene

The scene contains a fairly consistent background with no multimodal regions. Figure 7.2 illustrates the results of background initialization, and the selection map. In this case, the selection map contains 1 in all pixels. The standard deviation image contains values roughly between 0 and 10. Its contrast has been increased drastically to make the patterns visible. A frame in the test sequence and the result of motion segmentation masked with the original frame, are shown in Figure 7.3.

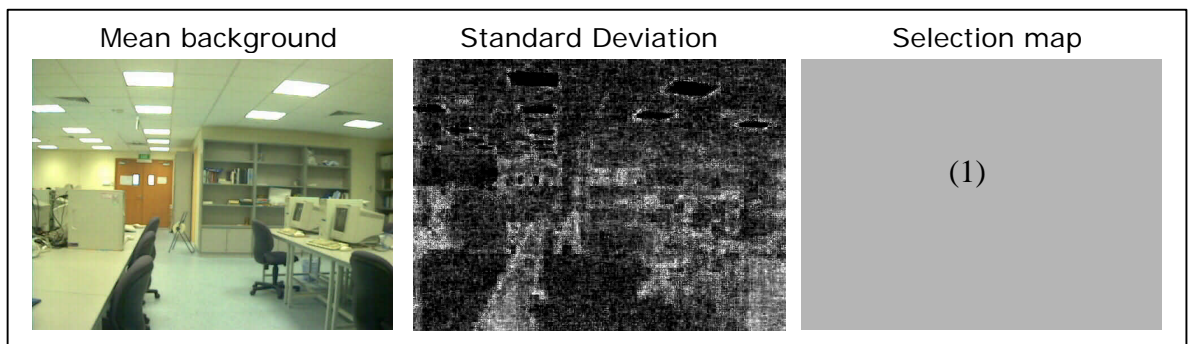


Figure 7.2: Background Initialization.

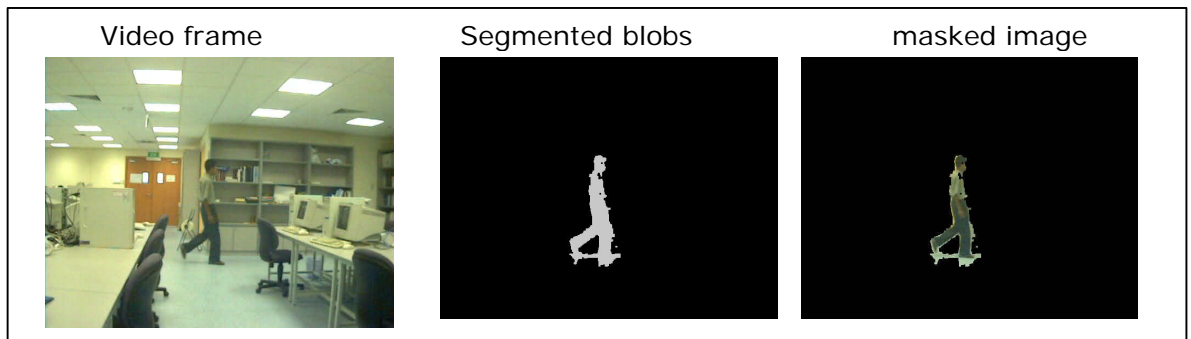


Figure 7.3: Motion Segmentation.

Sequence 2: Scene with high variation of illumination

Here the background is consistent. But there is a drastic change in lighting, in the middle of the sequence. Figure 7.4 illustrates the results of background initialization, together with the selection map superimposed on the mean background image. Again, the contrast of the standard deviation image has been increased drastically to make the patterns visible. In this case, motion segmentation is not needed in an area to the right of the image. Figure 7.5 displays the result of the abrupt change in lighting, if background adaptation was not used. Figure 7.6 illustrates how background adaptation produces better segmentation and recovers quickly. A few small blobs still remain, but these can be removed easily using size constraints in the subsequent modules for analyzing blobs.

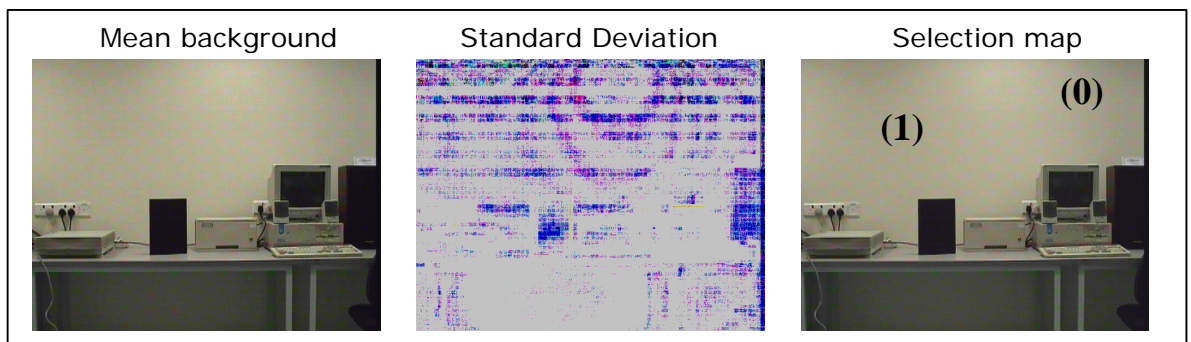


Figure 7.4: Background Initialization.

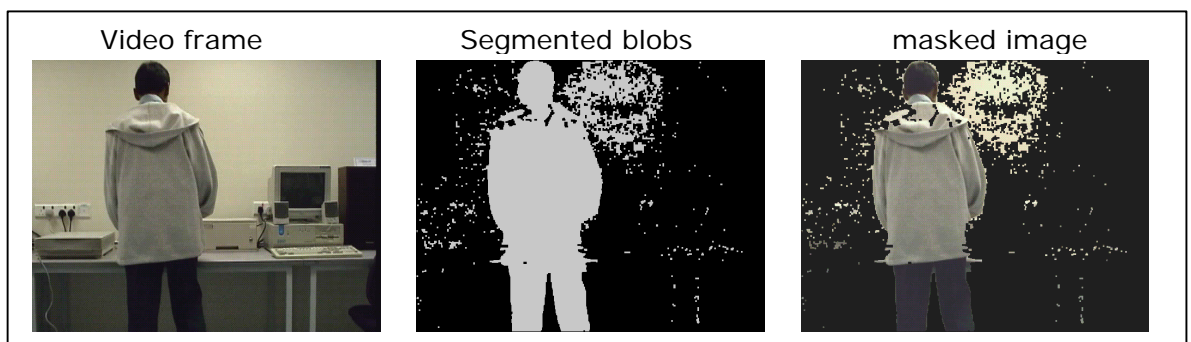


Figure 7.5: Motion Segmentation without Adaptation & Selection Map.

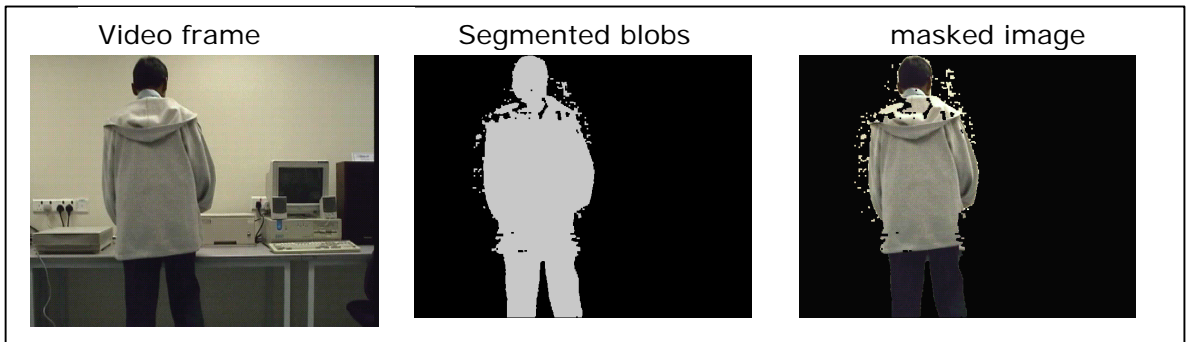


Figure 7.6: Adaptive Motion Segmentation.

Sequence 3: Scene with multimodal areas

The computer Monitor in this scene results in a bimodal region in the images. Figure 7.7 illustrates background initialization, and the selection map superimposed on the mean background image. The contrast enhanced standard deviation image shows that there is high variance in the region corresponding to the Computer monitor. Note the region with bimodal variance in the selection map. Two frames in the test sequence and the result of motion segmentation masked with the original frames are shown in Figures 7.8 and 7.9. Figure 7.9 illustrates how the foreground region in front of the multimodal region has been segmented with a reasonable accuracy.

Note that the standard deviations of all pixel values have been exaggerated so that their pattern can be visualized. Their actual levels are much lower.

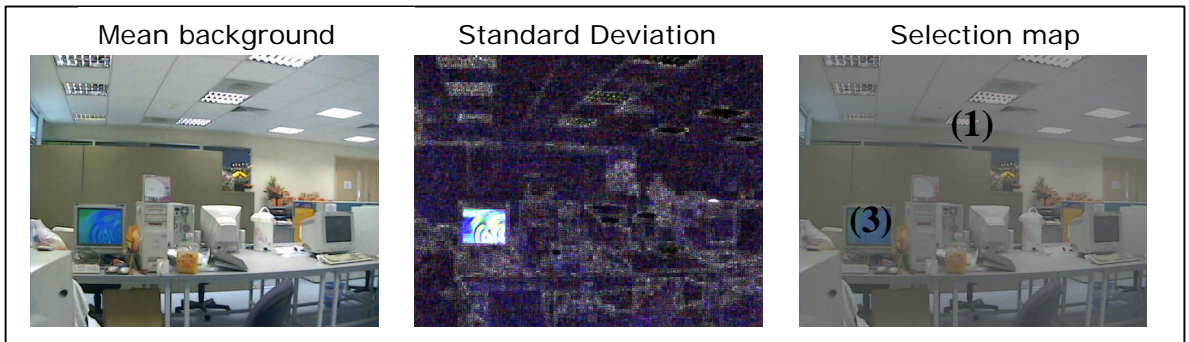


Figure 7.7: Background Initialization.

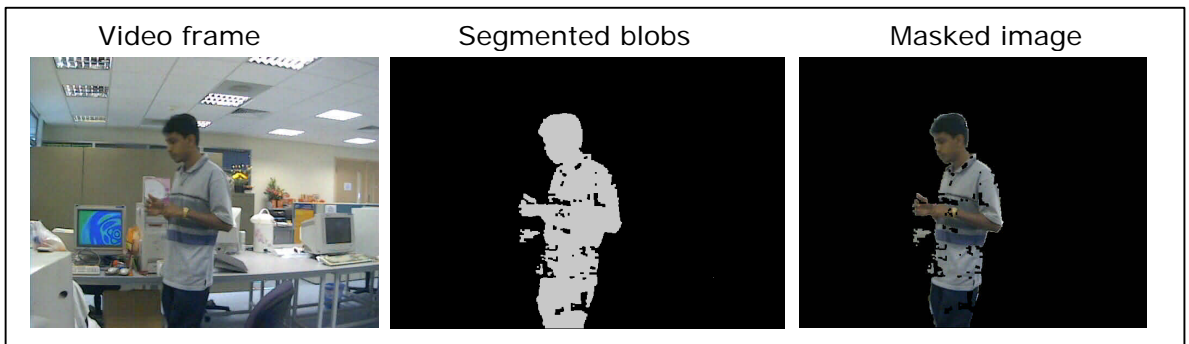


Figure 7.8: Adaptive Motion Segmentation.

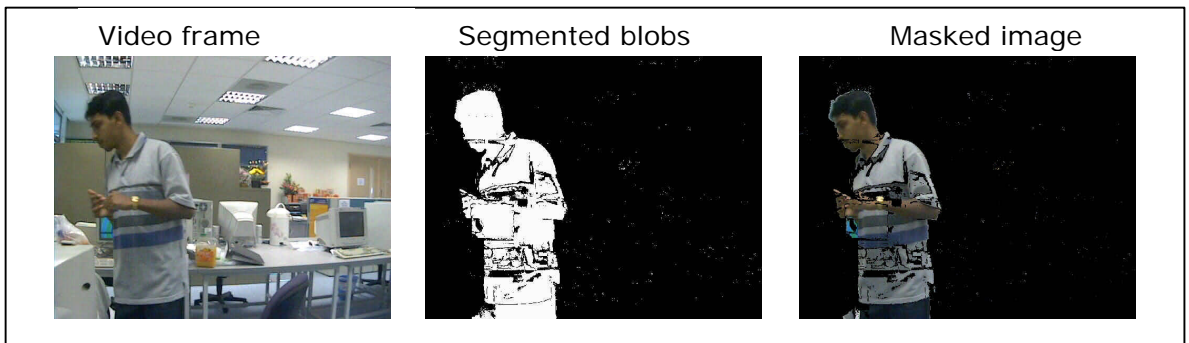


Figure 7.9: Adaptive Motion Segmentation in Multimodal Regions.

It is evident that the system performs motion segmentation accurately in all three sequences. The background adaptation enables accurate motion segmentation in a changing background. The results of segmentation are used as input to the next stage.

7.1.3 Quantitative Evaluation

For quantitative evaluation, the set of sequences described in Section 7.1.1 were used. Table 7.1 shows the values for accuracy and pixel ratio calculated for conventional non-adaptive foreground segmentation, using the equations 7.1 to 7.6.

Table 7.1: Results for Non-adaptive Foreground Segmentation.

Image sequence	Accuracy (%)			Pixel Ratio(dB)		
	Foreground	Background	Overall	Foreground	Background	Overall
1	99.36	97.28	99.28	21.92	15.54	21.41
2	93.14	97.95	94.42	11.33	16.79	12.28
3	98.24	92.30	97.48	17.48	10.79	15.88
Average	96.92	95.84	97.06	16.91	14.37	16.52

Table 7.2 shows the results obtained using the adaptive foreground segmentation technique that we designed and implemented.

Table 7.2: Results for Adaptive Foreground Segmentation.

Image sequence	Accuracy (%)			Pixel Ratio(dB)		
	Foreground	Background	Overall	Foreground	Background	Overall
1	99.39	97.86	99.33	22.09	16.61	21.70
2	99.22	97.62	98.86	21.02	16.12	19.38
3	99.59	91.83	98.71	23.80	10.51	18.85
Average	99.40	95.77	98.97	22.30	14.41	19.98

It is evident that the accuracy and the pixel ratio are nearly equal for the background for both techniques, in all three sequences. Moreover, foreground accuracy and SNR for sequence 1 have nearly equal values for both techniques. However, the adaptive segmentation technique provides much better results for

sequences 2 and 3, which include illumination changes and multi-modal regions respectively.

In addition to the above calculations, background accuracy was calculated for an image sequence with no foreground present. This calculation was made for every frame in the sequence, and the results were plotted against the frame numbers. The resulting graph is shown in Figure 7.10.

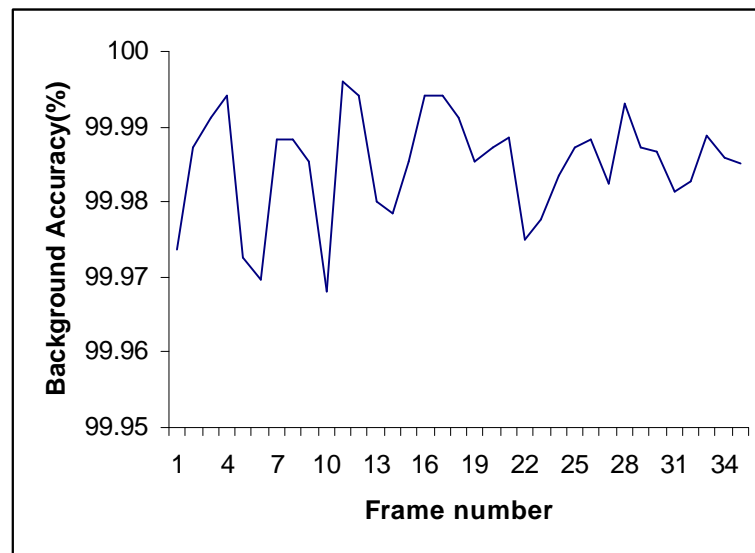


Figure 7.10: Variation of Background Accuracy over Time.

It is evident that background accuracy varies around an average of about 99.985%. This value is much higher than that for the image sequences analyzed before. The reason for this is the addition of false positive pixels due to morphological operations, in the presence of foreground. The amount of false positive pixels reduces in time due to background adaptation. It should be noted that this calculation can be automated since there is no foreground in any of the frames. A study of an image sequence with foreground requires an extremely large amount of manual image

processing, as the correct foreground for each frame has to be identified. Therefore, this calculation was not performed.

7.2 Human Detection and Body model Acquisition

The ability to detect humans in images was tested by using 400 images with 10 human subjects appearing in them, both alone and together. Subjects with different hairstyles and attire were selected to identify weaknesses in detection. These subjects appeared in images in different sizes and different degrees of occlusion. Objects were introduced to some of the images to test for false detections of them as humans. For convenience, the images were extracted from several image sequences from different scenes. For each scene, the background modeling and foreground extraction was carried out using the algorithms in Chapter 5.

7.2.1 Methods of Evaluation

The following parameters were defined to measure the performance of human detection:

$$\text{Accuracy} = \left(\frac{T_c}{T_h} \right) \times 100\% \quad (7.7)$$

$$\text{False detection rate} = \left(\frac{T_f}{T_h} \right) \times 100\% \quad (7.8)$$

where

T_h = Total number of human presences in the set of images

T_c = Number of humans detected correctly

T_f = number of false detection of foreground objects as human

One simple method of evaluation of the accuracy of human body model acquisition is subjective evaluation. The detected model was superimposed on the image to get a visual estimate of the accuracy of model acquisition. This method is straightforward and it is possible to get results using a large set of images.

Subjective evaluation of body model acquisition has two limitations. One problem is that the ranking system is subjective. The second problem is that it does not provide any quantified measure that can be used for improving the system or comparing its accuracy against other techniques where necessary. Therefore we proposed the following approach for quantitative evaluation of model acquisition. We manually acquired model parameters of 100 images, and compared with those obtained using the system. The manual acquisition of model parameters is a tedious task and cannot be guaranteed to be accurate or consistent in all images. For example one might estimate torso height with some error. This was performed only on images without occlusion as manual estimation of sizes for occluded regions is subjective. For each model parameter M_p , the accuracy of model acquisition was measured as

$$\text{Accuracy for } M_p = \frac{\text{The value of } M_p \text{ as obtained by the system}}{\text{The value of } M_p \text{ as obtained manually}} \times 100\% \quad (7.9)$$

The average accuracy for each parameter is calculated by averaging the accuracy over the selected set of images. Overall accuracy is calculated by averaging these accuracy values together.

7.2.2 Results of Human Detection

Table 7.3 summarizes the results of the evaluation of human detection. It is evident that the system has a very high accuracy in human detection and a very low false detection error.

Table 7.3: Results of Human Detection.

Number of images used	400
Number of human presences(T_h)	520
Number of humans detected correctly(T_c)	509
Number of false detection of foreground objects as humans(T_f)	4
Accuracy	97.8%
False detection rate	0.77%

Figure 7.11 shows some situations where accurate detection was possible. The background image which has a resolution of 640×480 pixels is included to demonstrate the performance of the system in detecting humans appearing in different sizes.

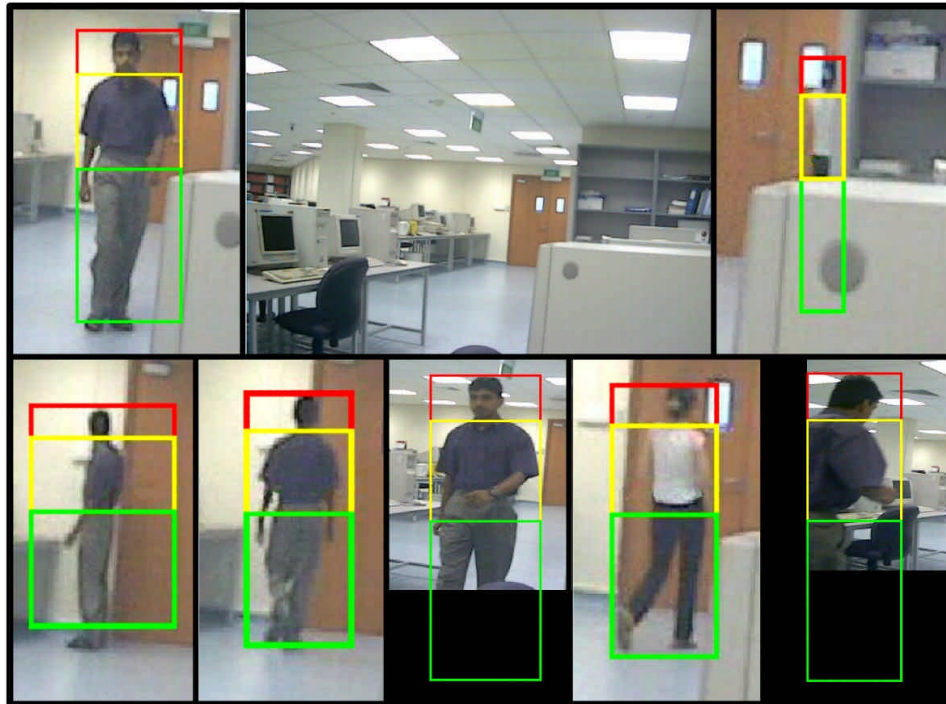


Figure 7.11: Results of Human Detection.

It is evident that human detection performs well despite very small regions representing a human, a high degree of occlusions and different angles of view.

7.2.3 Subjective Evaluation of Model Acquisition

A selected set of images where the detected model is superimposed on the image are shown in Figure 7.12. It is evident that the system has accurately detected the regions properly in images where the body is fully visible.

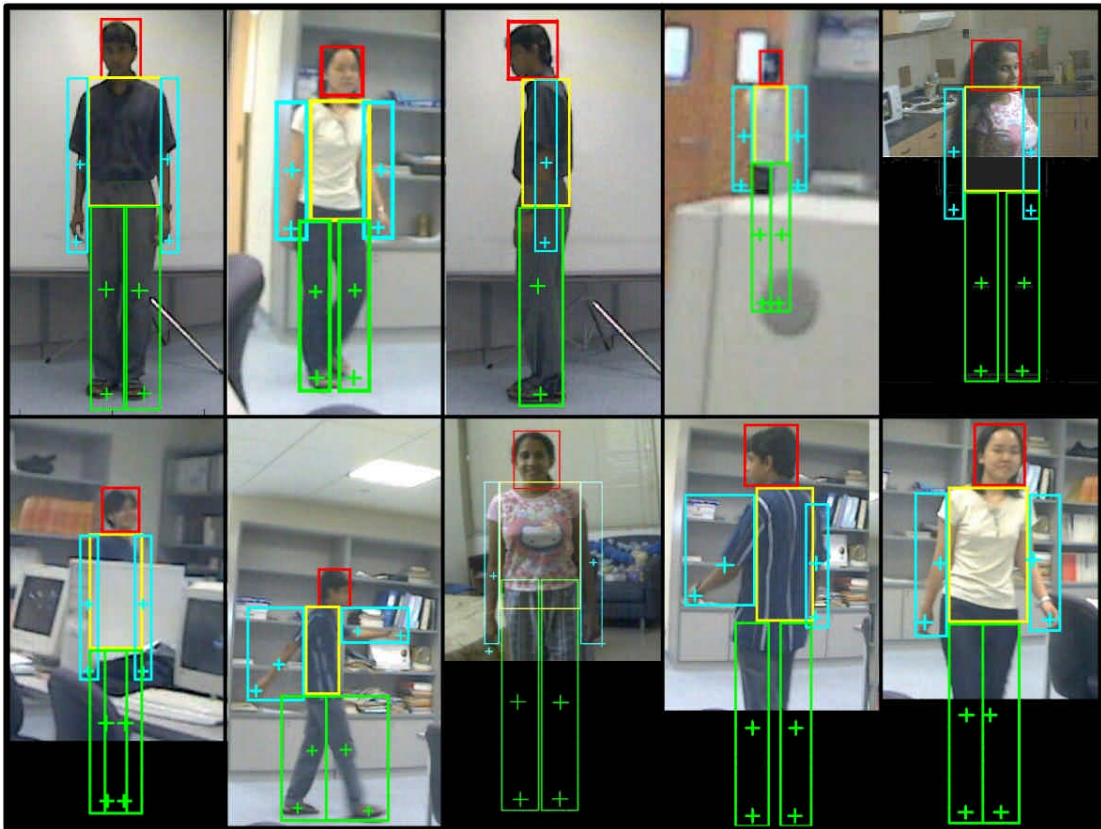


Figure 7.12: Images Used for Subjective Evaluation.

In the presence of occlusion, the proportions of the detected regions seem correct. It has been possible to estimate the model parameters with a reasonable accuracy, despite the presence of different hair styles.

7.2.4 Quantitative Evaluation of Model Acquisition

The results of the quantitative evaluation are summarized in Table 7.4. It is evident that an overall accuracy of above 94% can be achieved in complete body model acquisition.

Table 7.4: Accuracy of Body Model Acquisition.

Model parameter	Average accuracy (%)
Height of the head region	95.01
Height of the torso region	94.58
Width of the torso region	94.73
Length of arm	92.70
Length of leg	94.85
Distance between shoulder and elbow	93.68
Distance between waistline and knee	92.55
Overall average accuracy	94.01

7.3 Tracking

7.3.1 Quantitative Evaluation

Since tracking is an intermediate step of the system, there is no immediate method for evaluation. A method for quantitative evaluation is designed as follows.

We use 20 image sequences for evaluating the performance of tracking. For each image sequence, a set of frames is selected for evaluation of tracking. Each set of frames starts from the frame following the frame where a human is detected, and ends when the human leaves the scene or the sequence ends.

The accuracy of tracking, A_t , is measured as defined in equation 7.10.

$$A_t = \frac{N_t}{N} \times 100\% \quad (7.10)$$

where

N = total no. of frames selected

N_t = number of frames where the human has been tracked accurately

Table 7.5 shows the results of the quantitative evaluation using the selected image sequences. The average accuracy for tracking is around 90%.

Table 7.5: Results of Evaluation for Human Tracking.

Sequence	No. of Frames(N)	Tracked Frames(N_t)	Accuracy(A_t)
1	461	435	94.36
2	249	225	90.36
3	179	165	92.18
4	180	158	87.78
5	190	179	94.21
6	139	127	91.37
7	74	67	90.54
8	34	27	79.41
9	80	73	91.25
10	145	125	86.21
11	215	198	92.09
12	188	176	93.62
13	124	100	80.65
14	42	34	80.95
15	87	77	88.51
16	118	106	89.83
17	106	99	93.40
18	54	45	83.33
19	82	75	91.46
20	136	120	88.24
Total	2883	2611	90.57

7.4 Generation of the Index and Key Frames






An index generated by the system after running on a test image sequence and a set of key frames acquired for the same are presented in the following sub-section.






The key frames corresponding to all events and actions recognized by the system are presented separately for clarity.

7.4.1 Index of Events and Actions

Table 7.6 is an index created for one of the image sequences the system was tested on. The key frames are also included together with the index.

Table 7.6: Sample Index after Image Sequence Analysis.

Frame Number	Event/Action	Key frames
23	Enter	
43	Walk	
90	Stand	
153	Sit	
219	Stand	

239	walk	
310	Sit	
350	Use PC	
394	Stand	
414	Walk	

7.4.2 Key Frames

The following images are key frames corresponding to actions and events, extracted from different image sequences used for testing the system. Figure 7.13 shows a key frame saved when a person entered the room, while Figure 7.14 shows a key frame saved when a person left the room. The key frame shown in Figure 7.15 shows a person standing at one place in the room.



Figure 7.13: Key Frame Showing a Person Entering the Room.



Figure 7.14: Key Frame Showing a Person Leaving the Room.



Figure 7.15: Key Frame Showing a Standing Person.

When a person is sitting, a pair of key frames is generated showing the body postures before and after sitting. Figure 7.16 shows a pair of key frames saved when a person sits down.



Figure 7.16: Key Frames Showing a Person Sitting.

Figure 7.17 shows a key frame saved when a person is using a computer. The key frames recorded for the actions of placing an object on the table and taking an object away from the table are shown in Figures 7.18 and 7.19 respectively. Figure 7.20 shows a key frame recorded when a person in the room blocks the view of the camera.



Figure 7.17: Key Frame Showing a Person Using a Computer.



Figure 7.18: Key Frame Showing a Person Placing an Object.



Figure 7.19: Key Frame Showing a Person Removing an Object.



Figure 7.20: Key Frame Showing an Unusual Event.

7.4.3 Visualization of the path of movements

Figure 7.21 is an image frame extracted from an image sequence showing a person moving in the scene. Figure 7.22 illustrates the visualization of the path of movement for a single person in the scene.



Figure 7.21: Sample Frame from an Image Sequence Showing a Single Person.



Figure 7.22: Visualization of Motion Path for a Single Person.

Figure 7.23 is an image frame extracted from an image sequence showing two persons in the scene. Figures 7.24 shows how the paths of these two persons are visualized, enabling easy monitoring.



Figure 7.23: Sample Frame from an Image Sequence Showing Two Persons.



Figure 7.24: Visualization of Motion Path for Two Persons.

7.4.4 Evaluation of Event Recognition

The accuracy of action and event recognition was evaluated using 20 image sequences containing different actions and events. Table 7.6 shows the accuracy of recognition of events and actions in the image sequences that we used with the system.

Table 7.7: Accuracy of Action and Event Recognition.

Actual action/ event	Classified as									Not classified	Accuracy (%)
	Enter	Walk	Exit	Stand	Sit	Use PC	Take object	Place object	Unusual event		
Enter(10)	10	0	0	0	0	0	0	0	0	0	100
Walk(20)	0	18	2	0	0	0	0	0	0	0	90
Exit(10)	0	0	10	0	0	0	0	0	0	0	100
Stand(10)	0	0	0	8	0	0	0	0	0	2	80
Sit(10)	0	0	0	0	9	0	0	0	0	1	90
Use PC(10)	0	0	0	0	1	9	0	0	0	0	90
Take object(5)	0	0	0	0	0	0	5	0	0	0	100
Place object(5)	0	0	0	0	0	0	0	5	0	0	100
Unusual event (5)	0	0	0	0	0	0	0	0	5	0	100

The values obtained for classification accuracy are between 80% and 100%.

7.5 Discussion

In this research we designed and implemented a system that can detect humans and recognize a selected set of human actions and events using video image sequences acquired from stationary cameras mounted in an indoor scene. Context data related to the scene have been extensively used to achieve accurate results.

The assumptions made in the system design make it possible to be applied directly in indoor scenes under CCTV camera surveillance. The use of centralized

scene context increases the flexibility of the system. With only minimum changes to the rest of the system, the system can be tailored to function at different locations by entering appropriate scene context.

The results shown in the previous sections of this chapter indicate that the system is able to detect humans and recognize actions and events with a high accuracy. The following subsections will discuss the results obtained in different functional modules of the system.

7.5.1 Background Modelling and Segmentation

The background model that we have proposed uses a selection map to identify the algorithms most suitable for segmentation in each region. Since the technique is adaptive, slow changes in the background are allowed increasing the usability of the system. Background regions with high variance, such as those corresponding to computer monitors, are correctly segmented, as shown by the results.

An inherent problem with segmentation based on background subtraction is that it fails if the camera is moving. However, for small movements of the camera, background adaptation can remodel the background corresponding to the new camera position.

7.5.2 Human Detection and Modelling

The algorithm we propose here facilitates human detection in the presence of a very high degree of occlusion, as detection is based only in the head-shoulder region. As demonstrated by the high accuracy obtained, the algorithm is capable of human detection accurately irrespective of orientation, size and hair styles. The algorithm for modelling can acquire model parameters reasonably accurately, as seen in images used for subjective evaluation. The quantitative evaluation yields high accuracy, demonstrating that the algorithm is capable of modelling human bodies accurately despite the presence of occlusion.

7.5.3 Tracking and Generation of Results

Tracking of humans detected in the images is performed using three similarity measures. However, it seems that only two, namely overlapping bounding boxes and matching shoulder region histograms are sufficient. The histogram matching technique we suggest is quite suitable for situations where occlusion is present and tracking in images from multiple cameras is necessary.

The recognition of actions and events are based on a simple set of rules and a state machine. The state machine facilitates more accurate action recognition as the number of possible actions at a particular state is less than the total number of actions. The results indicate that some actions and events, such as entering the room and standing are detected more accurately than others.

The generation of an index to the video sequence results in reducing the search time drastically when it comes to tracing an incident. A faster way of browsing the image sequence is facilitated by the key frames. However, it is possible for the index to become large very quickly in a situation where several humans occupy the room and perform actions for a prolonged period of time, or if humans often move in and out of the room. Although an attempt can be made to detect re-entry to the scene using the histogram matching technique, its accuracy cannot be guaranteed as people with similar attire can be recognized as re-entrants. If a method for person recognition can be incorporated to the system, then it is possible to create a secondary index based on the name or identifier of the person so that the searches can be made narrower.

The visualization of the motion path is a simplified way of presenting how each human moved within the scene. However, the resulting image can be quite cluttered in case of a human moving a lot inside the scene.

Conclusion and Future Work

8.1 Conclusion

In this thesis, we have proposed the design and implementation of a system that can detect the presence of humans and recognize their actions in a closed environment, using video image sequences obtained from stationary cameras in the scene. An index to the sequences is created by recording the detected human presences and actions, facilitating faster searching of the image sequence for a particular event or action. Key frames extracted from the sequences and visualizations of paths traversed by the humans in the scene are recorded together with the index to provide a quick way of browsing the sequence.

The scene context is fed into the system in addition to the images to achieve accurate results. By changing the scene context as appropriate, the system can be used in different scenes.

The system uses Background subtraction for segmenting the foreground from the image frames. A new technique for background initialization and adaptation is used. The scene context and an initial image sequence are used for constructing a background model. This model is able to make a good representation of the scene due to the inclusion of a selection map. The selection map facilitates improved performance while avoiding unnecessary complexity in processing, by allowing different algorithms on different regions of the image. Segmentation and background adaptation take place together to ensure accurate segmentation under slow changes in

the background. The experimental results demonstrate that these techniques perform well in the presence of illumination changes and backgrounds with multimodal regions. It is possible to obtain an average segmentation accuracy of 99.0%, and a pixel ratio of 20.0 dB, as defined in Chapter 4. For image sequences with illumination changes, foreground segmentation has accuracy above 99%, compared to an accuracy of 93% using conventional foreground segmentation. . The background adaptation is robust and the background model stabilizes within 2-3 frames of an illumination change.

Human detection, in the presence of occlusion, is performed by using a top-down matching technique with a predefined head-shoulder model, up to an overall accuracy of 97.8%. The initial body model based on the detected height of the head is refined using image features and geometric constraints to achieve an average accuracy of 94% in constructing a complete body model. According to subjective evaluation, the accuracy of model acquisition was found to be reasonably accurate.

Once detected and modeled, the humans are tracked in the subsequent image frames. The average accuracy of tracking was found to be 90%. The model features obtained by tracking are then evaluated against a set of rules and a state machine to recognize actions and events. After evaluating using 20 sets of image sequences, the accuracy of Recognition was found to be between 80% and 100%. Subjective evaluation of key frames shows that they represent the corresponding actions reasonably well.

8.2 Future Directions

At its current state, the system can be used to facilitate semi-automated surveillance. Human intervention is necessary to search the index and key frames and observe the appropriate section of the video sequence to detect what exactly happened. However, the system can be improved in certain aspects such that it performs better and can be used as a system that can facilitate fully automated surveillance. The following sub-sections describe some such future directions.

8.2.1 Incorporating Person Recognition

As mentioned in section 7.5.3, the search time within the index can be reduced drastically if person recognition is incorporated to the system. This can be facilitated in a number of ways. It is possible to install smart-card enabled door locks and get the person identification from the card readers. However, the most common method used for person recognition in a smart environment is face recognition. This requires that a segmented image contains the face to be identified with a reasonable resolution. Several techniques for face recognition exist, while the most common techniques are based on Neural Networks and Elastic Graph Matching. Due to the high dimensionality of the problem, dimensionality reduction using Karhunen-Loeve Transform or Gabor wavelet transform is common. Although existing face recognition systems perform well on frontal images with consistent lighting, they are unable to recognize faces accurately in varying conditions using face images taken from different angles.

8.2.2 3D Human Body Modeling and Tracking

The human body model currently used is a 2 dimensional model. However, certain actions can be ambiguous when represented by a 2D model. Moreover, the system does not use the information from the whole model for action recognition. If a 3D body model can be acquired and used for action recognition, the results will be more accurate.

The current system performs tracking only with respect to image coordinates, not real world coordinates. If 3D tracking is possible, the actions can be recognized more accurately as the exact position of a human in the room is known. A set of calibrated cameras can be used to facilitate this, as the cameras used by the system are stationary and having fixed focus throughout the sequences.

8.2.3 Improving the Recognition Capability

The set of actions and events recognized by the system is quite limited. More actions can be incorporated to make the system more useful. Instead of limiting the system to recognize actions in terms of body gestures such as walking, standing and sitting, Hand and head gestures can also be incorporated. However, the main requirement for this is the ability to acquire high resolution images using a sufficient number of cameras to ensure that the hands are visible to at least one of the images all the time.

8.2.4 Facial Expression Recognition

If facial expressions of humans in the scene can be recognized, the functionality of the system can be greatly enhanced. Facial expressions convey a large

amount of information about human emotions and behavior. They convey this information much quicker than any other source of information. Moreover, the facial expressions are the most difficult to suppress, implying that the accuracy of the information is high, provided that the facial expressions can be identified correctly. For these reasons, automated recognition of facial expressions has been a very active research topic.

However, the task of automatic facial expression recognition is quite difficult, due to a number of reasons. The difference between some facial expressions is only a couple of muscular movements. Therefore adequate resolution in face images becomes a necessary condition. Some expressions, for example raising eye brows, have a temporal variation. In addition, noise, partial occlusions, different orientations of faces, has to be dealt with.

A popular approach to facial expression recognition is based on the Facial Action Coding System (FACS) developed by Ekman and Friesen in 1978. In this system, a face is broken down into 44 action units (AUs), 30 of which are related to a contraction of specific facial muscles, and 14 which are unspecified [97]. Research laboratories such as Vision and Autonomous Systems Centre, Carnegie Mellon University, expand on this system and create their own databases, which encompass many more variables [98]. This database can be used in facial expression recognition systems with greater accuracy than FACS.

However, there are other approaches for facial expression recognition. Most of them are common in the area of pattern recognition. These include maximum

likelihood estimates, artificial neural networks, Gabor filters and Eigen/Fisher face based algorithms [42][99].

Author's Publications

- [1]. G. C. De Silva, L. C. de Silva, S. Ranganath, "Background Modeling and adaptation using Selection Maps for Motion Segmentation", Proceedings of the "2002 International Conference on Imaging Science, Systems and Technology", Las Vegas, USA, June 2002, pp.682-688.

- [2]. Gamhewage C. de Silva, Michael J. Lyons, Shinjiro Kawato and Nobuji Tetsutani, "Human Factors Evaluation of a Vision-Based Facial Gesture Interface", Published in the proceedings of the Workshop in Human Computer Interaction of the IEEE conference in Computer Vision and Pattern Recognition, 2003.

- [3]. G. C. de Silva, C. R. de Silva, L. C. de Silva, "Robust Human Body Model Acquisition from Images in the Presence of Occlusion", Extended abstract accepted and full paper submitted to the Symposium on Research for Industry, Engineering Research Unit, University of Moratuwa, Sri Lanka.

- [4]. G. C. de Silva, Michael J. Lyons, Shinjiro Kawato and Nobuji Tetsutani, "Point with Nose and Click with Mouth: Vision-Based Face Tracking for Cursor Control", Published in the proceedings of the Computer Science and Engineering Conference 2003, Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka.

References

- [1]. A. P. Pentland, “Smart Rooms”, <http://vismod.www.media.mit.edu/vismod/demos/smartroom/ive.html>.

- [2]. A. Bobick, “Kids’ Room”, <http://www-white.media.mit.edu/vismod/demos/kidsroom/kidsroom.html>

- [3]. Massachusetts Institute of Technology, “MIT Media Laboratory”, <http://www.media.mit.edu>.

- [4]. M. J. Lyons, M. Haehnel, N. Tetsutani, “The *Mouthesizer*: A Facial Gesture Musical Interface”, Conference Abstracts, *Siggraph* 2001, p. 230.

- [5]. G. C. de Silva, M. J. Lyons, S. Kawato, N. Tetsutani, "Human Factors Evaluation of a Vision-Based Facial Gesture Interface", Accepted for publication in the proceedings of the Workshop in Human Computer Interaction of the IEEE conference in Computer Vision and Pattern Recognition, 2003.

- [6]. D.O. Gorodnichy, S. Malik, G. Roth, “Nouse: ‘Use Your Nose as a Mouse’ - a New Technology for Hands-free Games and Interfaces”, Proceedings of International Conference on Vision Interface (VI'2002), pp. 354-361.

- [7]. IGN Entertainment Inc. "Play Station: Dance Dance Revolution", <http://psx.ign.com/articles/131/131525p1.html>.
- [8]. S. J. McKenna, S. Jabri, Z. Duric, H. Wechsler, "Tracking interacting People", Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition, pp.348-353, 2000.
- [9]. C. R. Wren, A. Azerbayejani, T. Darrel, A. Pentland, "Pfinder: Real-Time Tracking of the Human Body", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, pp. 780-785, 1997.
- [10]. R. Rosales, S. Sclaroff, "Improved Tracking of Multiple humans with Trajectory Prediction and Occlusion Modeling", Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Workshop on the Interpretation of Visual Motion, Santa Barbara, CA, pp. 357-360, 1998.
- [11]. N. K. Piau, "Tracking people", Final Year Project Report, Department of Electrical and computer engineering, National university of Singapore, Singapore, 2001.
- [12]. I. Haritaoglu, D. Harwood, L. Davis, "W4: Who, When, Where, What: A Real-time system for Detecting and Tracking People", Third IEEE International Conference on Face and Gesture Recognition, pp. 222-227, Nara, Japan, 1998.

- [13]. T. Zhao, R. Nevatia and F. Lv. "Segmentation and Tracking of Multiple Humans in Complex Situations", In the proceedings of the IEEE conference on Computer Vision and Pattern Recognition, December 2001, Vol. II, pp. 194 - 201.
- [14]. A. Utsumi, H. Mori, J. Ohya, M. Yachide, "Multiple View-Based Tracking of multiple Humans", Proceedings of the Fourteenth International Conference on Pattern Recognition, 1998, Vol.1, pp.597 -601, 1998.
- [15]. C. Riddler, O. Munkelt, H. Kirchner, "Adaptive Background Estimation and Foreground Detection Using Kalman Filtering", Proceedings of International Conference on Recent Advances in Mechatronics, ICRAM '95, UNESCO Chair on Mechatronics, pp.193-199, 1995.
- [16]. C. Stauffer, W. E. L. Grimson, "Adaptive Background Mixture Model for Real-time Tracking", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 8, pp. 747-757, 2000.
- [17]. P. D. O'Malley, M. C. Nechyba, A. A. Arroyo, "Human Activity Tracking for Wide-Area Surveillance", Proceedings of the 2002 Florida Conference on Recent Advances in Robotics (FCRAR), pp. 7-13.
- [18]. S. Khan, M. Shah," Tracking People in Presence of Occlusion", *Asian Conference on Computer Vision*, Taipei, Taiwan, Jan 2000.

- [19]. G. C. de Silva, "Traffic Flow Measurement Using Video image Sequences", M. Eng. Thesis, Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka, 2001.
- [20]. A. Pentland, T. Choudhury, "Face Recognition for Smart Environments", *Computer*, February 2000, pp. 50-55, IEEE Press, United Kingdom, February 2000.
- [21]. I. Haritaoglu, M. Flickner, "Detection and Tracking of Shopping Groups in Stores", Proceedings of International conference in Computer Vision and Pattern Recognition, 2001, Vol. I, pp. 431-438.
- [22]. Y. Rosenberg and M. Werman, Real-Time Object Tracking from a Moving Video Camera: A Software Approach on a PC, IEEE Workshop on Applications of Computer Vision, Princeton, Oct 1998, pp. 238-239.
- [23]. D. S. Hyeon, S. Doo K, S. G. Jahng, J. S. Choi, "Human Detection in Images using Curvature Model", *International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC2001)*, Tokushima Japan, July, 2001.
- [24]. K. Tabb, N. Davey, S. George, R. Adams, Detecting Partial Occlusion of Humans Using Snakes and Neural Networks, Proc. 5th International Conference on Engineering Applications of Neural Networks (EANN'99), 34-39, 1999 [ISBN 8371745125].

- [25]. S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler, "Detection and location of people in video images using adaptive fusion of color and edge information, " in International Conference on Pattern Recognition, pp. 1-1, 2000.
- [26]. Schoepflin, T., Chalana, V., Haynor, D. R. and Kim, Y., "Video object tracking with a sequential hierarchy of template deformations," IEEE Trans. Circuits and Systems for Video Technology, Vol. 11, pp. 1171-1182, 2001.
- [27]. R. Want, G. Boriello. "Survey on Information Appliances", IEEE Computer Graphics & Applications, May/June 2000, pp. 24-31.
- [28]. M. Lee, " Detecting People in Cluttered Indoor Scenes", in Proceedings of the IEEE conference in Computer Vision and Pattern Recognition, 2000.
- [29]. G. D. Abowd, "Classroom 2000: An Experiment with the Instrumentation of a Living Educational Environment", IBM Systems J., Special Issue on Pervasive Computing, Vol. 38, No. 4, Oct. 1999, pp. 508-530.
- [30]. D. J. Moore et al., "Implementing Phicons: Combining Computer Vision with Infra Red Technology for Interactive Physical Icons", Proc. ACM UIST 1999, ACM Press, New York, pp. 67-78, Nov. 1999.

- [31]. D. Ayers, M. Shah. "Monitoring Human Behavior from Video Taken in an Office Environment", *Image and Vision Computing*, Vol. 19, Issue 12, pp. 833-846, Oct. 2001.
- [32]. A. P. Pentland, "Looking at People: Sensing for Ubiquitous and Wearable Computing", *IEE Trans. On Pattern Analysis and Machine intelligence*, Vol. 22, No. 1, pp. 107-118, January 2000.
- [33]. N. Oliver, B. Rosario, A. Pentland, "Statistical Modeling of Human Interactions, Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 39-46, 1998.
- [34]. Y. Ivanov, C. Stauffer, B. Bobick, W. E. L. Grimson, "Video Surveillance of Interactions", *IEEE Workshop on Video Surveillance*, Fort Collins, Colorado, June 1999.
- [35]. D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, S. Russal, "Towards Robust Automatic Traffic scene Analysis in Real-Time", *Proceedings of the International Conference on Pattern Recognition*, Israel, pp.224-229, 1994.
- [36]. M. Petkovic, W. Jonker, "Content-based video retrieval by integrating spatio-temporal and stochastic recognition of events", *Detection and Recognition of Events in Video*, 2001, *Proceedings. IEEE Workshop on*, 2001, pp. 75 -82.

- [37]. R. C. K. Hua, L. C. de Silva, P. Vadakkepat, "Detection and Tracking of Faces in Real-Time Environments", Proceedings of Conference in Imaging Science, Systems and Technology 2002, pp. 51-57.
- [38]. C. R. Wren, "Dynamic Models for Smart Rooms", <http://www-white.media.mit.edu/~cwren/ttt/wren.html>.
- [39]. M. J. Er, S. Wu, J. Lu; H. L. Toh, "Face Recognition with Radial Basis Function (RBF) Neural Networks", Neural Networks, IEEE Transactions on, Volume: 13 Issue: 3, May 2002, Pp. 697 -710.
- [40]. M. J. Escobar, J. Ruiz-del-Solar, "Biologically-Based Face Recognition Using Gabor Filters and Log-Polar Images", Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on, Volume: 2, 2002, pp. 1143 -1147.
- [41]. Q. Liu, R. Huang, H. Lu, S. Ma, "Face Recognition Using Kernel Based Fisher Discriminant Analysis", Automatic Face and Gesture Recognition Fifth IEEE International Conference on, 20-21 May 2002, pp. 187 -191.
- [42]. L. C. de Silva, P. Vadakkepat, W. K. Teo, "Facial Expression Detection and Recognition System", Proceedings of Conference in Imaging Science, Systems and Technology 2002, pp. 559-565.

- [43]. M. J. Lyons, J. Budynek, A. Plante, S. Akamatsu, "Classifying Facial Attributes Using a 2-D Gabor Wavelet Representation and Discriminant Analysis", Proceedings of the 4th International Conference on Automatic Face and Gesture Recognition, 28-30 March, 2000, Grenoble, France, IEEE Computer Society, pp. 202-207.
- [44]. B. E. Shpungin, J. R. Movellan, "A Multi-threaded Approach to Real Time Face Tracking", Technical Report TR2000.02, Machine Perception Lab, University of California San Diego.
- [45]. J. Rehg, T. Kanade, "Model-based Tracking of Self-occluding Articulated Objects", International Conference of Computer Vision, pp. 35-46, 1995.
- [46]. D. Hogg, "Model-based Vision: a Paradigm to See a Walking Person", Image and Vision Computing, 1-1, pp. 5-20, 1983.
- [47]. K. Rohr, "Towards Model-based Recognition of Human Movement in Image Sequences", CVGIP, Image Understanding, 59-1, pp. 94-115, 1995.
- [48]. D. Gavriila, L. Davis, "Tracking of Humans in Actions: A 3D Model-based Approach", ARPA Image Understanding Workshop, pp. 737-746, February 1996.

- [49]. J. W. Davis, "Appearance-based Motion Recognition of Human Actions", Technical Report No. 387, Perceptual Computing Group, M. I. T. Media Lab, 1996.
- [50]. T. Roberts, S. McKenna and I. Ricketts, "Adaptive Learning of Statistical Appearance Models for 3D Human Tracking", Proceedings of British Machine Vision Conference, 2002, pp. 333-342.
- [51]. I. Mikic, M. Trivedi, E. Hunter, P. Cosman, "Articulated Body Posture Estimation from Multi-Camera Voxel Data", Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2001, Vol. I, pp. 455 – 460.
- [52]. C. Sminchisescu, B. Triggs, "Covariance Scaled Sampling for Monocular 3D Body Tracking", Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2001, Vol. I, pp. 447 - 454.
- [53]. J. Yamato, J. Ohya, K. Ishii, "Recognizing Human Actions in Time Sequential Images Using Hidden Markov Models", In CVPR,1992, pp. 624-630.
- [54]. K. Akita, "Image Sequence Analysis of Real-world Human Motion", Pattern recognition, 17, 1994, pp. 73-83.

- [55]. R. Rosales, S. Sclaroff, "Trajectory Guided Tracking and Recognition of Actions", Technical Report BU-CS-TR-99-002, Computer science Department, Boston University, 1999.
- [56]. G. Shakhnarovich, L. Lee and T. Darrell, "Integrated Face and Gait Recognition with Multiple Views" Proceedings of IEEE Conference on Computer vision and Pattern Recognition, 2001, Vol. 1, pp. 439-446.
- [57]. R. Rosales, M. Siddiqui, J. Alon, S. Sclaroff, "Estimating 3D Body Pose using Uncalibrated Cameras ", Proceedings of IEEE Conference on Computer vision and Pattern Recognition, 2001, Vol. 1, pp. 821-827.
- [58]. R. Polana, R. Nelson, "Low Level Recognition of human motion", IEEE Workshop on Non-rigid and Articulated motion, 1994, pp. 77-82.
- [59]. E. Shavit, A. Jepson, "Motion Understanding Using Phase Portraits", In IJCAI Workshop: Looking at People, 1995.
- [60]. J. Little, J. Boyd, "Describing Motion for Recognition", International Symposium on Computer Vision pp.235-240, November 1995.
- [61]. C. Rao and M. Shah, "View Invariance in Action Recognition", Proceedings of International conference in Computer Vision and Pattern Recognition, 2001, Vol.2, pp.316-322.

- [62]. J. Ben-Arie, Z. Wang, P. Pandit, S. Rajaram, "Human Activity Recognition Using Multidimensional Indexing", *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(8), pp. 1091-1104 (2002)
- [63]. I. Mikic, "Human Body Model Acquisition and Tracking using multi Camera Voxel Data", PhD Thesis, Dept. of Electrical and computer Engineering, University of California, San Diego, 2002.
- [64]. C. Sminchisescu, "Estimation Algorithms for Ambiguous Visual Models", PhD Thesis, Institut National Polytechnique de Grenoble, France, 2002.
- [65]. T. Zhao, R. Nevatia, "3D Tracking of Human locomotion: A Tracking as Recognition Approach", In *Proceedings of International Conference in Pattern Recognition*, 2002.
- [66]. S. X. Ju, M. Black, Y. Yacoob, "Cardboard People: A Parameterized Model of Articulated Image Motion", *Proc. 2nd International Conference on Automatic Face and Gesture Recognition (FG '96)*, October 14 - 16, 1996, pp. 38-44.
- [67]. I. Kakadiaris, D. Metaxas, "Three-Dimensional Human Body Model Acquisition from Multiple Views", *International Journal of Computer Vision*, vol. 30, no. 3, 1998, pp. 191-218.

- [68]. I. Kakadiaris, D. Metaxas, "Model-Based Estimation of 3D Human Motion with Occlusion Based on Active Multi-Viewpoint Selection", Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 1996.
- [69]. D. Metaxas, D. Terzopoulos, "Shape and Non-rigid Motion Estimation through Physics-Based Synthesis", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 15, no. 6, June 1993, pp. 580-591.
- [70]. C. Bregler, J. Malik, "Tracking People with Twists and Exponential Maps", IEEE International Conference on Computer Vision and Pattern Recognition, 1998.
- [71]. J. Deutscher, A. Blake, I. Reid, "Articulated Body Motion Capture by Annealed Particle Filtering", IEEE International Conference on Computer Vision and Pattern Recognition, 2000.
- [72]. J. Deutscher, A. Davison, I. Reid, "Automatic Partitioning of High Dimensional Search Spaces associated with Articulated Body Motion Capture", IEEE Int. Conference on Computer Vision and Pattern Recognition, 2001.
- [73]. M. Isard, A. Blake, "Visual tracking by stochastic propagation of conditional density", Proc. 4th European Conference on Computer Vision, 1996.
- [74]. R. L. Hsu, M. Abdel-Mottaleb, A. K. Jain, "Face detection in color images," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 696--

706, May 2002.

- [75]. L. Zhao, “Dressed Human Modeling, Detection, and Parts Localization” doctoral dissertation, tech. report CMU-RI-TR-01-19, Robotics Institute, Carnegie Mellon University, July, 2001.
- [76]. S. Hongseng, R. Nevatia, “Multi-Agent Event Recognition”, in proceedings of IEEE International Conference on Computer Vision, 2001
- [77]. C. Farago, “Leonardo da Vinci: Leonardo’s Writings and theory of Art”, Garland Publishing Inc, New York, USA, 1999.
- [78]. C. McLean, C. Brown, “Drawing from Life: 2nd Edition”, Harcourt race College Publishers, Florida, USA, 1997.
- [79]. J. Sheppard, “Realistic Figure Drawing”, North Light Books, Ohio, USA, 1991.
- [80]. C. Riddler, O. Munkelt, H. Kirchner, “Adaptive Background Estimation and Foreground Detection Using Kalman Filtering”, Proceedings of International Conference on Recent Advances in Mechatronics, ICRAM ’95, UNESCO Chair on Mechatronics, pp.193-199, 1995.
- [81]. G. C. de Silva , L. C. de Silva, S. Ranganath, “Background Modeling and adaptation using Selection Maps for Motion Segmentation”, Proceedings of the “2002 International Conference on Imaging Science, Systems and Technology

(CISST)”, pp. 682-688.

- [82]. A. Donald, D. A. Jusko, Human Figure Drawing Proportions, <http://www.mauigateway.com/~donjusko/human.htm>.
- [83]. I. Sommerville, “Software Engineering: 5th Edition”, Addison-Wesley Publishers, USA, 1998.
- [84]. A. Mitiche, P. Bouthemy, “Computation and analysis of image motion: a synopsis of current problems and methods”, Journal of Computer Vision, 1996, Vol. 19, 29-55.
- [85]. B. Horn, B. Schunk, “Determining optical flow”, Artificial intelligence, 1981, Vol. 17, 185-203.
- [86]. H. H. Nagel, W. Enkelmann, “An investigation of smoothness constraints for the estimation of displacements vector fields from image sequences”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1996, Vol. 8, 565-593.
- [87]. C. Rao, M. Shah, View Invariance in Action Recognition, Computer Vision and Pattern Recognition, CVPR 2001, Kauai, Hawaii, Dec 11-13, 2001
- [88]. Tao Zhao, Ram Nevatia, 3D Tracking of Human Locomotion: a Tracking as Recognition Approach, *ICPR02*.

- [89]. C. Stauffer and W.E.L. Grimson. Similarity templates for detection and recognition. In *Proc. Computer Vision and Pattern Recognition*, Kauai, Hawaii, 2001, Vol.1, pp. 221-228.
- [90]. R. Rosales, Recognition of Human Action Using Moment-Based Features, Boston University Computer Science Technical Report BU 98-020, November 1998.
- [91]. A. Bobick and J. Davis, "The Representation and Recognition of Action Using Temporal Templates", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 3, 2001, pp. 257-267.
- [92]. Cen Rao, Mubarak Shah. A View-Invariant Representation of Human Action, International Conference on Control, Automation, Robotics and Vision, Singapore, Dec 5th-8th, 2000.
- [93]. B. Ozer, W. Wolf, Ali N. Akansu, "Human Activity Detection in MPEG Sequences", *Proceedings of IEEE Workshop on Human Motion*, Austin, pp. 61-66, December 2000.
- [94]. W. Wolf, B. Ozer, "A Smart Camera for Real-time Human Activity Recognition," *Proceedings of IEEE Workshop on Signal Processing Systems*, Antwerp, Belgium, September 2001.

- [95]. R. Jain, Extraction of Motion Information from Peripheral Processes, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume PAMI-3, Number 5, September 1981, pp. 489-503.
- [96]. P. Castro, R. Muntz, "Managing Context Data for Smart Spaces", IEEE Personal Communications, October 2000, pp. 44-46.
- [97]. G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, T. J. Sejnowski, "Classifying facial actions" IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(10), pp.974-989.
- [98]. Robotics Institute, Carnegie Mellon University, "Facial Expression Analysis", http://www.ri.cmu.edu/projects/project_10.html
- [99]. M. J. Lyons, J. Budynek, A. Plante, S. Akamatsu, "Classifying Facial Attributes Using a 2-D Gabor Wavelet Representation and Discriminant Analysis", Proceedings of the 4th International Conference on Automatic Face and Gesture Recognition, 28-30 March, 2000, Grenoble, France, IEEE Computer Society, pp. 202-207.

Appendix A: Additional Contributions

The following research was carried out at ATR Media Information Science Laboratories, Kyoto, Japan, while the author was working there as an intern researcher.

A.1 Overview

A vision based interface for cursor control by head movement is presented. A face detection and tracking system was modified to map the tip of the nose of the user to the cursor position. Another vision-based algorithm allowed the user to enter a click by opening the mouth. The system was evaluated using the ISO 9241-9 international standard techniques for testing input devices. The Fitts' law information throughput rate of cursor movements was measured to be 2.0 bits/sec. Results of a usability assessment based on the same standard are also reported and discussed. The interface was used together with Dasher, a software system that can be used to enter text using cursor movements, as a hands free text entering application, and the results were studied. A typing speed of 7-12 words/minute was measured, depending on the level of user expertise. Performance of the system is compared to a conventional mouse interface.

A.2 Background

At present, the interfacing between computers and their users is dominated by keyboard and mouse. There is a growing interest in alternative input devices due to a number of reasons. Both keyboard and mouse requires extensive use of hands and

fingers, and therefore are difficult to use for people with disabilities related to hands/fingers. The human gestural capability is not restricted to hand movement, so it is possible to make use of this capability by developing devices that can interface these gestures and thereby enhance human computer interaction. Another advantage of alternative input devices is that they allow the hands to be used for some other purpose while allowing interaction with the computer.

Head movement has some advantages over other gestures when it comes to interacting with a computer effectively. Head movement is independent of the movements of limbs. Moreover, head movement remains possible for most disabled persons. Therefore it can be considered a means of data input. These data can be text, graphics, or graphical user interface control. Numerous researches have studied head movement [7] and its use as a human-computer interface [1, 2, 8, 9].

Our main interest is on using the head movement for cursor control. For this, the three dimensional head movements have to be tracked and mapped to two dimensional cursor movements on the screen. This can be facilitated using wearable devices, infrared beams or computer vision. Wearable devices are intrusive to an ordinary user and expensive at their current state. Eye trackers based on infra red sources [12] are expensive, though they do not require wearables or markers. With the availability of powerful hardware and easy-to-install PC cameras at relatively low prices, the use of vision based interfaces have highly prospective. Some of the systems use markers on the head/face for head tracking. The *Nouse* system [4, 5] tracks the tip of the nose and uses it to control the cursor. However, this system needs initialization for nose detection, making complete hands-free operation impossible.

Another problem associated with Nouse is the fact that it maps the displacement of nose in the input image to the velocity, not displacement, of cursor movement in the same direction, making input tasks like drawing extremely difficult.

The remainder of this chapter presents the development and evaluation of a vision based interface for cursor control using head movement. Head movement is used to control the cursor position while clicking can be performed by opening the mouth. Section A.3 describes the algorithms used in the system. Section A.4 describes the procedure and results of the performance evaluation. A brief description of the usability assessment is contained in Sections A.5 and A.6. Section A.7 presents the results of using the system to interface to the text entry software called Dasher [11] to create a hands-free text entering system, and compares the predicted and measured typing speeds. Section A.8 contains a brief description on using the system for figure drawing. Sections A.9 and A.10 present the conclusion and possible future directions respectively.

A.3 Approach

Figure A.1 outlines the functionality of the system. Images captured using the camera mounted on the computer are used to detect the user's face. Face detection is initiated by blinking. The tip of the nose is subsequently detected. The movement of the nose tip is tracked and mapped to the movement of the cursor. The mouth region is found and tracked to detect whether the mouth is opened and clicking of the left mouse button is simulated by opening the mouth. The following sub-sections describe these functions in detail.

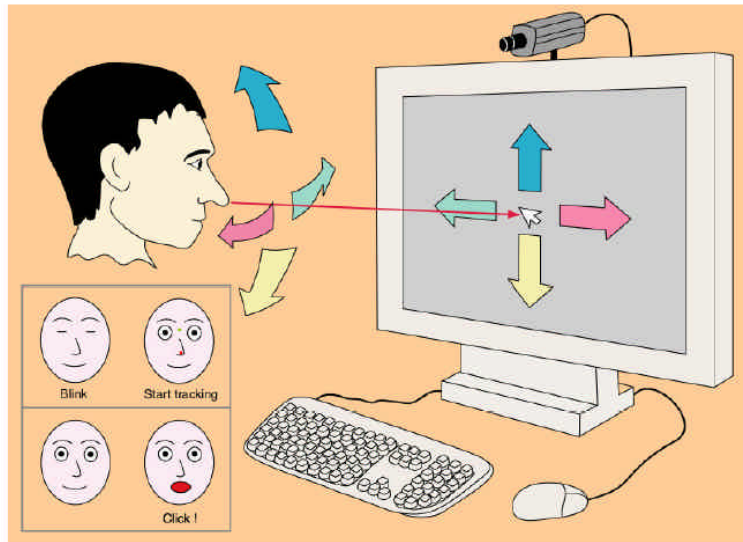


Figure A.1: Schematic of the Face Tracking Interface.

A.3.1 Detecting and Tracking the Eyes

Detection of the eyes is based on blink detection. This eliminates the need of initialization, and makes the system independent of skin color and lower face features that cause problems in color-based face detection systems. A modified version of a system that was previously developed by our group has been used. First, the difference image between the current frame and the previous frame is calculated and thresholded to extract pixels corresponding to the user's movements. Then, head movement is estimated and cancelled out so that only eye movement remains in the resulting image. Blinking of eyes causes a symmetric pattern in this image. After connected component labeling, candidate patterns are selected and validated against a set of geometrical constraints (size, distance and alignment). This is possible under the assumption that the user is sitting in front of the computer while the camera is fixed just above the monitor. If a pattern satisfying the conditions is found, a face is detected. If there are multiple patterns, the best match is selected.

However, blinking does not result in a pattern that is suitable for tracking a face. Because of this, a “Between-the-Eyes” template is used to track the location of the face. This region of the face has a distinctive pattern; a relatively bright part at the nose-bridge and relatively dark parts at the eyes like wedges on both sides. Moreover, the region is relatively stable for changes in facial expressions. Therefore, face tracking is based on tracking this region and then locating the eyes by searching in a small area in relation to this region. To ensure accurate tracking under illumination changes due to face motion or background lighting, the between-the-eyes template is updated for each frame.

A.3.2 Detecting and Tracking the Nose Tip

After the eyes are located, the tip of the nose is detected. This is located in a small region in relation to the eyes as shown in Figure A.2; therefore the search area is much smaller compared to the full image. The nose tip is convex shaped and appears bright when compared to the other regions of the face in this search area. Because of its approximately spherical shape, this bright point is relatively stable under head movement if the lighting is fixed. In the light of the above observations, the brightest point in the nose tip search area is selected as a candidate for nose tip. This is validated for approximately equal distance from the eyes. If this criterion is fulfilled, the nose tip is tracked in subsequent frames using an updating template; that is a small rectangular region around the detected nose tip. In subsequent frames, the best matching point with the template is searched around the previous position. Then the nose tip is registered again to the brightest point in a very small region around the matching point. Thereafter, the nose tip template is updated. If the template is found to lie out of the nose tip search area, nose detection is performed again. Figure A.3

shows the results of the face tracking algorithm, indicating positions of the detected eyes and nose tip.

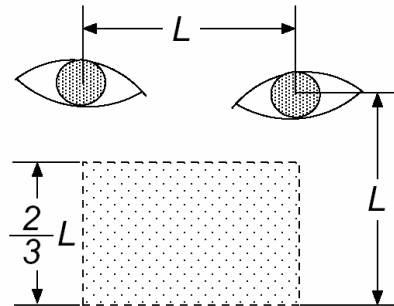


Figure A.2: Nose Tip Search Area Relative to the Eyes.

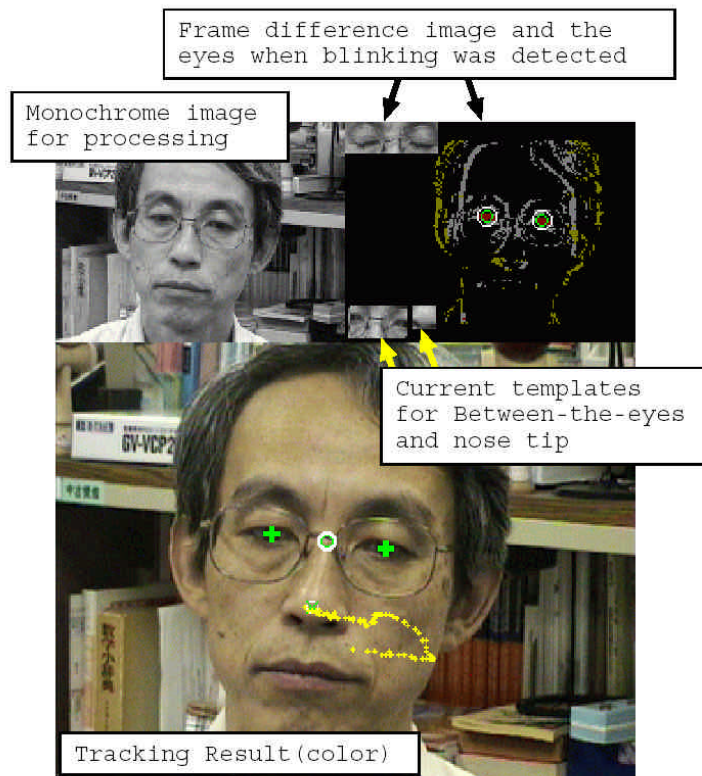


Figure A.3: Detecting and Tracking Points Corresponding to Between-the-eyes, Eyes, and Nose Tip.

A.3.3 Mapping Nose Tip Movement to Cursor Movement

At this point it is necessary to match the nose tip, which is specified in image coordinates, to the cursor position, which is specified in screen coordinates. We use a simple mapping scheme which needs very little calibration. The initial position of the nose tip, once detected, is mapped to the center of the screen. Assuming that a user looks faces the monitor and blinks when he starts using the system; this provides a quite comfortable mapping. However, it is possible to adjust the cursor position if the initial matching is found to be offset.

The displacement of the nose tip from its initial position is mapped to the displacement of the cursor from the center of the screen. Setting of appropriate gain between these two displacements is very important. Too low a gain will result in neck fatigue as a large amount of neck movement is needed to reach the corners of the screen. On the other hand, if the gain is too high, cursor control will be more difficult. Also, jitter in cursor position (due to noise in images) increases with high gain. We adjusted the horizontal gain such that it is possible to cover the entire screen without uncomfortable neck movements. Since extension/flexion (looking up/down) requires more effort compared to rotation of the head (looking left and right), we set a ratio of 1:1.4 between the vertical gain and the horizontal gain.

To reduce the jitter in cursor movement, the cursor coordinate is refined by temporal low pass filtering. This is performed by taking a weighted average of the 8 most recent cursor positions.

A.3.4 Using the Mouth to Click

We use the gesture of opening the mouth as a click of the left mouse button. The algorithm we used for this is a simplified version of a system that was designed to use mouth movement as an analog input to a computer. For this, the first step is to locate the region of the image corresponding to the mouth region of the face. An initial approximation is made by using the results of face detection, that is, the positions of the eyes and nose. To ensure that the mouth region is extracted accurately under different head rotations, this has to be refined further. The local intensity minimum present beneath the upper lip is used for this purpose.

The second step is to detect open mouth. Since open mouth is a cavity, it appears as a darker area than the rest of the face, under most lighting conditions. Also, the amount of red color present in this region is relatively high. Based on these observations, we evaluate the percentage of pixels that have intensity below a given threshold and red component above another threshold. If this percentage is found to be higher than a predetermined value, a single mouse click event is sent. Keeping the mouth open or closing the mouth does not have an effect.

A.3.5 Implementation

We used a *Sony DFW-V500* digital camera with a *Firewire* interface for image acquisition. The reason for using the Firewire interface was to ensure a high data rate from the camera to the system. The system can be configured to run with a USB camera without a problem. However, it is desired to have a frame rate of at least 15 frames per second, which some of the low end USB cameras are unable to provide.

The system was written using C⁺⁺, and runs at 30 frames per second on a PC with an *Intel Pentium III 850 MHz* processor. CPU utilization is 34% until face is detected, decreases to 22% afterwards. Memory usage is around 8-10 MB.

A.4 Performance Evaluation

We decided to evaluate the performance of our system as a pointing device. This is very useful for three main reasons. If it is possible to obtain a quantitative measure of usability for this system, it is possible to compare it with other input devices. Weaknesses of the system, if any, can be found so that improvements can be made. Strong points of the system can be detected to identify prospective applications. The remainder of this section describes the performance evaluation we conducted and the results of the same.

A.4.1 The ISO 9241-9 Standard

The ISO 9241-9 standard [6] defines the requirements and performance evaluation techniques for non-keyboard input devices. This defines the *throughput*, in bits/second, as a performance index for these devices. The calculation of the throughput is based on *Fitts Law* for moving to a target [3]. Procedures and formulae for calculating the throughputs for different tasks like tapping, and tracing are defined here. Also included in the standard is a usability assessment questionnaire. We selected the multi-directional tapping task defined in the standard and adopted the usability assessment questionnaire for evaluating our system.

A.4.2 Multi-direction Tapping Task

Figure A.4 illustrates the ISO 9241-9 task [6] as implemented in our experiments. A 240 pixel diameter circle was displayed at the center of a 640x480 pixel resolution monitor. Seventeen circular targets, each with a diameter of 21 pixels, were spaced equally around the perimeter of the circle. Subjects were required to move the cursor from one target to another, in the opposite side of the circle, according to the sequence indicated by arrows in the figure. Subjects pressed the space bar to indicate reaching a target, and the next target was highlighted by changing its color to red.

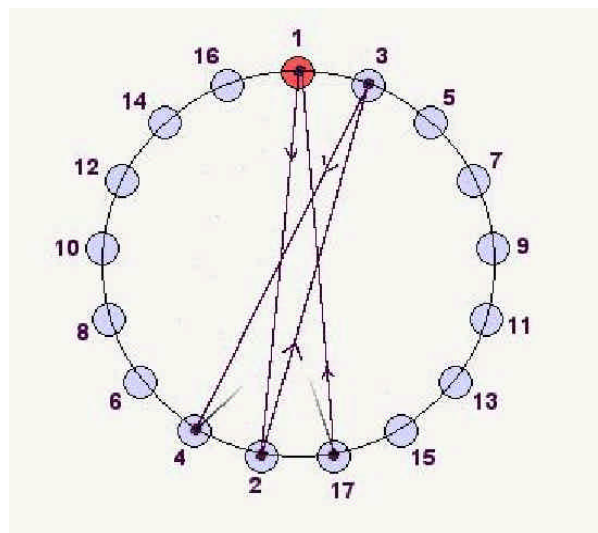


Figure A.4: Multi-directional Tapping Task.

A.4.3 Experimental Procedure

Eight volunteers, who were regular computer users, participated in the experiment. None of them were involved in the development of the system, or had used it before the experiment. Each subject was briefed about the task at the beginning of each experiment. Since we wanted to observe the effect of learning to use the device, warm-up trials were not allowed.

Each test subject performed a total of 20 repetitions(hereafter referred to as *blocks*) of the multi-directional pointing task, alternating between using the mouse and the face-tracker to control the cursor, starting, in all cases with the mouse since the users are experienced mouse users. Breaks were allowed between blocks. Total time for completion of all 20 blocks was about 40 minutes. To calculate throughput values, pixel coordinates of the cursor at time of target selection were recorded, together with time taken to reach it.

A.4.4 Results

Figure A.5 shows sample trajectories for one user completing one block of the experiment. It is evident that there are some shaky movements present in the cursor trajectories for the system. This is due to both the jitter in the movement and possibly head movement behavior too. Figure A.6 shows the variation of movement times, averaged over the eight subjects, for successive blocks. Learning effect of the task is shown by the decrease of movement time with block number. Lower values for movement time were observed for mouse while face tracker exhibited a better learning effect, especially for the first three blocks. Neither the mouse nor the vision-based face tracker shows significant orientation dependence of movement times. The mouse throughput, averaged over our last five trials, was 4.7 bits/sec, which is similar to the value of 4.9 bits/sec measured recently also using the ISO task [6]. For the face tracking pointer, the average throughput was 2.0 bits/sec. This value exceeds the 1.8 bits/sec reported for a joystick, but is lower than the 3.0 bit/sec for a trackball and the 2.9 bit/sec for a touchpad, measured previously using the ISO task [8]. Individual user mouse and nose pointer throughputs were not significantly correlated (Pearson $r = -$

0.05; Spearman's $\rho = 0.05$), suggesting that there is no strong relationship between motor skills for using these pointing devices. In contrast to findings with a head-worn head tracking system [9] the vision-based system we studied showed no significant dependence of throughput on movement direction orientation, as shown in the polar plot of throughput versus orientation, as may be seen in figure A.7.

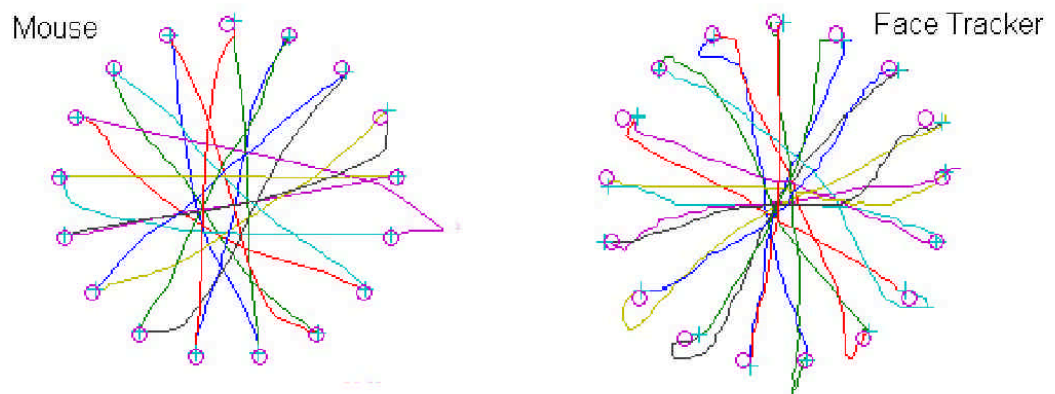


Figure A.5: Sample Trajectories for the ISO Standard Multi-directional Tapping Task.

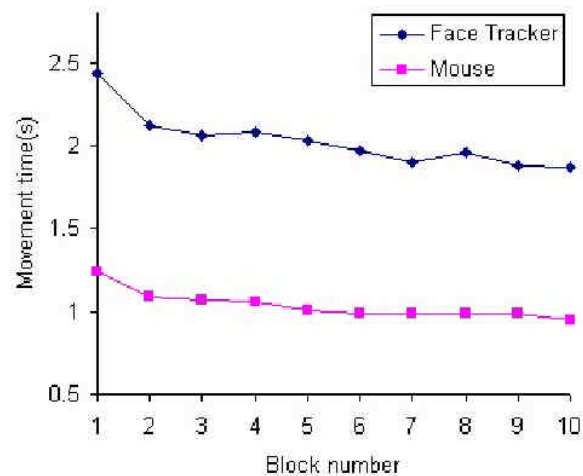


Figure A.6: Learning Curves.

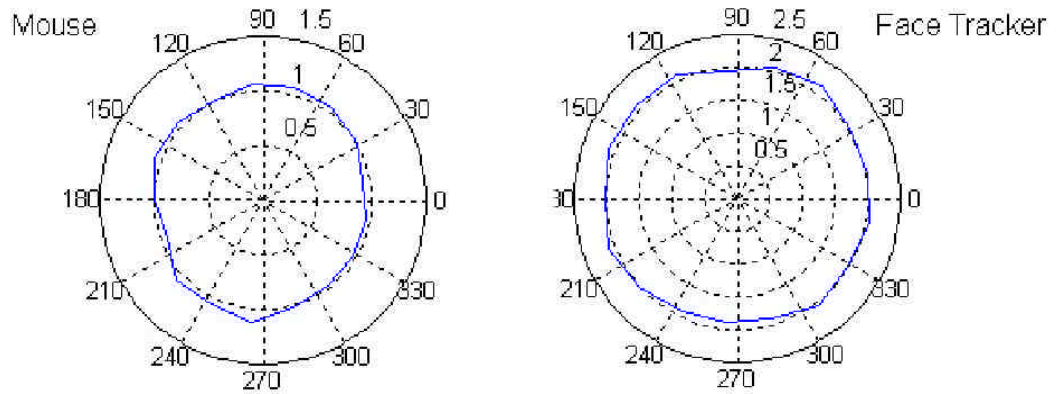


Figure A.7: Average Movement Time (sec) Versus Orientation (deg) for the ISO Standard Multi-directional Tapping Task.

A.5 Usability Assessment

A usability assessment was conducted together with the ISO tapping task. The eight subjects answered a questionnaire adopted from the ISO 9241-9 standard, rating the system on eight different criteria. The rating had a seven-point scale with 1 representing the worst rating and 7 the highest. A summary of this rating is presented in Table A.1.

Table A.1: Summary of Responses to the Usability Assessment Questionnaire

Criterion	Response mean	Mode	Range
Strength required	4.5	5	3-5
Smoothness	3.9	(2,3,6)	2-6
Effort required	4.0	(3,5)	3-5
Accuracy	3.5	(2,3,4,5)	2-5
Speed	4.1	(2)	2-7
Comfort	4.2	(4)	3-5
Fatigue	4.0	(4)	2-6
Overall	4.9	(6)	4-6

In addition to the above, Overall neck effort, when rated on Borg's 11 point scale [6], was found to be 3.7 (3=weak, 4 = moderate), with the responses ranging from 0 to 6 with a mode of 4.

The above responses indicate that all of the average responses are close to the middle of the scale. The strong points of the system are the overall usability and low strength. The main weaknesses are the lack of smoothness and accuracy.

A.6 Descriptive User Feedback

The users were asked two questions for which they were supposed to provide descriptive answers. The questions and the answers obtained are shown below. The number in brackets in front of each answer indicates the number of subjects who provided the answer.

- First question: "What are your suggestions for improvements?"
 - greater smoothness (3)
 - greater accuracy (2)
 - more displacement gain (2)
 - less displacement gain (1)
 - velocity rather than position control (1).

Most users suggested improvements in smoothness and accuracy of the pointing device, in agreement with the results from the first part of the questionnaire, listed in the previous section.

- Second question: “How would you imagine the system being used?”
 - interface for the disabled (5)
 - for use as a dual pointer (2)
 - interface for computer games (2).

A.7 Hands-Free Text Entry

Text entry to an electronic device is a task that involves a large amount of hand movement, making it a difficult task for disabled users. The Dasher system [11, 12] is a text entering system based on 2 dimensional movements using a pointing device. The user moves the mouse cursor towards letters arranged vertically in alphabetical order. The speed can be controlled by the horizontal displacement of the cursor. When a letter is selected by reaching it, subsequent letters appear within regions proportional to their conditional probabilities so that it becomes easier to select more common words. These probabilities have been learnt into the system by training it with a large amount of text. Our objective was to test Dasher with the system we designed, to evaluate its usability for hands-free text entry. Figure A.8 show a screen capture of Dasher being used together with our system to enter text.

Dasher can provide an average typing rate of 90 characters per minute using mouse, that is, a pointing device with a throughput of 5 bits per second. Based on this observation, we predicted that the system should be able to enter text at a rate of 38 characters per minute. To test whether we can achieve the predicted speed, we conducted an experiment using two subjects who were familiar with using the system.

entry is needed. Figure A.9 illustrates a simple drawing made by one of the authors. It was observed that jitter in the cursor position made drawing a bit difficult, but the drawings looked fairly similar to those done using the mouse. However, it was also observed that being able to look at the drawing directly while drawing is an advantage over using a tablet, and having no hard ‘pen’ or ‘brush’ did not cause a problem.

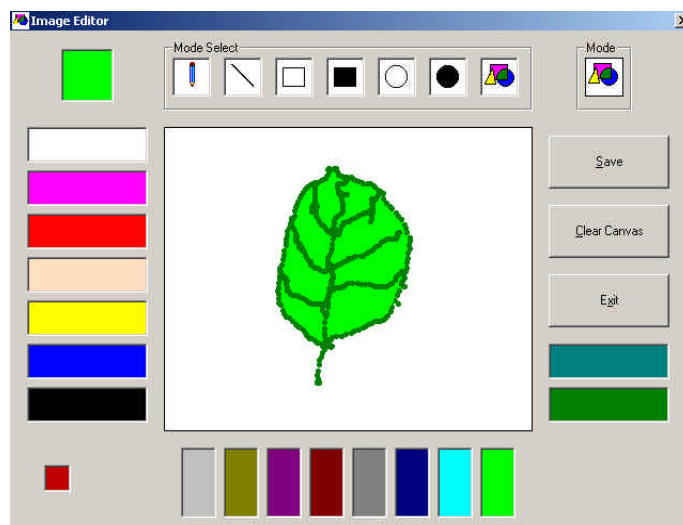


Figure A.9: A Drawing Created Using Head Movements

A.9 Conclusion

We have developed a vision-based interface that allows the user to control cursor position by pointing with the nose and to enter single clicks by opening the mouth. No special hardware is required, other than for a Firewire or a USB camera. The system initialization and operation is completely hands-free, although novice users sometimes require minor calibration. Using the international standard method for evaluating pointing devices we measured the information throughput of the system and found it to be lower than a mouse but slightly higher than a joystick. Use of Dasher together with the system demonstrated how the system can be used directly for hands-free text entry. With the measured throughput, we were able to accurately

predict the typing rate for using Dasher with our system. Although there was no systematic evaluation, the system was used for drawing simple pictures and it was observed that the appearance of the images was similar to those drawn using mouse.

A.10 Future Directions

From the trajectories of cursor movement and user feedback, it is evident that the performance of the system can be improved further by reducing jitter present in the nose tip coordinate. Evaluation of the performance of mouth clicking will be useful. Functionality can be improved by incorporating other forms of inputs such as double clicks and drag-drops. Further applications will also be considered. Another important future direction is to evaluate the system with disabled users, to whom the system will be more useful than for others.

A.11 References

- [1]. T. Darrell, N. Checka, A. Oh, and L.P. Morency, “Exploring Vision-Based Interfaces: How to Use Your Head in Dual Pointing Tasks”, *MIT AI Memo 2002-001*, 2002.
- [2]. J.W. Davis and S. Vaks, “A Perceptual User Interface for Recognizing Head Gesture Acknowledgements” *Proc. PUI’01*, 2001
- [3]. P.M. Fitts, “The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement,” *Journal of Experimental Psychology*, 47, pp. 381-391, 1954.

- [4]. D. O. Gorodnichy, "On Importance of Nose for Face Tracking," *Proc. FG'02*, pp. 188-193, 2002.
- [5]. D.O. Gorodnichy, S. Malik and G. Roth. Nouse 'Use Your Nose as a Mouse' - a New Technology for Hands-free Games and Interfaces, *Proceedings of International Conference on Vision Interface (VI'2002)*, pp. 354-361, Calgary, May 27-29, 2002.
- [6]. ISO 9241-9:2000(E), "Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs), Part 9: Requirements for Non-Keyboard Input Devices," International Standards Organization, 2000-02-15.
- [7]. R. J. Jagacinski and D. L. Monk, "Fitts' law in two dimensions with hand and head movements," *Journal of Motor Behavior*, 17, pp. 77-95, 1985.
- [8]. I. S. MacKenzie, T. Kauppinen, and, M. Silfverberg, "Accuracy Measures for Evaluating Computer Pointing Devices" *Proc. CHI'01*, pp. 9-16, 2001.
- [9]. R. G. Radwin, G. C. Vanderheiden, and M. L. Lin, "A method for evaluating head-controlled computer input devices using Fitts' law," *Human Factors*, 32, 423-438, 1990.

- [10]. J. A. Schaab, R. G. Radwin, G. C. Vanderheiden, and P. K. Hansen, "A Comparison of Two Control-Display Gain Measures for Head-Controlled Computer Input Devices", *Human Factors*, 38(3), pp. 390-403, 1996.
- [11]. D. J. Ward, A. F. Blackwell, and D.J.C. MacKay, "Dasher - a Data Entry Interface Using Continuous Gestures and Language Models", *Proc., UIST'00*, pp. 129 - 137, 2000.
- [12]. D. J. Ward and D. J. C. MacKay, "Fast hands-free writing by gaze direction", *Nature*, 418, p. 838, 2002.