Computer Vision System for Head Movement Detection and Tracking

by

Anne Lavergne

B.Sc., Simon Fraser University, 1991

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE STUDIES

(Department of Computer Science)

we accept this thesis as conforming
to the required standard

The University of British Columbia

August 1999

In presenting this thesis/essay in partial fulfillment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying for this thesis for scholarly purposes may be granted by the Head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

OCT. 27 1999
Date

Computer Science
The University of British Columbia
2366 Main mall
Vancovuer, BC
Canada V6T 1Z4

ABSTRACT

The increased popularity of applications requiring head tracking, such as teleconferencing and virtual reality, have fuelled research efforts to provide computer vision solutions to the problem of real-time head movement tracking. The attractiveness of this type of solution rests on the fact that head tracking can be performed without the use of expensive and cumbersome physical devices.

We propose a computer vision approach that detects the head movements of a user seated at a computer workstation. We model head translation and head rotation using distinct sets of templates synthesized from an initially captured image of the head and representing this head in various positions (and sizes) and orientations. Using correlation-based template matching, we achieve detection by correlating these sets of templates against each image of the head captured by a camera positioned on the top of the monitor. The best-correlating template from the set modelling head translation and the best from the set modelling head rotation represent good approximations of the three-dimensional position and orientation of the head in the scene, respectively. We improve on these approximations by defining two functions that interpolate the correlation scores of each set and by obtaining the minimum of each of these functions. We use these two minima, which represent head position and orientation, to synthesize a new template based on the initially captured image of the head. This new synthesized template represents the image of the head that most closely approximates the head position and orientation in the scene. Head movement tracking is performed by comparing the closest approximation of the head found in two consecutive images of the scene.

We have implemented our head movement tracking approach and found our system to track head position, on average, to within one pixel of the measured head position in both the x- and y-axis directions, and to detect head size (width and height), on average, to within one and two pixels of the measured head width and height, respectively. Our head movement tracking system tracks head rotations, on average, to within 1.4° of the measured angles. Our tracker is capable of processing up to eight captured images per second.

TABLE OF CONTENTS

ABSTRACT ii

TABLE OF CONTENTS iii

LIST OF TABLES vi

LIST OF FIGURES vii

ACKNOWLEDGEMENTS viii

CHAPTER I INTRODUCTION 1

Head Movement Requirements 2

Axes of Motion 2

Additional Requirements 3

Overview of the Approach 4

Overview of the Results 5

CHAPTER II PREVIOUS WORK 7

CHAPTER III APPROACH 11

Definitions 11

Modelling Head Movement 12

Modelling Head Translation along the z-axis 12

Modelling Head Translation in the Image Plane 13

Modelling Head Rotation in the Image Plane 13

Complete Model 13

Correlation-Based Template Matching 15

Template Matching 15

Correlation Function 15

Advantages of Template Matching 16

Disadvantages of Template Matching 16

Scale and Rotation Variance of Template Matching 16

Detection of Head Pose 17

Interpolation 18

Shifting 20
Tracking 22
CHAPTER IV ALGORITHM 23
Initialize 25
Capture Frame 25
Capture Master Frame 26
Create/Update Core Template 26
Restore Saved Model 26
Create/Update Model 27
Scale 27
Rotate 27
Filter and Resample 29
Convert to Greyscale 29
Save Model 29
Order of Subprocesses 30
Track 30
Detect Head Size 31
Zone of No-Motion 33
Detect Head Rotation 33
Zone of No-Rotation 34
Shifting 34
Synthesize New Core Template 36
Locate New Core Template 36
Display 36
Failure and Recovery 38
CHAPTER V RESULTS 40
Head Movement Detection 40
Tracking Results 41
Head Movements 42
1) Head Translation in Image Plane 42
2) Head Rotation in Image Plane 42
3) Head Translation along z-axis 43
Failure 43
Processing Time 44

Spatial Accuracy 45
Translation 45
Rotation 47
Movement in the Background 48
Small Changes in Facial Expression 48
Occlusion 48
Lighting Conditions 49
Camera Noise 49
Object Tracking 50
CHAPTER VI CONCLUSIONS AND FUTURE WORK 62
Rotation in Depth 63
Calibration 63

BIBLIOGRAPHY 64
APPENDIX I TRACKING SYSTEM DEMO 67

LIST OF TABLES

Table 5-1 Tracking Processes with Associated Times 45
Table 5-2 Tracked and Measured Portrait Positions and Dimensions with Associated Errors .. 46
Table 5-3 Tracked and Measured Head Rotation Angles with Associated Errors 47

LIST OF FIGURES

Figure 1-1 Axes of Motion 2
Figure 3-1 Head Movement Model 14
Figure 3-2 Interpolation Process 19
Figure 3-3 Shifting Process 21
Figure 4-1 Head Movement Tracking System Setup 23
Figure 4-2 Control Flow Diagram of Head Movement Tracking Algorithm 24
Figure 4-3 Master Template Capture Process 25
Figure 4-4 Rotation Algorithm 28
Figure 4-5 Correlation Windows Distribution around Head Position 31
Figure 4-6 Displayed Frame with Detected Head Position, Size, and Orientation 37
Figure 5-1 Representation of Best-Matching Template in Tracking Results 41
Figure 5-2 Tracking of Head Translation in Image Plane 51
Figure 5-3 Tracking of Head Rotation in Image Plane 52
Figure 5-4 Tracking of Head Translation along the z-axis 53
Figure 5-5 Tracking Failure 54
Figure 5-6 Tracking Failure and Recovery 55
Figure 5-7 Tracking Accuracy – Translation 56
Figure 5-8 Tracking Accuracy – Rotation 57
Figure 5-9 Tracking with Motion in Background 58
Figure 5-10 Tracking with Varying Facial Expressions 59
Figure 5-11 Tracking with Occlusion 60
Figure 5-12 Tracking of Object 61
Figure A-1 Avatar of Tracking System Demo 67

ACKNOWLEDGMENTS

I would like to acknowledge both of my supervisors: Dr. David Lowe, for his invaluable contribution and encouragement, and Dr. Kellogg S. Booth, for his financial support.

I also wish to thank J. Lang, C. Jennings, V. Summers, R. Walker, and A. Siebert for their insightful comments.

Financial support for this thesis was provided by the Institute for Robotics and Intelligent Systems (IRIS) and the Natural Science and Engineering Research Council of Canada (NSERC).

CHAPTER I INTRODUCTION

Several motivations have fuelled research efforts to provide computer vision solutions to the problem of real-time head movement tracking:

- cheaper and faster computer hardware promising real-time image processing,
- increased affordability of video devices such as cameras and video input cards, and
- the growing popularity of applications requiring head tracking, such as teleconferencing, virtual and augmented reality, computer games, surveillance, image compression, face recognition, etc.

Computer vision solutions are attractive as they promise to solve the real-time head movement tracking problem without the use of expensive and tethering physical head trackers, such as the ADL-1 and the Polhemus, even though these devices have successfully solved the real-time head tracking problem. Additionally, computer vision solutions offer flexibility as they can be used in a wider range of situations. However, since most of the computer vision solutions are based on analyzing the images of a scene, they are vulnerable to any changes affecting the content of these images. Therefore, they must take into consideration effects such as motion occurring in the background, varying facial expressions, partial or complete occlusion of the head, and changes in the lighting conditions of the scene. They must also deal with problems inherent to image formation such as camera noise. Finally, these solutions must employ carefully designed algorithms to ensure their performance in real time.

In this thesis, we investigated the problem of head movement tracking. Our goal was to design and implement a head movement tracking algorithm. Our algorithm would track the movement of the head of a user, seated at a computer workstation, by detecting the position and orientation of the head in the captured image of that scene using a two-dimensional technique, namely correlation-based template matching.

Head Movement Requirements

We required our head movement tracking system to track the following head movements:

1) head translation in the image plane, i.e., the (x,y)-plane parallel to the monitor;
2) head translation towards and away from the monitor, along the z-axis; and
3) head rotation in the image plane, i.e., about the z-axis (roll), in the clockwise and counterclockwise directions.

Head rotations in depth ("out of plane" rotation) about the x-axis (yaw) and the y-axis (pitch) are currently not dealt with by our head tracking system. However, the template matching technique, being tolerant of small head movements, allows our system to perform successfully when the head has undergone small head rotations in depth. In Chapter VI, we suggest ways of dealing with head rotations in depth.

Axes of Motion

Figure 1-1 shows the three axes of motion, from a user-centred perspective, adopted in this thesis. The *image plane* is the plane formed by the x- and y-axis.

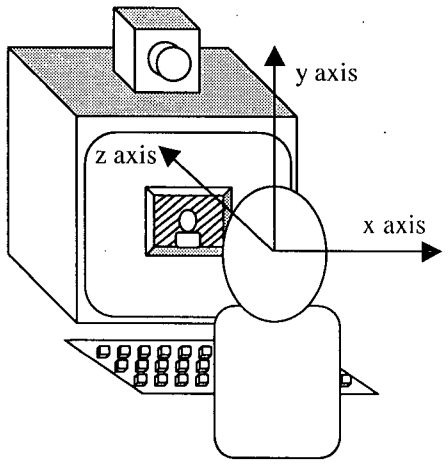


Figure 1-1
Axes of Motion

Additional Requirements

Additionally, we required our head movement tracking system to:

- **Use commonly available hardware**: No special image processing hardware was to be utilized and an inexpensive, colour static camera and video input card were to be used.[1]
- **Be non-intrusive**: The wearing of any special devices or markings was not to be required.
- **Remain responsive**: The processing time of our head movement tracking system was to remain fast enough so that there would be no noticeable lag time, to the user, between head movements and the display of the tracking results.
- **Be accurate**: Finally, we required our head movement tracking system to detect and report head position and rotation at least as accurately as head tracking systems described in the computer vision literature.

A literature survey revealed a wide array of head trackers offering solutions ranging from the tracking of only two-dimensional head positions up to the tracking of the head's 6 degrees of freedom. The tracking processing time of these numerous types of head trackers varied from processing a few images [Chen 98] up to thirty images per second.[2] Most often, the latter processing figure was achieved with the use of special image processing hardware [Azarbayejani 93]. Few approaches stated their accuracy. Chen et al. [Chen 98] report that for large head rotation angles (35°), the discrepancy between the tracked and measured angle was small (2°) but was worsening for smaller angles. For example, for estimated angles of 15° and 2°, they observed errors of 5° and 3°, respectively. Chen et al. do not furnish an explanation for these results. For head position tracking, Graf et al. [Graf 96] report a high tracking accuracy with five erroneously detected head positions out of 100.

[1] We restrict our approach to the use of such cameras because, as they are becoming standard peripherals to commonly available personal computers, this increases the number of users that could utilize our head movement tracking system.

[2] The capture rate of a standard camera is thirty frames per second.

Overview of the Approach

In our approach, we have modelled each type of head movement, i.e., head translation and head rotation, using a separate set of descriptive templates. Before tracking takes place, we capture the initial image of the head in a *master template*. A copy of this master template becomes the *core template* of our model and is used to create the template sets. The first set is used to model three-dimensional head translations. Head translations along the x- and the y-axis are straightforward to model. However, we represent head translation along the z-axis using templates that image the head with varying sizes, since the head size in the captured images changes as the head moves towards or away from the monitor. The second set is used to model clockwise and counterclockwise head rotations in the image plane using templates that image the head with various orientations. We create the first set by scaling and the second set by rotating our core template.

We achieve detection by correlating each of these sets against the image of the head captured by a camera positioned on the top of the monitor. Amongst the templates of the set modelling head translation, the one that produces the best correlation score represents a good approximation of the three-dimensional position of the head in the scene. The best-correlating template from the set modelling head rotation represents a good approximation of the head orientation in the scene.

We improve on these approximations by defining two functions that interpolate the correlation scores of each set and by obtaining the minimum of each of these functions. We use these two minima, which represent head sizes and rotation angles, to synthesize a new core template from the master template. This newly synthesized core template represents the image of the head that most closely approximates the head position and orientation in the scene. In preparation for the processing of the next captured image, we update our model by replacing the previous core template with this newly synthesized core template and by reproducing the two sets of templates. The detection process is then repeated using the next captured image of the scene and our newly updated sets of templates.

Head movement tracking is performed by comparing the closest approximation of the head found in two consecutive images of the scene. For every subsequent image, the detection process is eased by the use of the head position and orientation found in the previous image to predict the new head position and orientation.

Contrary to several previously proposed computer vision solutions, our approach does not rely on the detection of facial features or on the use of complex tracking algorithms, nor does it make use of depth or skin information. The approach we propose is generally robust with respect to movements in the background, small varying facial expressions and partial occlusion of the head. However, it is less robust when the lighting conditions of the scene change or when greater head occlusion occurs. We have designed our algorithm to alleviate, to a degree, the effects produced by camera noise.

Our head movement tracking system graphically reports its results by overlaying a tracking box on the displayed captured image at the detected position of the head in the image plane (expressed in pixel coordinates). The size of the tracking box reflects the detected size of the head, which is a function of its position along the z-axis. Finally, at the centre of the tracking box, we draw a crosshair with the same orientation as the detected rotation angle of the head (expressed in degrees).

Overview of the Results

We have implemented our head movement tracking approach and found that our system tracks head position, on average, to within one pixel of the measured head position in both the x- and the y-axis directions. Also, our tracker detects head size (width and height), on average, to within one and two pixels of the measured head width and height, respectively. However, its performance becomes less reliable, yet still acceptable, when tracking head rotations in the image plane. On average, it tracks head rotation angles to within 1.4° of the measured angles. Our tracking system fails when there is a large amount of head rotation in depth. However, if the angle of rotation in depth is reduced as the head continues its motion, our tracking system often recovers. Our tracker, running on a 400 MHz Pentium II workstation, processes three to eight captured images per second. The amount of processing each frame requires is dictated by the type of head movement detected.

In the next chapters, we discuss some of the previous computer vision approaches that have been used to solve the head movement tracking problem. We give details of our approach in Chapter III and explain the algorithm underlying the implementation of our head movement tracking

system in Chapter IV. We report our results in Chapter V and conclude in Chapter VI, along with suggestions for future work.

CHAPTER II PREVIOUS WORK

Over the past decade, the problem of real-time head movement tracking has been the focus of much attention from the computer vision research community. In contrast to the use of physical tracking devices, computer vision solutions are less cumbersome for the user, are more flexible as they can be used in a wider range of situations, and are less costly.

One of the many purposes for the detection and tracking of head movements is to allow the user to interact with a computer. In such a situation, the user often does not have to rotate her/his head by a large amount to keep eye contact with the relatively small surface of the monitor screen, and the distance between the user and the monitor can be assumed to be approximately constant. It is therefore sufficient for the head tracking system dealing with such a situation to simply track two-dimensional head positions.

An example of such a tracker is the system developed by Rekimoto [Rekimoto 95], which detects and tracks the head position of a computer user for the purpose of implementing a fish-tank virtual reality workstation. Using image subtraction, Rekimoto's system first segments the user's head from the background in the current image, then it detects the position of the head by correlating the resulting image of the user's head against a template, i.e., a portion of the captured image selected by the user. Using the estimated head position and camera parameters, Rekimoto's tracking system updates a transformation matrix that is used to display a three-dimensional scene in the fish tank. Rekimoto's tracking is simple: assuming slow head movements, he locates the head in the current frame by searching the neighbouring area around the previously detected head position.

Our approach conceptually extends Rekimoto's tracking algorithm. While that author focused on the detection of head positions, we have incorporated the detection of head rotation in the image plane and head translation towards and away from the monitor. That is to say that we no longer assume constant distance between the user and the monitor. Additionally, we have borrowed an idea common to most template-based trackers, namely the concept whereby the user interactively

selects the initial template. The content of this template represents the "object" in the scene that the head tracking system is to track over a sequence of captured images.

Often user intervention in a head tracking system is seen as a disadvantage as it prohibits the system from fully automating the tracking process, and as it increases the possibility of errors due to invalid user-supplied input to the system. Tang et al. [Tang 98] propose a "user intervention free" solution in which the shape of a user's face is modelled by an ellipse as opposed to a user-selected template. In their approach, they used a Maximum Likelihood head detector that locates a head by maximizing the match between the edges in a pair of stereo images of the scene and an elliptical template. The head is found by determining the best two candidates that form a valid stereo corresponding pair. When the system is tracking a head, the position and size of the head found in the previous image is used to predict the size of the elliptical template to be used for the next image.

Having initially experimented with a "user intervention free" approach, we found that, contrary to the viewpoint exemplified by the approach described above, on average, the reliability and quality of the template supplied by the user most often outweighed the possibility of erroneous input into the tracking system.

Sobottka et al. [Sobottka 96] propose a solution in which the detection phase is similar to the one proposed by Tang et al. but with a novel tracking phase. Once a face has been detected, it is tracked using a "snake", a deformable contour that keeps track of the face by minimizing the distance (the snake's energy) between itself and the edge of the face. Since the accuracy of the tracking is a function of the stability in the snake's behaviour, the choice of the terms defining the snake's energy equation is crucial.

In our approach, we utilize the simple head position tracking stratagem described by Rekimoto and Tang et al. This type of tracking, based on the assumption of slow head movement, continuous in space and time, is performed by repeating the detection phase at slightly different positions in the next captured frame. These positions are constrained by the previously detected position of the head.

In the literature, it is often stated that combining various techniques can increase the robustness of a head tracking system, where the weakness of one technique may be compensated by the use

of other techniques. Graf et al. [Graf 96], for example, have developed a modular head position tracking system that combines the information obtained from different simple and fast techniques not only to improve robustness and accuracy but also to accelerate the speed of their tracking system. They find facial feature candidates by analyzing images using shape, skin colour, and motion detection. The resulting feature candidates are grouped in various sets and a search is undertaken to ascertain whether these sets of features could form a face. The multi-module tracking system developed by Darrell et al. [Darrell 98] uses depth information, obtained by correlating stereo images, to eliminate objects located in the background. They use skin colour classification for fast tracking and pattern detection along with a neural network to discriminate a face from other body parts.

Contrary to the solutions described above, our approach uses one technique, the correlation-based template matching technique, to detect the position and orientation of the user's head.

The more extensive head movement tracking systems do not limit the user's head movements to only translational movements within the image plane, but consider head rotations as well. Most of these tracking systems estimate head position and orientation by first detecting and tracking head features from one frame to the next. Part of the image can be used as features that can then be detected using correlation-based template matching [Azarbayejani 93, Heinzmann 98]. Maurer et al. [Maurer 96] use Gabor jets as visual features and the strong phase-variation Gabor filters produce to compute the features' displacements from one frame to the next. In such tracking solutions, an Extended Kalman Filter (EKF) is often used to convert the two-dimensional position of the features into estimates of three-dimensional head position and orientation. Because EKF is based on physical dynamics, it allows for predictive estimation of motion parameters. Also, its recursive nature is computationally efficient. However, it does require the use of a physical dynamic model of the motion of the head, a measurement model that relates image feature positions to motion parameters and a three-dimensional model of the head. Heinzmann et al. [Heinzmann 98], on the other hand, use an inverse affine projection algorithm and a 3-point model fitting to recover the three-dimensional pose of the head.

Chen et al. [Chen 98] propose a holistic approach where the whole image of the face is considered as opposed to various features within the face. They locate the face using a fuzzy theory-based face detector and extract the skin and hair regions. The centre and the axis of least inertia of both regions are computed and used to calculate the head orientation. They claim their

approach is more robust than the feature-based approaches because they use global information, which can be extracted in a more stable manner and is not sensitive to feature variation. However, their approach does not determine the three-dimensional position of the head.

In our approach, we do not extract feature-based information from the image. Instead, we have adopted a more holistic approach whereby we estimate head position and orientation by establishing the level of best fit between the captured image and templates, which have been scaled and rotated to represent views of the head when it has undergone certain movements.

Finally, other purposes for the detection and the tracking of head movements are face recognition [McKenna 96, Steffens 98], facial image coding (image compression) [Li 94], and facial expression detection [Essa 96]. In a face recognition system, head detection and tracking is often used as a preliminary step allowing input images to represent heads in various orientations. Once these head orientations have been determined by the detection and tracking process, the recognition system can warp these head images into a frontal head orientation and therefore make use of common face databases, which are most often composed of images of heads in the frontal orientation. This preliminary step renders the recognition system more flexible since it no longer needs to restrict the type of head orientation the system requires to achieve its goal [Maurer 95]. In facial image coding and facial expression detection systems, head movement tracking is used to determine head motion, which contributes globally to the overall face motion.

CHAPTER III APPROACH

We have developed a computer vision approach to solve the problem of head movement tracking. Our approach tracks the head movement of a user seated at a computer workstation by analyzing the images of a scene, captured by a camera positioned on the top of the computer monitor, using a two-dimensional technique, namely correlation-based template matching. In this chapter, we describe our approach.

Definitions

The description of our approach rests on the following definitions.

A *head movement* is a three-dimensional continuous and contiguous event occurring in the scene. It can be represented by a sequence of two-dimensional images of the head, which are produced by sampling the head movement as it occurs in the scene. These temporal samples represent the head in various positions and orientations during its motion. Even though a discrete sampling process has occurred, for the purposes of our approach, we consider the sampled head movement represented by a sequence of images to be continuous and contiguous. We call the image of a head in a particular position and orientation a *head pose*. A temporal sample represents a head pose.

In an effort to solve the problem of head movement tracking, we first solve the problem of detecting the head pose in each captured image of a sequence, which is defined as the problem of detecting the position and the orientation of the head. We then can solve the problem of head movement tracking by comparing these detected head poses in consecutive images of the scene and, using our head movement model, we can infer the occurrence and the type of head movement in the scene.

We model head movement by considering the various head poses it produces. We call the images representing these head poses *templates*.

Modelling Head Movement

We construct our head movement model in the following way. Before any tracking takes place, the user defines a portion of the captured image, which we name the *master template*. This template represents the initial head pose of the user[1], i.e., the initial position and orientation of the user's head before tracking takes place. This master template is not part of our model but is used to create its first template, the *core template*, and all subsequent core templates required during tracking. Therefore, this first core template also represents the initial head pose.

Since the movement the head undergoes in a scene can be composed of a translation and a rotation, we have developed our approach so as to detect these two types of movement independently by dividing our model into two sets of templates.

Modelling Head Translation along the z-axis

First, we model head movement towards the monitor, and thus the camera, by creating a template that represents the same image of the head pose found in the core template but with a larger size. This follows from the fact that, as the head moves towards the monitor, its captured image increases in size. Similarly, we model head movement away from the monitor by creating a template that represents the head pose found in the core template but with a smaller size because, as the head in the scene moves away from the monitor, its image decreases in size. These two templates, along with our core template, correspond to the three types of head pose representing head translation towards the monitor, away from the monitor, or the absence of translation, respectively. These three templates form a set that will allow us to detect if this type of head translation has occurred and its direction.

[1] During the capture of the master template and the establishment of the initial head pose, the user is not constrained to adopt a straight, frontal head pose.

Modelling Head Translation in the Image Plane

When the head moves sideways, up, or down in the scene (i.e., within the (x,y)-plane), its size in the captured image does not increase or decrease and its appearance in the captured image remains approximately the same.[2] Because this type of head movement does not affect the size or appearance of the image of the head, we can utilize one template, namely the core template, to represent such head movement. Therefore, no additional templates are required to represent head poses produced by head motion within the image plane.

Modelling Head Rotation in the Image Plane

We model clockwise and counterclockwise head rotations[3] by creating two templates: one representing the head pose found in the core template but with a clockwise rotation, and another representing that same head pose but with a counterclockwise rotation. These two templates, along with our core template, correspond to the three types of head pose representing a head rotation clockwise, counterclockwise, or the absence of rotation, respectively. These three templates form a set that will allow us to detect if a head rotation has occurred, the amount of rotation, and its direction.

Complete Model

Our model is now complete since it accounts for all the head movements we require our head tracking system to detect. We display an example of the templates comprising our model in Figure 3-1.

[2] The set of head features that the image represents remains approximately the same, even though the actual digital image representing the head has been sampled on a shifted grid.
[3] Note that all rotations in this thesis are named from a user-centred perspective.

This set of templates models head translations towards and away from the monitor. The small-sized head template represents head movement away from the monitor, the middle template represents no head translation towards or away from the monitor, and the large-sized template represents head movement towards the monitor.

This set of templates models head rotation in clockwise and counterclockwise directions. The template on the left represents a clockwise head rotation, the middle template represents no head rotation, and the template on the right represents a counterclockwise head rotation.

Figure 3-1

Head Movement Model

The set of three templates displayed in the first row is an example of a set used to detect three-dimensional head translation. The left template represents a head pose resulting from a head movement away from the monitor. The middle (core) template represents the resulting head pose when no head translation towards or away from the monitor has occurred. However, this template is used to detect head translation in the image plane. Finally, the right template represents a head pose resulting from a head movement towards the monitor. The set of three templates displayed in the second row is an example of a set used to detect clockwise and counterclockwise head rotation. The left template represents a head pose resulting from a clockwise head rotation. The middle (core) template represents the resulting head pose when no head rotation has occurred. The right template represents a head pose resulting from a counterclockwise head rotation.

Correlation-Based Template Matching

Template Matching

Template matching is a process that determines how closely a template matches a section of an image called a *window,* or *correlation window.* The set of windows considered may cover all or part of the image. How well a template matches, or the degree of similarity, is computed by correlating the template against each window [Haralick 93]. The correlation method is performed by considering each pixel in the template and its corresponding pixel in the window. The value of these pixels is used in a correlation function that produces a correlation or similarity score. There are various types of correlation functions such as sum of squared differences, sum of absolute valued differences, mean-squared differences, and cross correlation [Fua 93].

Correlation Function

In our approach, we use the cross correlation function:

$$s = 1.0 - \frac{\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}(T(i,j)W(i,j))}{\sqrt{\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}T^2(i,j)}\sqrt{\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}W^2(i,j)}}, \quad (1)$$

where T is the template and W is the window, both of size $M \times N$, and s is the resulting correlation score. We have chosen this correlation function based on the results published by Hotz [Hotz 91] and discussed by Fua [Fua 93]. In these studies, four correlation functions were investigated, namely two versions of the mean-squared difference function and two versions of the cross correlation function. The difference between the two versions of each of these functions was the added robustness designed to lessen the effects of pixel variation. Exercising each of these four functions with templates of various sizes, Fua concluded that the cross correlation function, shown above, designed without this robustness, performed as well as the more robust version of the two functions, without suffering from the additional computational cost of the added robustness.

Advantages of Template Matching

Some advantages of template matching are its simplicity [Brunelli 93] and its flexibility. Its correlation-based algorithm is straightforward to implement and it allows us to use templates that have either been created using a section of the image or that have been derived therefrom.

Disadvantages of Template Matching

On the other hand, one of the main disadvantages of template matching is its computational cost since correlating requires the "perusal" of each pixel in the template over several windows in the image. However, several cost-reducing improvements can lead to faster template matching performance. Cohen et al. [Cohen 94] show that their random "cluster search" approach to template matching is, on average, faster than deterministic schemes. Krattenthaler et al. [Krattenthaler 94] propose a "point correlation" where matching is not performed over the entire template but over a smaller set of pixels. Harvey et al. [Harvey 91] had previously suggested a sparse template technique. As will be discussed in Chapter IV, we have reduced the cost of correlation by fine-tuning various factors in our approach, namely the number of windows we consider in the captured image of the scene, their size and the size of the templates. Finally, as illustrated by Fua [Fua 93], a judicious choice of the correlation function may lead to a fast and fairly accurate tracking system.

Scale and Rotation Variance of Template Matching

Finally, we have chosen template matching because, beyond a certain scale (head size) and angle of rotation, it is scale and rotation variant. Contrary to some previous work that made use of template matching, where this variance was seen as a shortcoming [Takács 94], we count on correlation-based template matching to be able to differentiate between templates representing different head sizes and head orientations. For example, if we correlate three templates representing the head at different sizes against the image of the scene, we expect to obtain three different correlation scores, and presumably the best correlation score will be associated with the template that most closely resembles the pose of the head in the image of the scene.

We have established through experimentation the amount of head size and rotation angle variation that our detection approach requires in order to fulfill our expectation of different correlation scores. We discuss the setting of head size and rotation angle values based on this amount of variation in Chapter IV.

The correlation scores produced by the template matching technique are insensitive to the occurrence of small scale and rotation angle changes when those differences are below the size and rotation angle values we have experimentally determined as mentioned above. This tolerance allows us to deal summarily with head rotation in depth, about the x- and the y-axis. Rekimoto [Rekimoto 95] reports a slight tolerance of the correlation matching technique when dealing with tilted or scaled images, although the author does not provide any size or angular values for which this tolerance occurred. Essa et al. [Essa 96] report larger changes in viewing distance ($\pm$ 15%) and in head rotation angles ($\pm$ 10°) below which they observe the matching technique becoming insensitive.

Detection of Head Pose

To achieve our goal of detecting head poses (position and orientation), we correlate each set of templates with a set of correlation windows covering an area of the captured image of the scene. The template producing the best correlation score, from the set modelling head translation, represents a good approximation of the actual size of the head in the captured image. The best-correlating template from the set modelling head rotation represents a good approximation of the rotation of the head in the captured image. The correlation windows at which those best-correlating scores are obtained represent the location of the head in the capture image.

We use the word *approximation* when we describe the relationship between the head pose represented by the best-matching template and the head pose found in the image of the scene. The reason is that, most often, the head in the image of the scene is not quite as large, or as small or as rotated as is the head represented by the best-matching template.

Interpolation

We would like to refine our detection process so as to find the head size and rotation that would most closely match the head pose found in the image of the scene. This optimal head size (and head rotation angle) would be located within an interval of head size values (and head rotation angles) formed by the smallest and the largest head sizes (and rotation angles) that were used to synthesize the templates in the first set (and in the second set, respectively). Ideally, we could correlate a larger set of templates representing numerous head sizes (and rotation angles) within these intervals. Unfortunately, because the cost of template matching is a function of the number of templates we correlate, this solution is not feasible. Therefore, we introduce an interpolating process that allows us to consider all the head sizes and rotation angles that lie within these intervals without increasing the number of templates.

We now describe this interpolating process by presenting an example. After correlating the set of templates representing the head with three different sizes, we plot the resulting correlation scores. The x-axis of the plot represents head size values while its y-axis represents the numerical values of the correlation scores. We name the score produced by the template representing a small-sized head a ($x = 0.92$).[4] The score produced by the core template ($x = 1.0$) is named b and c is the score produced by the large-sized head template ($x = 1.08$). We derive a function that interpolates the three scores a, b and c from the quadratic interpolating function:

$$f(x) = b + \left(\frac{c-a}{2h}\right)x + \left(\frac{1}{h^2}\left(c - b - \left(\frac{c-a}{2}\right)\right)\right)x^2, \qquad (2)$$

where $h = 0.08$, the distance on the x-axis separating the head size values.

In this example, the template representing a small-sized head scored 0.021607, the core template, 0.008933 and the large-sized head template, 0.024097. Because the core template has the best (smallest) correlation score, it is a good approximation of the head size found in the scene. Figure 3-2 illustrates the resulting interpolating function and its minimum.

[4] We discuss the setting of head sizes in more detail in Chapter IV.

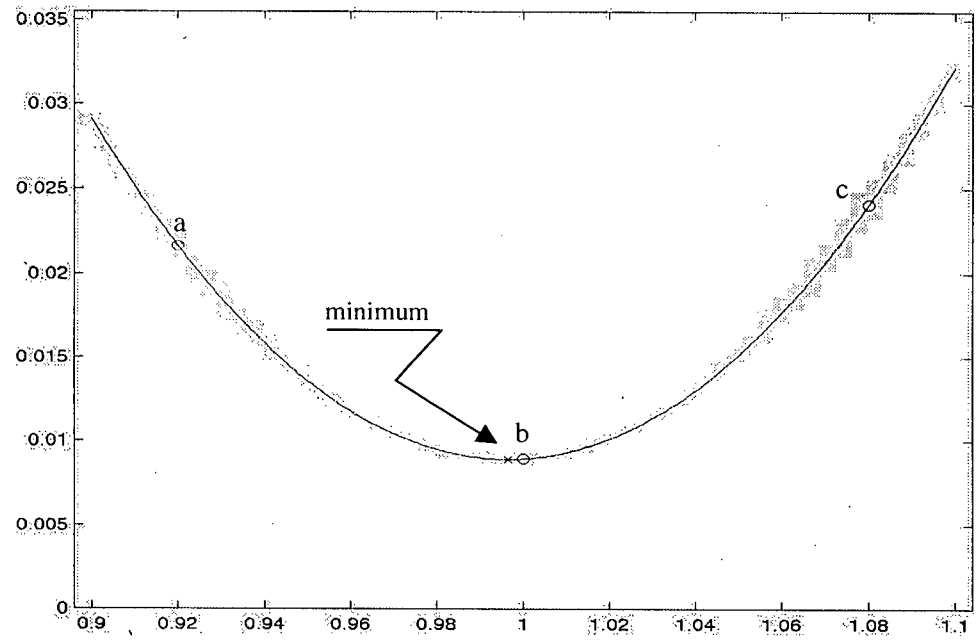


Figure 3-2
Interpolation Process

Plotting the function (2) between our three points *a*, *b* and *c* gives us a curve that represents all the correlation scores for the head sizes found within the interval (*a*, *c*). The head size (0.9964) for which the smallest correlation score is produced is the minimum of this curve and is the head size that most closely matches the head size found in the image of the scene. This minimum is found by setting the first derivative of function (2) to zero:

$$\frac{df(x)}{dx} = \left(\frac{c-a}{2h}\right) + \left(\frac{2}{h^2}\left(c-b-\left(\frac{c-a}{2}\right)\right)\right)x = 0, \tag{3}$$

then solving for x and accounting for the shifted scale of the x-axis[5] :

$$x = \frac{(a-c)h}{2(a-2b+c)} + 1.0. \tag{4}$$

A similar process is performed to obtain the head rotation angle that most closely matches the one found in the image of the scene.

[5] The origin of the x-axis of the plot is located at 1.0.

Shifting

In our example above, we expected to obtain a minimum and to find it within the interval (a, c). This only occurs when the best of our three correlation scores is the point plotted at the origin, i.e., when the best-matching template is the core template. However, it may be the case that the best correlation score is produced by the template representing the large-sized or the small-sized head. This would occur if the user had quickly moved her/his head towards or away from the monitor. In these situations, because the lesser-valued correlation score would now be located either to the right or to the left of the point plotted at the origin, the interpolated function might no longer yield a minimum within the interval formed by these scores or it might produce a maximum.

For a minimum to be produced within the interval (a, c), i.e., to ensure that the best of our three correlation scores is always the point plotted at the origin and hence that the best-matching template is always the core template at the centre position in a set, a shifting process is devised to reshuffle the templates within a set. Shifting occurs as follows: the template that produced the worst (largest) correlation score is discarded. This makes sense since this template must represent the image of the head the least similar to the one in the current scene. The core template is shifted into the discarded template's position and the other non-core template, the one that produced the best correlation score, is shifted into the core template's position. This increases the possibility of having the best correlation score plotted in the centre of our curve. Finally, to fill in the position this last template left empty, another template is synthesized to represent one of the following:

- If the best-matching template represented a large-sized head or a small-sized head, this new template will represent an even larger-sized head or an even smaller-sized head, respectively.
- If the best-matching template represented a counterclockwise-rotated head or a clockwise-rotated head, this new template will represent an even more counterclockwise-rotated head or an even more clockwise-rotated head, respectively.

By shifting templates and creating a new one, we are shifting the interval of head sizes or head rotation angles to reflect the head movement inferred by the correlation scores. This increases

the likelihood that the new interval would now include the head size or rotation angle that would best match the head pose found in the current image of the scene.

To better illustrate this shifting process, let's examine a sequence captured while the user was quickly moving her head away from the monitor. In this situation, the template representing a small-sized head produces the best correlation score. The core template represents the next best approximation of the head in the scene and the template representing a large-sized head produces the worst (largest) correlation score. The function we interpolate between these three correlation scores does not yield a minimum on the interval (a, c). We therefore shift the templates so that the best correlation score is now plotted at the centre of our curve. This is done by first discarding the template representing a large-sized head and by replacing it using the core template. The smaller-sized head template is moved into the core template position and an extra-small-sized head template is synthesized and put into the small-sized head template position. Figure 3-3 depicts this shifting process.

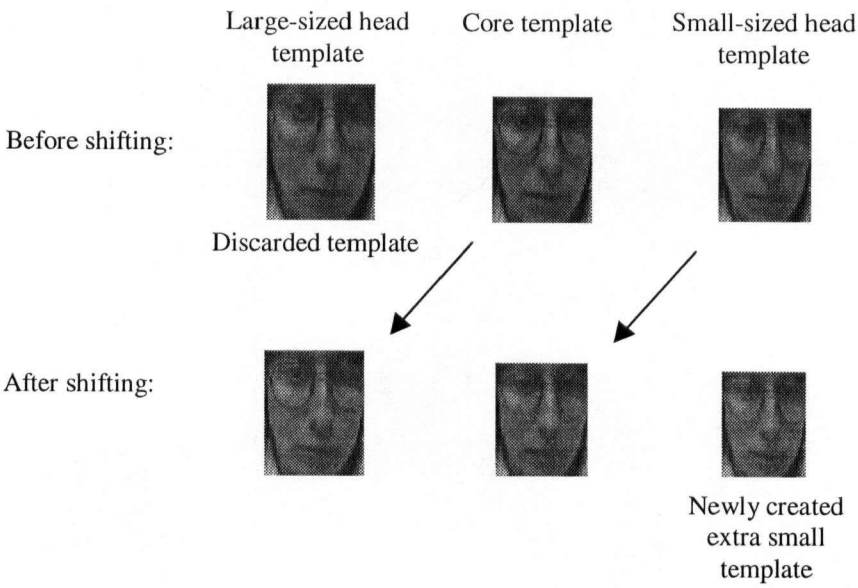


Figure 3-3
Shifting Process

The newly created template is matched against the same image of the scene and a second plot is created. If the tracking is successful, the minimum produced by this new interpolated function will be located within the interval formed by the correlation scores of the new set of templates and will represent the optimal approximation of the head size found in the image of the scene.

However, if, after shifting took place, the template located at the centre still does not yield the best (smallest) correlation score, we conclude that the tracking has failed. Tracking failure is further discussed in Chapter V.

After analyzing a captured image and having determined the best head size and rotation angle, we use these values and the master template to synthesize a new template. This newly synthesized template becomes the new core template of our model and represents the closest approximation of the head pose found in that image of the scene. We then update our model by recreating our two sets of templates using this new core template.

Tracking

To achieve our goal of tracking head movements, we perform head pose detection for every captured image in a sequence. By assuming slow, contiguous, and continuous head movements, we can predict the position and orientation of the head pose in the next image of the scene using the detected position and orientation of the head pose in the current image. By comparing consecutive head poses, we can infer, using our head movement model, the occurrence and the type of head movement in the scene.

CHAPTER IV ALGORITHM

To demonstrate the feasibility of the approach we have developed, we have implemented a head movement tracking system that tracks the head of a user seated at a computer workstation. A static colour camera, located on the top of the monitor, captures the image of the scene, i.e., the image of the user and the background. The setup is depicted in Figure 4-1.

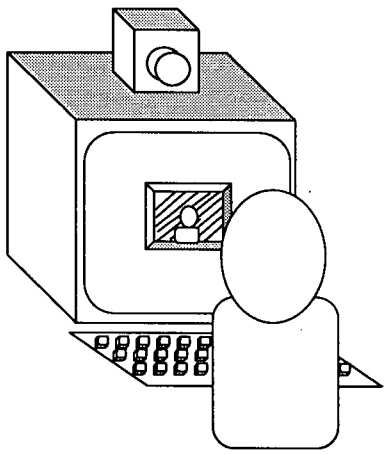


Figure 4-1
Head Movement Tracking System Setup

In this chapter, we discuss the algorithm of our head movement tracking system. The processes involved in our algorithm are depicted in the control flow diagram of Figure 4-2 and are explained below.

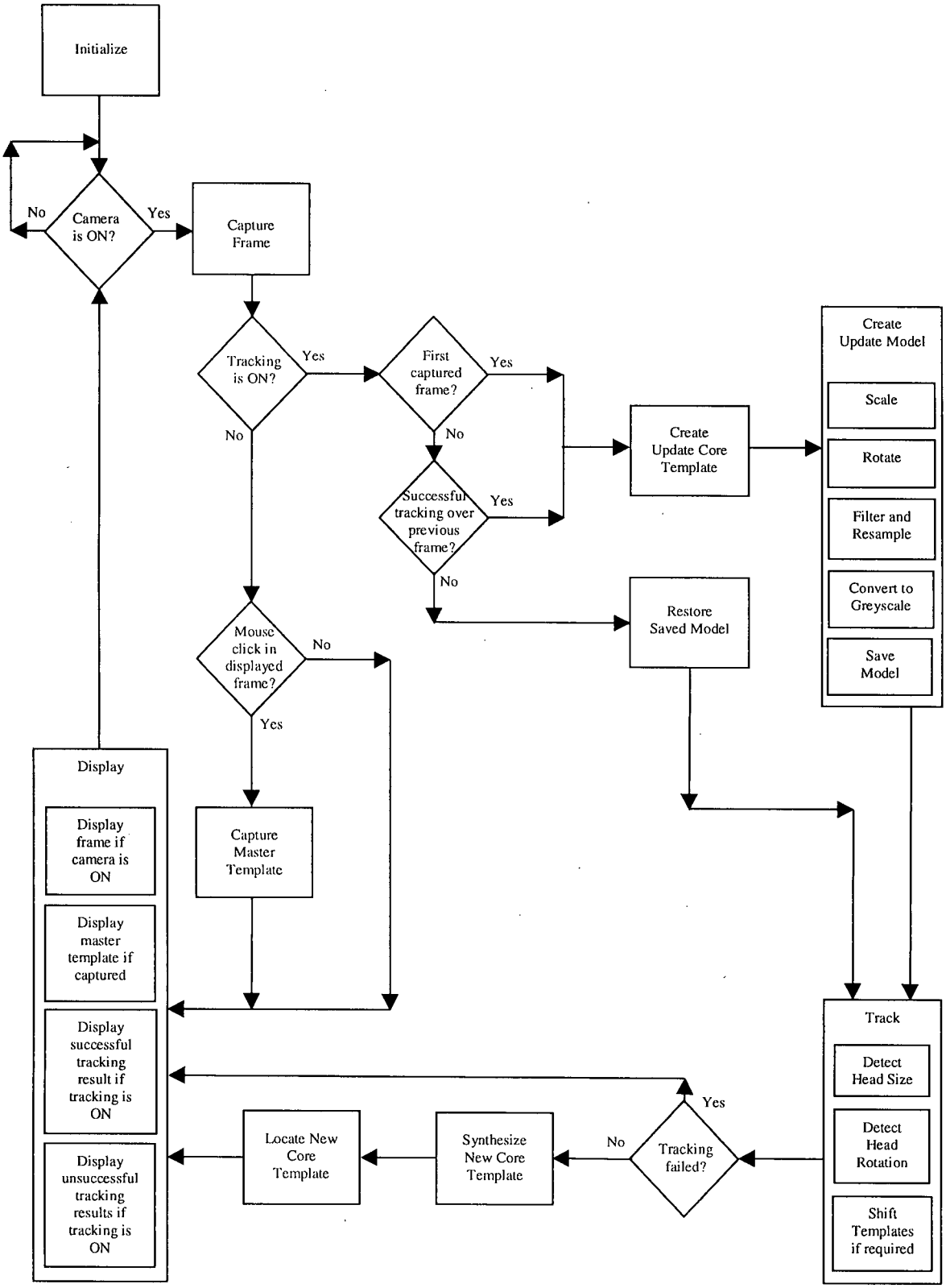


Figure 4-2

Control Flow Diagram of Head Movement Tracking Algorithm

Initialize

In the initialization process, the initial state of the head movement tracking system is defined. The camera and the tracking process are initially inactive and all the templates of the model, along with the master template, are non-existent.

Capture Frame

This process starts when the user activates the camera. Images of the current scene are captured and displayed on the monitor screen. The dimensions of a captured frame are 180 × 180 pixels. Figure 4-3 (a) shows an example of a displayed captured frame.

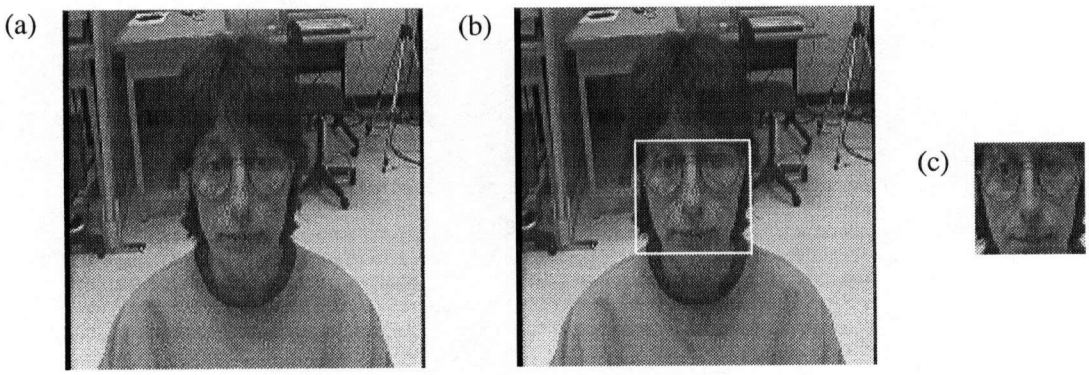


Figure 4-3
Master Template Capture Process

In our algorithm, we utilize a static camera since we are interested in tracking typical head movements performed by a user seated at a computer workstation. The extent of the user's head movements, while keeping eye contact with the computer screen, would usually be restricted to the front of the monitor and would remain, for the most part, within the field of view of the camera. This eliminates the need for an expensive active camera, typically utilized when the movements to be tracked extend beyond a static field of view. Finally, the use of a static colour camera satisfies our hardware requirement, which is to use commonly available and inexpensive hardware.[1]

[1] Inexpensive colour cameras are becoming standard peripherals when purchasing a personal computer and are therefore more readily available than the black and white or greyscale cameras.

Capture Master Template

As a prerequisite to the tracking process, the user is required to select the part of the scene that is to be tracked. This selection is done interactively by having the user draw a box around the desired part of the scene. Usually, the face is selected for tracking. The content of the box becomes the master template.

The tracking system displays the resulting master template by drawing a white box around the part of the scene the user has selected. This is depicted in Figure 4-3 (b). Figure 4-3 (c) displays the resulting master template.

If the captured template is found unsatisfactory, the user may repeat this capturing process as long as the tracking process has not yet been initiated.

Create/Update Core Template

If tracking has just been initiated, i.e., no captured frame has yet been processed, we create the core template of our model using the user-selected master template.

On the other hand, if tracking has already been initiated and has been successful over the previous frame, we update the core template of our model using the newly synthesized core template that has just been created as a result of the tracking process over the previous frame.

Restore Saved Model

However, if tracking was unsuccessful over the previous frame, the best-approximating head size and rotation angle were not computed for that frame, and consequently no newly synthesized core template was created and used to update the core template. In this situation, since we do not have an updated core template we can use to re-synthesize our model, we upgrade the latter by restoring the templates of a previously saved model. This often allows the tracking process to recover over the next frames.

Create/Update Model

Our model is composed of two sets of three templates each and both sets share the core template, which, at this point, has already been either created or updated. We build the two template sets by scaling or rotating this core template.

Scale

To produce the templates representing a large-sized and a small-sized head, we scale the core template using factors of 1.08 and 0.92, respectively, where the scaling is uniformly applied to the width and height of the core template.

The head size scale factors (1.08 and 0.92) were experimentally determined. In applying these factors, one must ensure that the produced head size does not become unrealistically large, considering the speed at which a human head can move, or too small so as to fall within the range over which template matching is scale (head size) tolerant, as discussed in the Scale and Rotation Variance of Template Matching of Chapter III.

Our scaling algorithm is a modified version of the bilinear interpolation module in the Vista library [Pope 94].

Rotate

To produce the templates representing the image of a clockwise and a counterclockwise rotated head, we rotate the core template by -10.0° and 10.0°, respectively. The head rotation angles were also experimentally determined. The constraints discussed in the Scale section above also apply in the setting of the head rotation angles.

The rotation algorithm we use is a modified version of the rotation module found in the Vista library [Pope 94] which is itself based on a multiple shearing algorithm developed by Catmull and Smith [Catmull 80] and simplified by Paeth [Paeth 86]. The Vista rotation module rotates and pads the original image, hence producing a larger image.

In our approach, we need to preserve the original size of the template once it has been rotated. To do so, we supply to the modified Vista algorithm an enlarged version of the template we wish to rotate. This enlarged template is formed by capturing a square area of the image of the scene that contains the template at its centre. We determine this area by adding approximately 80% of the template's largest dimension to its smallest sides. We then complete the square by sufficiently increasing the template's longest sides. We have found that enlarged templates of such size produce rotated templates that contain enough of the features of the original template for the correlation process to yield satisfactory results. Also, in our modified version of the Vista rotation module, the padding is no longer required.

We illustrate the various steps in the rotation process and their associated result in Figure 4-4.

(a) Original Template (b) Shear in X (c) Shear in Y (d) Shear in X (e) Cropped rotated template

Figure 4-4
Rotation Algorithm

Figure 4-4 (a) shows the template we wish to rotate. Its enlarged version is first sheared in the *x*-axis direction as shown in Figure 4-4 (b). It is then sheared in the *y*-axis direction, shown in Figure 4-4 (c), and the result is sheared again in the *x*-axis direction, shown in Figure 4-4 (d). The resulting enlarged, rotated template is then cropped producing the final rotated template shown in Figure 4-4 (e). This final template has the same size as the original template shown in Figure 4-4 (a). The cropping step does impose some constraints on the master template capture process. The salient features of the part of the image to be tracked (user's face) need to be centrally located in the captured master template. However, this constraint is most often satisfied since, intuitively, users do tend to select, as the object to be tracked, the part of the image of the scene in which the main features of their face are centred.

Filter and Resample

The scaled and rotated templates, along with the captured images, are filtered to attenuate image noise and are downsampled to speed up correlation.

We have chosen a simple yet fast filtering and resampling algorithm, in which the average of the values of four connected pixels becomes the value of the corresponding pixel in the resampled template. This has the effect of reducing the initial width and height of the scaled and rotated templates and captured images by a factor of two.

Convert to Greyscale

The image of the scene is captured and displayed as a coloured RGB image. However, to speed up all template manipulation processes, the three colour pixel values are converted to one greyscale value using the conversion formula:

$$grey = 0.299red + 0.587green + 0.114blue \tag{5}$$

where *grey* is the truncated greyscale value of the pixel, and *red*, *green* and *blue* are the respective RGB colour values of the pixel [Pope 94].

Save Model

When our model is completed, we save a copy of all its templates. This saved model is used when tracking has failed for a particular frame. In such a situation, because the best-approximating head size and rotation angle were not computed for that frame, it is not possible to synthesize a new core template and use it to update our model. Therefore, to process the next frame, we update our model by restoring this saved model.

Order of Subprocesses

We found that scaling and rotating the templates before filtering and resampling them, as opposed to scaling and rotating filtered and resampled templates, produces better correlation results even though it is more computationally expensive. We observed that filtering and resampling the templates as a last step preserved more details of the image. This increase in nuances in the pixel intensity pattern facilitates the template matching process in establishing more accurately similarity levels between a template and various correlation windows in the captured image.

Track

For the sake of readability, we shall name the templates comprising our model as follows:

- TEMPLATE: The core template of our model.
- LARGE: The large-sized head version of TEMPLATE.
- SMALL: The small-sized head version of TEMPLATE.
- CW: The clockwise-rotated head version of TEMPLATE.
- CCW: The counterclockwise-rotated head version of TEMPLATE.
- NEW TEMPLATE: The newly synthesized core template.

Once the user is satisfied with the captured master template, the tracking process can be initiated.

The tracking process is divided into two subprocesses. In the first subprocess, we detect the size of the user's head in the current frame by correlating TEMPLATE, LARGE, and SMALL against the current frame. In the second subprocess, we detect the rotation angle of the user's head in the current frame by correlating TEMPLATE, CCW, and CW against the current frame.

Since these two detection subprocesses perform independently from each other, the ordering is irrelevant. We have chosen to start with the detection of the size of the user's head.

Detect Head Size

Correlation over the entire frame would be too costly. Therefore we correlate TEMPLATE over a reduced area which we establish in the following fashion. Using the position of the head detected in the previous frame (or the position of the captured master template, if the tracking process has just been initiated) as the initial (centred) position, we consider a set of correlation windows that we symmetrically distribute around this position. Each correlation window is offset from its neighbour by one pixel in either the horizontal or the vertical direction or both. Figure 4-5 illustrates this distribution using a simple example in which we consider eight correlation windows (dotted white) distributed around the initial head position (solid white). The size of the correlation windows is set to the size of the template we correlate (i.e., the size of the centred position) which is, in the current case, the size of TEMPLATE. The window that produces the best correlation score is considered to represent the location of the head within the current frame. The number of correlation windows was experimentally established. We observed that considering eleven horizontal and eleven vertical positions (for a total of 121 correlation windows) was sufficient, in most cases, to detect the new position of the head, while keeping the computational cost of the correlation process to a minimum.

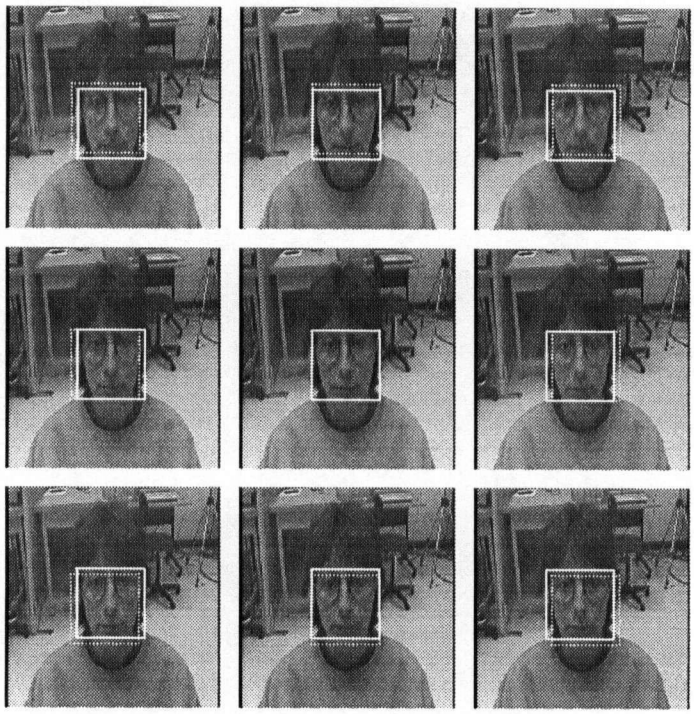


Figure 4-5
Correlation Windows Distribution around Head Position

Before performing the correlation, we ensure that all the correlation windows, surrounding the initial position of the user's head, are fully within the image. This is to detect the situation where the head is moving out of the field of view of the camera. If this occurs, the tracking stops. This situation is further discussed in the Failure and Recovery section located at the end of this chapter.

To define the area in the image over which we correlate LARGE, we use the detected location of TEMPLATE in the current frame as our initial location and symmetrically distribute a smaller number of correlation windows (five horizontal and five vertical positions for a total of 25 correlation windows) around this location. We use a reduced number of correlation windows because we already know the approximate position of the head in the current frame. Our purpose in correlating LARGE against the current frame is to determine the best size of the head in this frame at that approximate location. For the same purpose and using the same initial head location found while correlating TEMPLATE, we repeat this process using SMALL. In both cases, the size of the correlation windows is set to the size of the template being correlated. We correlate LARGE and SMALL over a small number of correlation windows, as opposed to correlating them over only one window, i.e., the window identifying the approximate head position in the current frame. Since this approximate location was obtained by correlating TEMPLATE and the head size might actually be larger or smaller, this position could be inaccurate.

Originally, we thought of defining the placement of the correlation windows as a function of the perceived direction of the head movement. We believed, but did not test, that this scheme would lead to a more efficient (i.e., less costly) template matching process. In this scheme, not only would we have positioned the windows in the area of the frame where we expected the head to be located, i.e., focusing on one direction, but this scheme might have reduced the number of windows over which we correlated TEMPLATE, depending on the detected speed of the head. However, using the current scheme we have been describing in this thesis, we have observed that tracking often recovers partly due to the fact that the area covered by this larger number of windows is symmetrically centred on the previous head location. This increases the potential of finding the new head position since it may be now located on either side of the previously detected head position.

Upon correlation, we ensure that TEMPLATE has produced the best correlation score and shift the templates if this is not the case. We then interpolate our three correlation scores and compute the minimum of the resulting function, which represents the best approximation of the size of the head found in the current frame.

Zone of No-Motion

We initially observed that, when the user keeps her/his head immobile, often the detected head size would not remain constant. These jittery results, due to camera noise and rounding-off of pixel values, were alleviated by defining a zone of *no-motion* about the origin of the plot. If the minimum of the interpolated function falls within this zone of no-motion, we simply conclude that the size of the user's head in the captured image has remained more or less unchanged. Therefore no approximated head size for the current frame is computed and the previously approximated head size is kept. If the minimum falls between this zone and the correlation score of SMALL, we conclude that a decrease in head size has occurred, implying a possible head movement away from the monitor. On the other hand, if the minimum falls between the zone of no-motion and the correlation score of LARGE, we conclude that an increase in head size has occurred, implying a possible head movement towards the monitor. In both cases, a new approximation for the user's head size in the current frame is computed. The interval covered by the no-motion zone was experimentally defined to include all head sizes from 98.5% to 101.5% of the size of the head imaged by the core template. This represents a head scale-factor interval [0.985, 1.015].

Detect Head Rotation

If the head size detection is successful, for the sake of efficiency, we do not correlate TEMPLATE again but simply use the correlation score it obtained during the head size detection subprocess along with its detected location. On the other hand, if the head size detection is unsuccessful and shifting occurs, TEMPLATE is restored for the second tracking subprocess and the correlation score it previously obtained against the current frame is reused. This restoration is necessary since shifting changes the image of the head contained in TEMPLATE.

Since a head rotation occurring in the image plane may increase the width of the area covered by the head, we must ensure that the head is enclosed within the area over which we correlate CCW and CW by selecting a sufficiently large portion of the captured image. To simplify the task of setting the number of correlation windows in our algorithm, we utilize the same number we used to correlate TEMPLATE, i.e., eleven horizontal and vertical windows, symmetrically centred about the location of the head we detected by correlating TEMPLATE.

Again, once we have obtained the three correlation scores, we ensure that TEMPLATE has produced the best correlation score and shift the templates if this is not the case. We then define a function that interpolates the resulting three correlation scores and compute its minimum, which represents the best approximation of the rotation angle of the head found in the current frame.

Zone of No-Rotation

To deal with jittery results, we define a zone of *no-rotation* about the origin of the plot. In the case of head rotation, the x-axis of the plot represents head rotation angle. The interval covered by the no-rotation zone was defined to include all head rotation angles from -5° to +5° as determined experimentally. If the minimum of the interpolated function falls within this zone, it may be the case that the detected head rotation angle is the effect of camera noise, and we therefore conclude that the user's head has remained more or less unchanged from the previous frame. In this situation, we do not compute a new head rotation angle for the current frame and the previously approximated angle is retained. If the minimum is located outside of this zone, we conclude that a rotation has occurred and a new approximation of the rotation angle of the user's head in the current frame is computed.

Shifting

The shifting is done as explained in Chapter III. However, here are a few algorithmic details.

Template shifting can be performed during each section of the tracking, independently. If shifting occurred during the head size detection section, TEMPLATE is restored for the head rotation section.

The extra LARGE or extra SMALL template produced by the shifting process corresponds to the next head size 0.08 or -0.08 away from LARGE or SMALL, respectively, on the x-axis of the plot, or the "head size" axis. Extra LARGE and extra SMALL are created by scaling the core template by 1.16 (1.08 + 0.8) and 0.84 (0.92 - 0.08), respectively. When we repeat the head size detection section with the new and shifted templates, only the newly created extra LARGE or extra SMALL is correlated.

The extra CCW or extra CW template produced by the shifting process also corresponds to the next head rotation angle -10° or 10° away from CCW or CW, respectively, on the x-axis of the plot, or the "head rotation angle" axis. Extra CCW and extra CW are created by rotating the core template by -20° and 20°, respectively. When we repeat the head rotation detection section for the second time using the new and shifted templates, only the newly created extra CCW or extra CW is correlated.

For either of the two tracking sections, shifting is never performed more than once. During experimentation, we observed that, if tracking has not failed, TEMPLATE usually has the best correlation score after performing the shifting process only once.

If, after shifting the templates, LARGE, SMALL, CCW, or CW is still the best approximation for the image of the head in the current frame, we conclude that either or both head detection subprocesses have failed. If the head size detection subprocess has failed, we ignore this subprocess (head size is not computed) and move on to the head rotation detection subprocess, using the same frame. If the head rotation detection subprocess has failed, but the head size detection subprocess was successful, we move on to the synthesis of a new core template. If both subprocesses have failed, our previously saved model is restored and the tracking process starts anew with the next frame.[2] The various reasons causing the failure of the detection subprocesses are discussed in the Failure and Recovery section below.

[2] Repeated failures over several frames are dealt with in a similar fashion.

Synthesize New Core Template

If at least one section of the tracking was successfully performed, the best approximation for the image of the head pose found in the current frame is created. To do so, a newly synthesized core template, or NEW TEMPLATE, is created using a cumulative version of the head size and head rotation angle that represents the head size and head rotation angle the head has undergone since the very first frame. These cumulative head size and head rotation angle values are updated whenever a detection subprocess of the tracking was successfully performed and a new head size or head rotation angle was computed for that frame. The reason why we use such cumulative values is because we produce the NEW TEMPLATE using an enlarged version of the originally captured master template, which we have previously saved. This is to alleviate degradation that the newly synthesized template would suffer over a certain amount of frames, if it were created using previous NEW TEMPLATE's. The newly synthesized core template is then created by rotating the enlarged version of the master template only once, using the cumulative head rotation angle, and by scaling the result of this rotation only once, using the cumulative head size.

Locate New Core Template

Although we have just synthesized a new core template representing the closest approximation of the image of the head pose, we do not know its exact location in the current frame. We find its location by correlating NEW TEMPLATE against the current frame following a process similar to the one described in the Detection sections above.

Display

This process displays different images depending on the state of the head movement tracking system.

The captured frames are displayed when the camera is activated. When the user has captured the master template, the display process outputs the captured frame and overlays a white box upon it, indicating the location of the master template within the captured image and its size. This is depicted in Figure 4-3 (b). When the tracking process has been initiated and has successfully

produced and located NEW TEMPLATE, our head movement tracking system indicates this location by displaying a white tracking box in the captured image it outputs. Our tracker represents the location of NEW TEMPLATE by using the pixel coordinates of the top left corner of the best-correlating window. Our tracker uses the detected head size found in the current image to determine the size (width and height, expressed in pixels) of the tracking box. Finally, it uses the detected head rotation angle, expressed in degrees, and its orientation, expressed by the sign of the angle, to orient the crosshair it draws at the centre of the tracking box. Note that the location of the crosshair does not correspond to any facial features within the tracking box.

Figure 4-6 gives an example of a displayed frame once the location, size and orientation of the head have been detected.

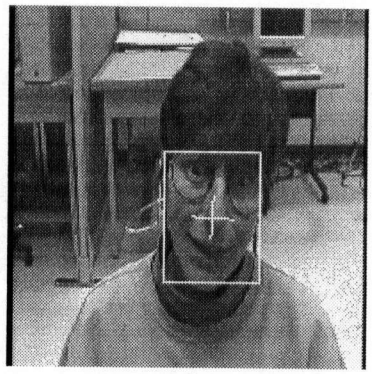


Figure 4-6
Displayed Frame with Detected Head Position, Size, and Orientation

The tracking box is always drawn in white except when the tracking fails. In the case when the tracking is temporarily suspended for a particular frame, or sequence of frames, the box is drawn in light blue. When the tracking is permanently turned off, the box becomes black. When TEMPLATE is the best-matched template during the head size and head rotation detection subprocesses, the crosshair is drawn in white. If LARGE or SMALL was initially the best-matched template during the head size detection section of the tracking, before shifting occurred, the horizontal line of the crosshair is drawn in yellow or blue, respectively. If CW or CCW was initially the best-matched template during the head rotation detection section, before shifting occurred, the vertical line of the crosshair is drawn in purple or green, respectively.

Failure and Recovery

We mentioned that the results of the correlation-based template matching technique are affected by pixel value changes if they occur asymmetrically, i.e., in the captured image but not in the template. Such changes in pixel values may be due to:

- occlusion,
- lighting condition changes in the scene,
- head movement not modelled by our templates, or
- head movement outside the field of view of the camera.

All these situations will cause corresponding pixels in the template and the captured image of the scene to differ greatly in their values.

When any one of the first three situations listed above arises, the head size and rotation angle approximations determined by the template matching process are unreliable and produce templates that no longer resemble the head pose in the scene. Initially, when we experimented with the shifting process, we observed that this situation often leads to repeated shifting of the templates resulting in shrinking or expanding the templates to a size that no longer allowed the head tracking system to recover.[3] To prevent such failure, we have limited the shifting process to occur only once per frame. If the shifting needs to occur a second time, we skip that section of the tracking, restore the model and move on to the other section of the tracking or move on to the next frame. We have noticed that limiting the shifting helps recovery since the size of the templates, after shifting only once, are often still comparable to the size of the head in the scene.

The situation where the head moves outside the field of view of the camera is dealt with by sensing when the correlation windows associated with TEMPLATE are found to be partially or totally outside the image space. When this occurs, tracking is turned off for the remaining frames of the sequence. In an earlier version of our head movement tracking system, we did threshold the correlation scores to sense when the captured image of the scene no longer

[3] It was observed that SMALL often was erroneously found to be the best match and hence forced shifting to be performed many times. The tracking would then produce a newly synthesized template with decreasing size until the template would represent too small a head to realistically match the size of the head in the scene. Also, because the correlation windows are set to the size of the template, a decreasing template would restrict the search area for each processed frame, making recovery less likely.

contained the entire image of the head as portrayed in the templates. When the correlation scores were above a certain threshold, indicating a growing dissimilarity between the captured image and the templates, the current model was saved and its templates were constantly correlated against each incoming frame at the location where the head had left the scene. This was based on the assumption that the head would reenter the scene at approximately the same location at which it left it. A tracking process, similar to the one described in this thesis, would then simply resume tracking when the head was sensed to have reentered the scene, i.e., when the correlation scores would be below the threshold value. This earlier version was abandoned because we found threshold values to be very sensitive to any changes occurring in the tracking situation such as lighting conditions, size and rotation angles of the head modelled by our templates, etc. More time would be needed to further investigate the use of threshold values in a robust approach.

We implemented our head movement tracking algorithm using C++ on a Pentium II workstation (400 MHz CPU) running the Linux operating system.[1] No special image processing hardware was utilized. The X Motif library was used to develop the graphical user interface (GUI) of our tracking system. During tracking, the images of the scene are captured using the S-video signals of a Panasonic CP410 colour camera and a BT848 video input card. The size of the captured images is set to 180 × 180 pixels. These images are displayed using the RGB colour format with thirty two bits per pixel.[2] The size of the templates varies based on the size of the user-selected master template. All images and templates are filtered and sampled down by a factor of two and converted to greyscale before correlation is performed.

In this chapter, we discuss how our head movement tracking system has satisfied (or failed to satisfy) our initial requirements stated in Chapter I.

Head Movement Detection

We required our head movement tracking system to be able to detect and track the following head movements:

1) head translation in the image plane, i.e., the (x,y)-plane parallel to the monitor;
2) head translation towards and away from the monitor, along the z-axis; and
3) head rotation in the image plane, i.e., about the z-axis (roll), in the clockwise and counterclockwise directions.

We now provide results that qualitatively demonstrate that our head movement tracking system satisfies the above requirements.

[1] Linux RedHat version 6.0.
[2] The value of each of the three colours is represented using eight bits. The last eight bits are assigned to a value called alpha or A which is unused in this system.

Tracking Results

For sake of readability, we shall continue to name our templates TEMPLATE, LARGE, SMALL, CW, CCW and NEW TEMPLATE.

The figures presenting our tracking results are located at the end of this chapter. As stated in the Display section of Chapter IV, a tracking box is drawn in the frame to indicate the position and size of the tracked head, while a crosshair is drawn to indicate its rotation angle. We also indicate, for each displayed frame, its best-matching template, from each set of templates, using the following scheme.

- When TEMPLATE best matches the current image, both lines of the crosshair are drawn in white.
- When LARGE (SMALL) initially best matched the current image, before shifting occurred, the horizontal line of the crosshair is drawn as a dotted white (black) line as illustrated in Figures 5-1 (a) and (b), respectively.
- When CCW (CW) initially best matched the current image, before shifting occurred, the vertical line of the crosshair is drawn as a dotted white (black) line as illustrated in Figures 5-1 (c) and (d), respectively.

(a) The white dotted horizontal line of the crosshair indicates that LARGE best matched the current image, before shifting occurred.

(b) The black dotted horizontal line of the crosshair indicates that SMALL best matched the current image, before shifting occurred.

(c) The white dotted vertical line of the crosshair indicates that CCW best matched the current image, before shifting occurred.

(d) The black dotted vertical line of the crosshair indicates that CW best matched the current image, before shifting occurred.

Figure 5-1

Representation of Best-Matching Template in Tracking Results

A grey, crosshair-less tracking box signifies that tracking has temporarily failed.

Head Movements

Our head movement tracking system tracks head translation and rotation in the image plane.

1) Head Translation in Image Plane

Figure 5-2, located at the end of this chapter, displays twelve frames from a video sequence in which the head translates from the right side of the scene to the left side (frames 3, 20, 25 and 39) then downward (frames 69, 79 and 89) and finally upward (frames 100 and 106). The tracking box correctly follows the translating head position.

2) Head Rotation in Image Plane

Figure 5-3 displays twelve frames from a video sequence in which the head rotates counterclockwise in the scene. The orientation of the crosshair drawn in the centre of the tracking box correctly follows the changing rotation angle of the head. In the first two frames, the head rotation is too small for the tracking system to detect as indicated by the non-rotated crosshair. In frames 14 and 20, the rotated, white, solid crosshair indicates that a relatively small counterclockwise head rotation was detected since TEMPLATE was found to best approximate the head rotation in these frames. In frames 10, 16 and 26, the vertical white dotted line of the crosshair indicates that the tracking system has detected a large counterclockwise head rotation since CCW was found to best approximate the head rotation in these frames. In frames 27, 34 and 36, the vertical black dotted line of the crosshair indicates that the tracking system has detected a large clockwise head rotation. In this situation, having CW being the best-correlating template indicates that, even though the head is still in a counterclockwise rotated position, it has started to rotate in the opposite direction.

In Figure 5-2, a slight counterclockwise rotation of the head is detected in frames 25 and 39 as indicated by the slightly slanted white crosshair.

3) Head Translation along z-axis

Our head movement tracking system tracks head translation towards and away from the monitor. Figure 5-4 displays nine frames from a video sequence in which the head translates along the z-axis, away from the monitor. Not only does the tracking box correctly follow the head position but it also detects its movement away from the monitor. This movement is inferred by the detected decreasing head size and indicated by the decreasing size of the white tracking box.

Looking back at Figure 5-2, head translation along the z-axis is also detected. In frames 128, 140 and 156, the tracking box increases in size. In frame 89, not only is the size of the tracking box increasing, but the horizontal white dotted line of the crosshair indicates that LARGE was found to be the best-correlating template for this frame, implying that the head motion was faster than in frames 128, 140 and 156. Over these frames, the tracking system has detected an increase in head size which, following our model, indicates that the head is moving towards the monitor. In frame 39, the decrease in size of the tracking box and the horizontal black dotted line of the crosshair indicate that SMALL was found to be the best-correlating template for this frame. In this situation, the tracking system detected a much smaller head size, which indicates that the head is moving away from the monitor at a quick pace. Notice that over these sequences, the area of the scene enclosed by the tracking box remains proportionally the same as the head translates and the size of the box changes.

Failure

Our head movement tracking system does not always successfully track head movements. It fails when there is a significant amount of head rotation in depth. Figure 5-5 shows a sequence of nine frames over which the tracking of the head rotation is unsuccessful. Even though the overall head movement is a clockwise head rotation in the image plane, a head rotation in depth may also be part of the head movement causing features of the face, present (absent) in the master template, to disappear (appear) in the captured image. Also, shadows are now obscuring the eye area in frames 43 and 45 altering the intensity value of these pixels and consequently affecting the results of the correlation process and hence the tracking itself.

Figure 5-6 also shows a sequence of nine frames over which tracking fails. The light gray-coloured box without a crosshair in frame 38 signifies that the tracking temporarily fails for that frame. However, towards the end of the sequence, as the rotation in depth lessens in the head movement, the tracking recovers.

Figures 5-2 to 5-6 also display, in their upper left corner, the selected master template used to perform the tracking over these sequences.

Processing Time

The time needed for our head movement tracking system to process a frame is a function of the type of head movement occurring in the scene. The fastest processing time is obtained when no motion is detected. In these situations, no template shifting occurs. Also, since both minima fall within the zone of no-motion and no-rotation, no synthesis of a new core template is performed. Our tracking system then processes approximately eight frames per second.

The slowest processing time is obtained when large head translation and rotation are detected. In these situations, template shifting is performed twice (once for the head size template set and once for the head rotation template set) along with additional correlation. The synthesis of the new core template also requires both scaling and rotating to be performed. Our tracking system then processes about three frames per second.

Time-consuming processes are the frame capture process (0.01 second), the model creation or update process (on average 0.055 second), the template shifting process (on average 0.0225 second), the new core template synthesis (on average 0.06 second), and finally the correlation process. The time required by the latter process is a function of the number of windows, in the captured image, over which correlation is performed. Correlating TEMPLATE, CW, and CCW requires more processing time (nearly 0.02 second) than correlating LARGE and SMALL (approximately 0.01 second) since the former templates are correlated over a larger portion of the image. Table 5-1 lists these processes and their associated times.

Tracking Process	Time (average in seconds)
Frame Capture	0.01
Model Creation/Update	0.055
Shifting	0.0225
New Core Template Synthesis	0.06
Correlating TEMPLATE, CW, CCW	0.02
Correlating LARGE, SMALL	0.01

Table 5-1
Tracking Processes with Associated Times

Our head movement tracking system performs rather slowly but it does not create a great lag time between the moment the user moves her/his head and the moment the tracking result is displayed.

Spatial Accuracy

Translation

To quantitatively define the accuracy with which our head movement tracking system detects head position and head size (width and height), we tracked a two-dimensional object. We used a portrait, framed with a black-coloured border to ease the process of locating its position and size within the captured image of the scene. Our tracking system displays its results in a two-dimensional fashion by drawing a tracking box over the tracked position of the object. Therefore, by having our system tracking a portrait that was being translated in all three directions in the scene (but not rotated), it was easy to compare the measured position and size of our portrait against the position and size obtained from our tracker and compute the difference.

We used the portrait of a head so as to keep the tracking situation as close as possible to the situation for which our head movement tracking system would usually be used, i.e., tracking heads.

Specifically, to measure the position and dimensions (width and height) of our head portrait in the image captured by the camera, we located the top left and bottom right corners of the portrait's black border using image editing and processing software. We expressed these corners in terms of image pixel coordinates.

Figure 5-7 shows a sequence of eight frames along with the tracked and measured positions of our head portrait, its tracked and measured sizes (width and height) and the discrepancies (error) between the measured and tracked values. Table 5-2 lists these tracked and measured positions and dimensions along with their associated errors.

Tracked Portrait (x,y) Position (in pixels)	**Measured Portrait (x,y) Position (in pixels)**	**Computed (x,y) Error (in pixels)**	**Tracked Portrait Width and Height (in pixels)**	**Measured Portrait Width and Height (in pixels)**	**Computed Width and Height Errors (in pixels)**
(88,54)	(88,55)	(0,-1)	w: 54 h: 94	w: 52 h: 92	Δw: 2 Δh: 2
(60,36)	(59,38)	(1,-2)	w: 43 h: 74	w: 43 h: 73	Δw: 0 Δh: 1
(76,54)	(74,55)	(2,-1)	w: 53 h: 92	w: 54 h: 90	Δw: -1 Δh: 2
(100,32)	(100,33)	(0,-1)	w: 37 h: 67	w: 40 h: 69	Δw: -3 Δh: -2
(48,48)	(45,48)	(3,0)	w: 48 h: 84	w: 51 h: 83	Δw: -3 Δh: 1
(108,46)	(109,46)	(-1,0)	w: 45 h: 83	w: 45 h: 82	Δw: 0 Δh: 1
(54,40)	(54,41)	(0,-1)	w: 45 h: 78	w: 45 h: 76	Δw: 0 Δh: 2
(84,52)	(85,53)	(-1,-1)	w: 52 h: 92	w: 51 h: 90	Δw: 1 Δh: 2

Table 5-2

Tracked and Measured Portrait Positions and Dimensions with Associated Errors

The errors represent the number of pixels by which the tracked position and size of the portrait under- or over-estimated its measured position and size. In the worst cases, our tracker failed to detect the measured position of the portrait by three pixels in the horizontal direction and two pixels in the vertical direction. It incorrectly detected width and height by three and two pixels, respectively. On average, our tracker failed to correctly estimate the position of the head portrait

by one pixel in both horizontal and vertical directions and its measured width and height by one and two pixels, respectively.

Rotation

We measured head rotation angles[3] and compared them with the head rotation angles obtained from our head movement tracking system. Figure 5-8 shows the frames we utilized and our results. The angles were measured from the centre of the displayed tracking box. The head represented by the master template for the sequence had an initial rotation angle of +4°. This angle was added to the measured angles before the discrepancy between both the measured and the tracked angles was computed. Table 5-3 lists these tracked and measured angles along with their associated errors.

Tracked Head Rotation Angles (in degrees)	Measured Head Rotation Angle (in degrees)	Computed Error (in degrees)
-5.1	-8.0	-2.9
-27.8	-27.0	0.8
-10.3	-12.0	-1.7
-15.3	-17.0	-1.7
-16.5	-16.0	0.5
-9.9	-9.0	0.9

Table 5-3

Tracked and Measured Head Rotation Angles with Associated Errors

The errors represent the degrees of arc by which the tracked head rotation angle under- or over-estimated the measured angle. In the worst case, our tracker failed to detect the measured rotation angle by -2.9°. This worst case occurred for a measured head rotation angle of -8.0°, which was the smallest measured angle of the sequence. This seems to corroborate the results

[3] We measured head rotation angles using a tool with a precision of ±0.05°.

reported by Chen et al. [Chen 98] where their largest error occurred for the smallest measured angle. This can be explained by considering the proximity of such small angles to the range of angles over which template matching is tolerant. Therefore, template matching may not detect small angles as reliably as larger angles. On average, our tracker estimated the head rotation angle to within 1.4° of the measured angle. Thus, our head movement tracking system tracks head rotation angles as accurately as the tracking system developed by Chen et al. [Chen 98] as discussed in Chapters I and II.

Movement in the Background

Movement in the background usually does not affect our head movement tracking system as the sequence shown in Figure 5-9 demonstrates. Above each frame of this sequence, we display the newly synthesized core template resulting from the detection of the size and rotation angle of the head. This new synthesized core template is the best approximation of the head pose in that frame.

Small Changes in Facial Expression

Varying facial expressions usually do not affect our head movement tracking system as the sequence shown in Figure 5-10 demonstrates. The master template for this sequence is similar to the content of the tracking box displayed in the first frame (frame 2). The tracking is performed successfully even though the facial expression of the head, or pixel pattern, contained in the displayed frames of the sequence continuously differs from the facial expression of the head contained in the master template.

Occlusion

We experimented with occlusion. Theoretically, since our tracking system detects the degree of similarity between portions of the image and our templates, we expect the performance of our tracking system to be affected when other objects in the scene are occluding the object we are

tracking, hence reducing this similarity. Figure 5-11 displays two sequences depicting tracking performed while partial occlusion occurs.

In the first sequence, a hand occludes the face. The tracking fails because, even though both the hand and the face have similar colour, the hand is brighter (pixels with different values) and the overall image of the hand occluding the face produces a low similarity score when correlated with the template of a face. However, before the tracking fails, i.e., when the part of the face the hand is occluding is still small, this pixel value similarity fools the correlation process into considering the hand to be part of the face. Notice how the hand is "included" in the box in frame 58. When the occlusion is done using a non-skin object, as in the second sequence where a piece a paper is utilized, the produced pixel intensity of the image of the non-skin object differs greatly from the pixel intensity of the image of the face. The correlation process senses this dissimilarity and the object is excluded from the portion of the image that is being tracked. Notice, in frame 60, the box has shifted away so as not to include the piece of paper. In frame 68, as the paper is taken away, the box returns to cover the whole face. When a non-skin object occludes a small portion of the face (less than 50%), as in the second sequence, tracking seems unaffected by the occlusion.

Lighting Conditions

Experimenting with our head movement tracking system in environments with different lighting conditions, we noticed that our tracker performed less reliably in darker scenes, as depicted in Figure 5-5. We also noticed a poorer performance from our tracker when part of the image darkened due to shading.

Camera Noise

In developing our head movement tracking system, we tried to reduce the effect of camera noise, as discussed in Chapters III and IV. In situations where the user kept her head immobile, we observed that our tracker was not affected by such noise and displayed a motionless tracking box.

Object Tracking

The approach we took in solving the head movement tracking problem was not restricted to human heads as we did not utilize any physical, geometrical or colour information related to this type of object. Figure 5-12 displays the results we obtained while tracking a rubber duck. The sequence of twelve frames shows successful tracking of the object as it translates in the image plane (frames 38, 50, 60 and 100), as it rotates clockwise in the image plane (frames 120, 128, 136 and 145), and as it translates towards the monitor (frames 2, 18 and 34).

However, the rather slow processing time of our tracker restricts the choice of objects it can successfully track. Also, the technique we used, namely correlation-based template matching, restricts the choice of objects with which our tracker can successfully deal. Because template matching relies on pixel values to establish similarity levels between a template and various locations in the captured images (correlation windows), good tracking results are obtained if the object has a distinct pattern or texture represented by a wide range of pixel values. Correlating a template that images an object having predominantly one colour against various correlation windows in the captured image containing that object would not allow template matching to determine decisively which window would best match the template representing the object since these correlation windows would all be producing similar correlation scores.

Captured master template for this tracking sequence.

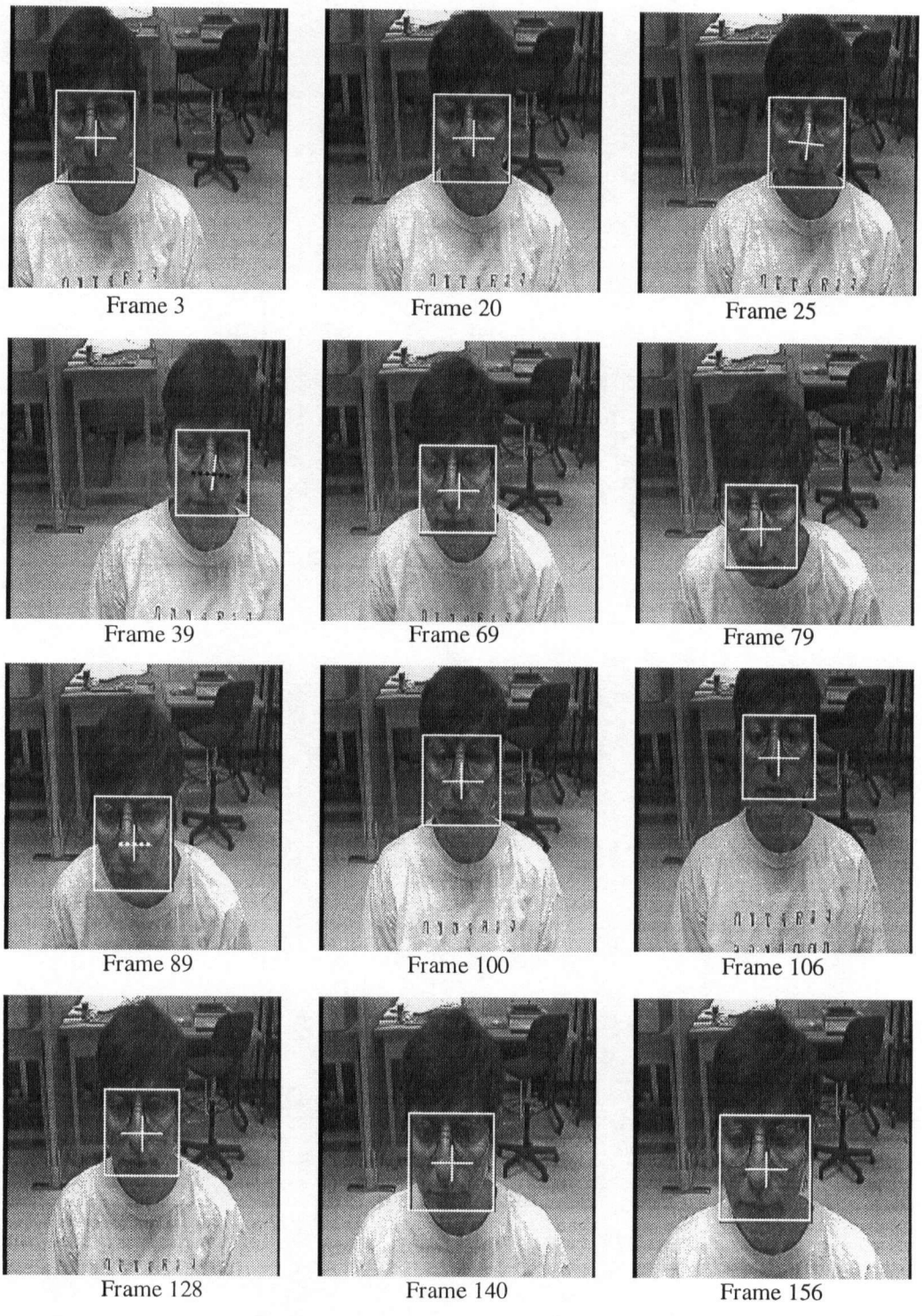
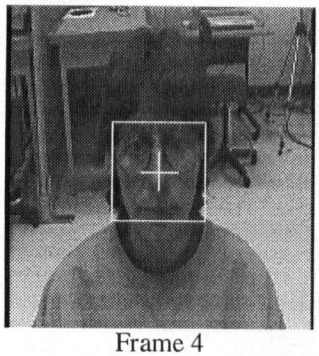


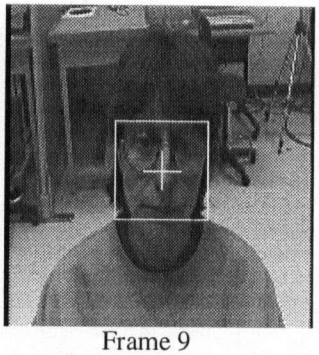
Figure 5-2 Tracking of Head Translation in Image Plane

The core template and multiple frames of a video sequence representing the tracking of head translations in the image plane.

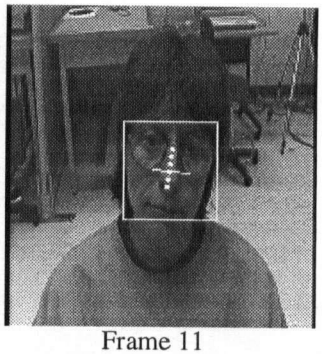
Captured master template for this tracking sequence.



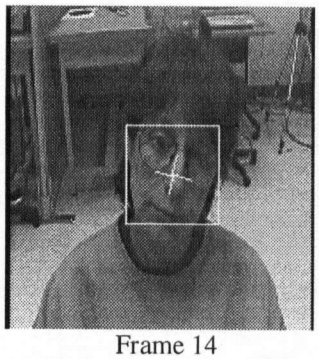
Frame 4



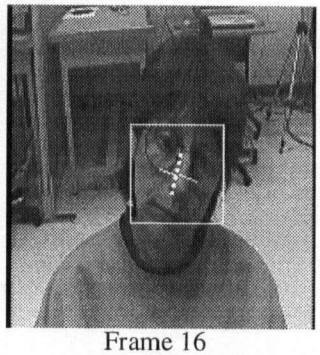
Frame 9



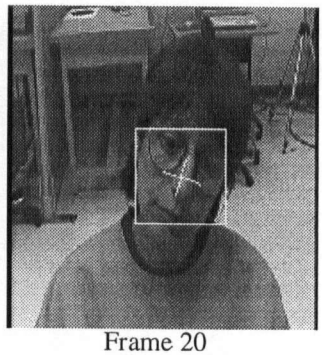
Frame 11



Frame 14

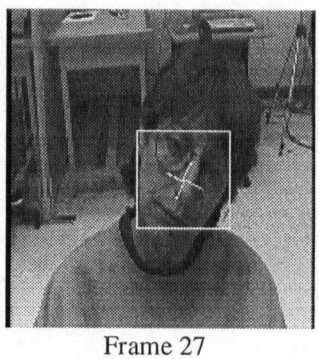


Frame 16

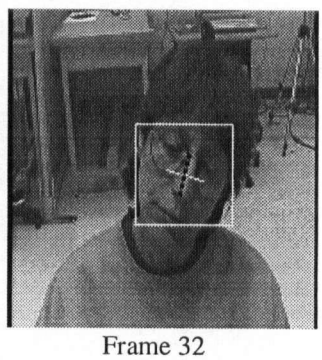


Frame 20

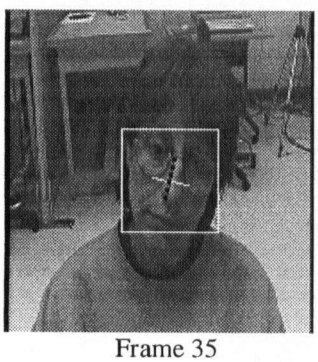
Frame 26



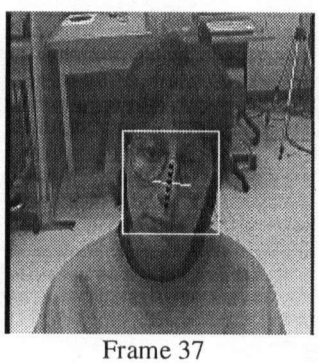
Frame 27



Frame 32



Frame 35



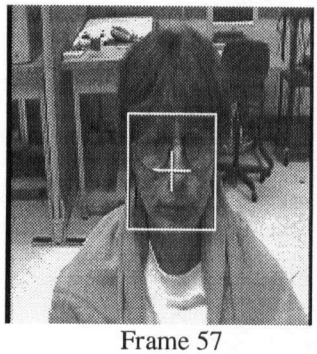
Frame 37

Frame 39

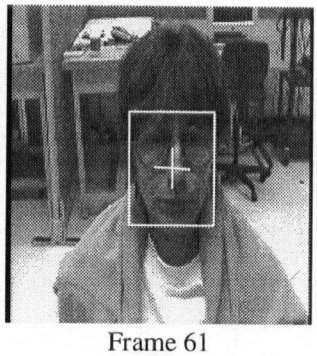
Figure 5-3 Tracking of Head Rotation in Image Plane

The core template and multiple frames of a video sequence representing the tracking of a counterclockwise head rotation followed by a clockwise head rotation.

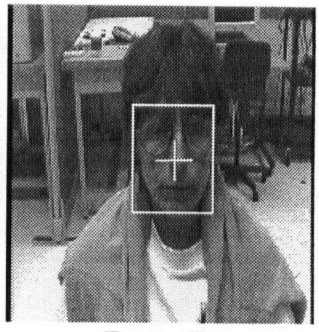
Captured master template for this tracking sequence.



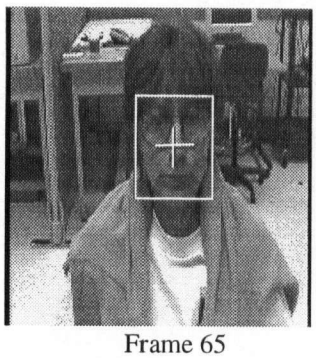
Frame 57



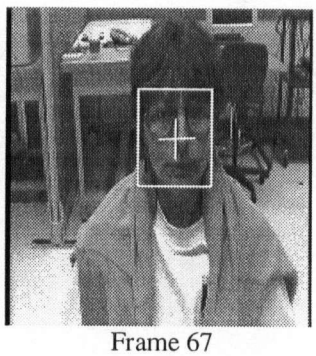
Frame 61



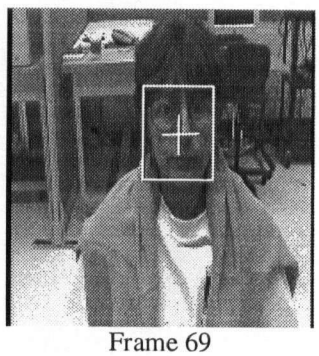
Frame 63



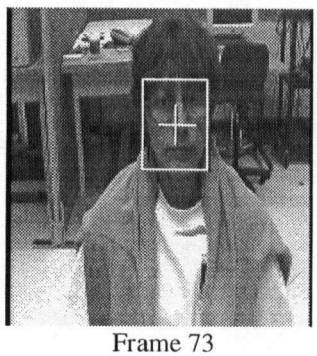
Frame 65



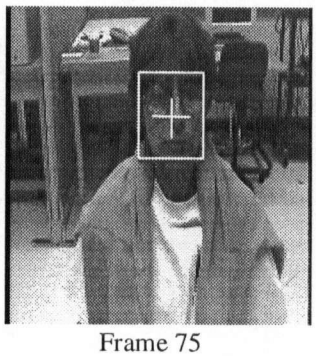
Frame 67



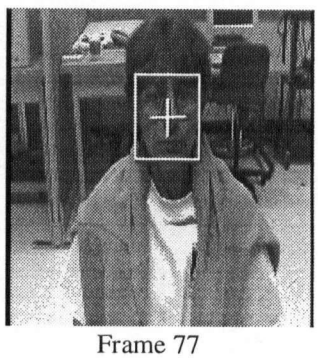
Frame 69



Frame 73



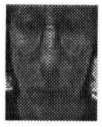
Frame 75



Frame 77

Figure 5-4 Tracking of Head Translation along the z-axis

The core template and multiple frames of a video sequence representing the tracking of a head translation away from the monitor.



Captured master template for this tracking sequence.

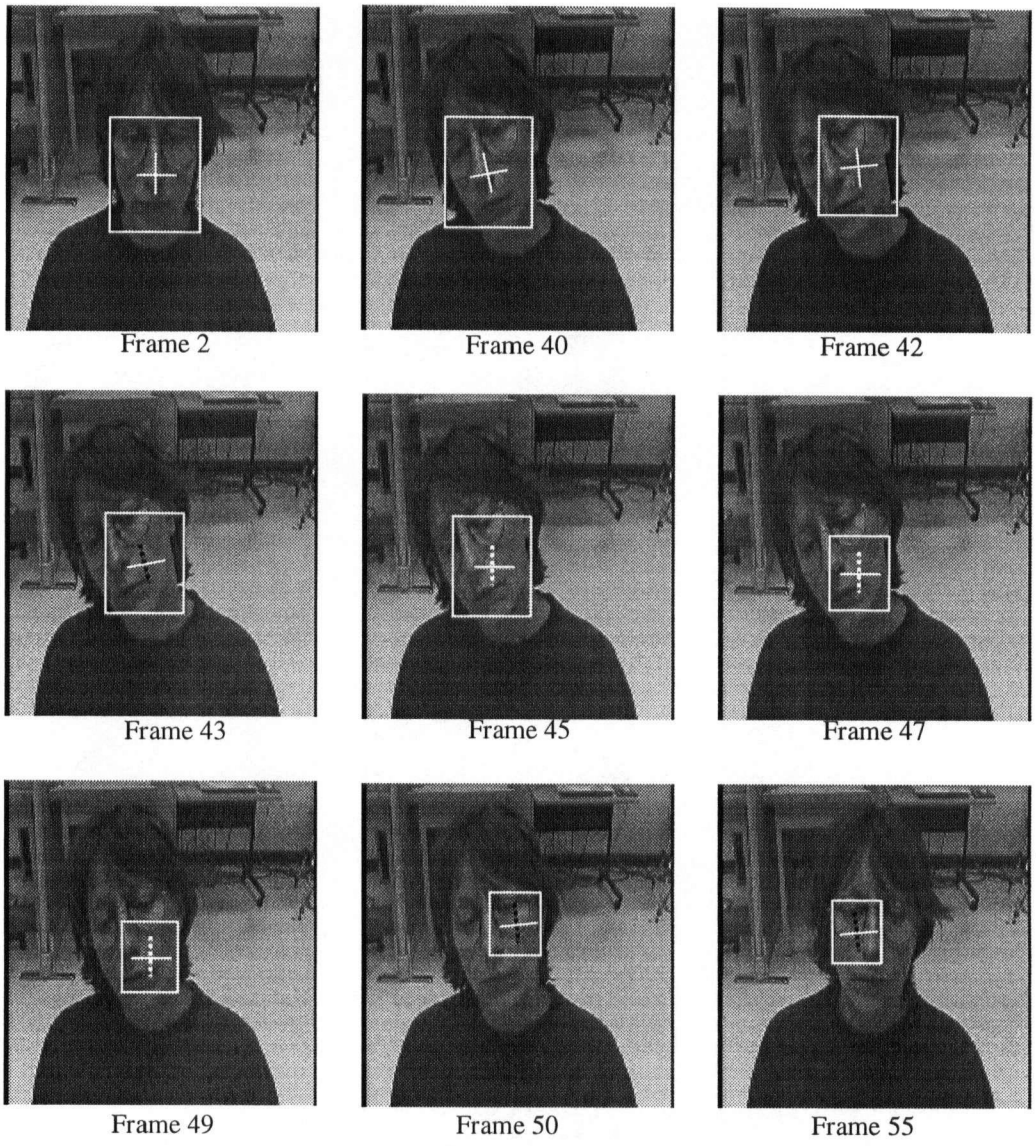


Figure 5-5 Tracking Failure

The core template and multiple frames of a video sequence for which the tracking fails.

Captured master template for this tracking sequence.

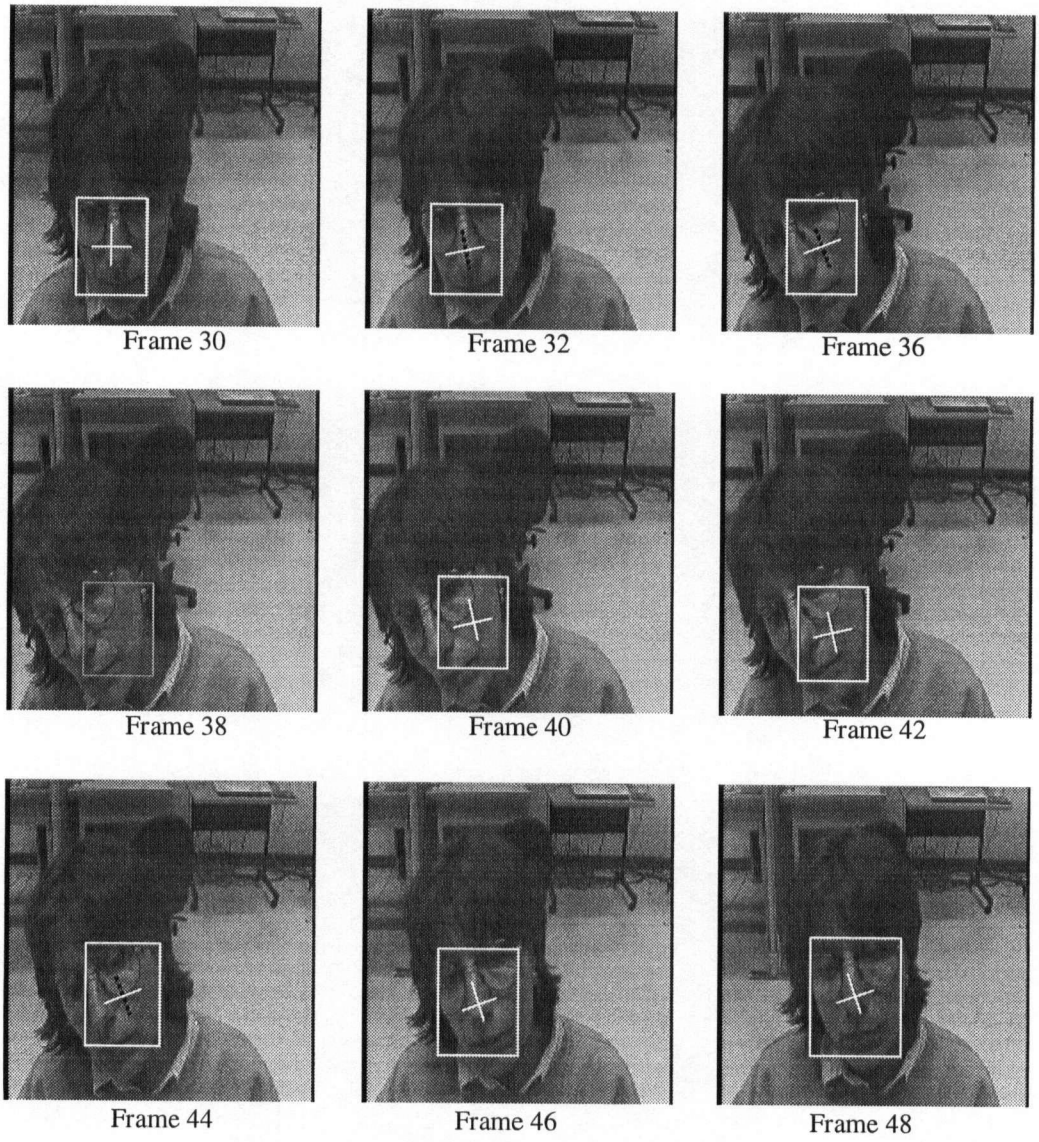
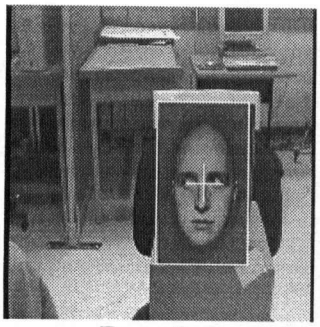


Figure 5-6 Tracking Failure and Recovery

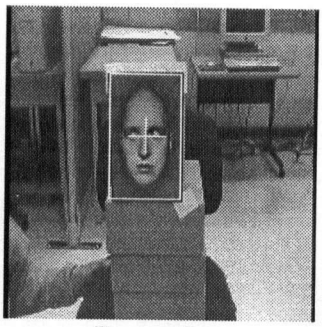
The core template and multiple frames of a video sequence representing tracking failure and successful recovery.



Measured position:
(88,55)
Tracked position:
(88,54)
Error in x: 0 pixels
Error in y: 1 pixel

Measured width: 52 pixels
Measured height: 92 pixels
Tracked width: 54 pixels
Tracked height: 94 pixels
Error in width: 2 pixels
Error in height: 2 pixels

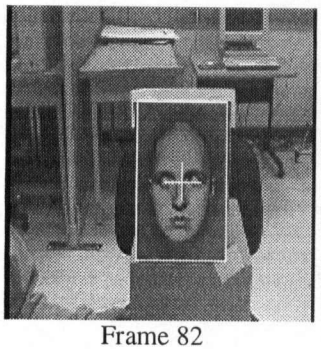
Frame 2



Measured position:
(59, 38)
Tracked position:
(60,36)
Error in x: 1 pixel
Error in y: 2 pixels

Measured width: 43 pixels
Measured height: 73 pixels
Tracked width: 43 pixels
Tracked height: 74 pixels
Error in width: 0 pixels
Error in height: 1 pixel

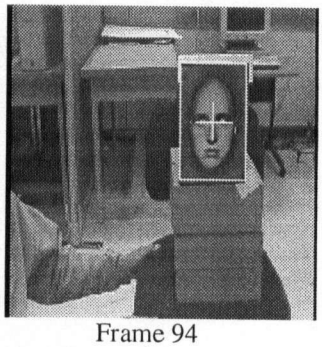
Frame 49



Measured position:
(74,55)
Tracked position:
76,54)
Error in x: 2 pixels
Error in y: 1 pixel

Measured width: 54 pixels
Measured height: 90 pixels
Tracked width: 53 pixels
Tracked height: 92 pixels
Error in width: 1 pixel
Error in height: 2 pixels

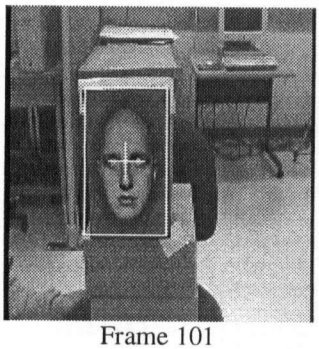
Frame 82



Measured position:
(100,33)
Tracked position:
(100,32)
Error in x: 0 pixels
Error in y: 1pixel

Measured width: 37 pixels
Measured height: 67 pixels
Tracked width: 40 pixels
Tracked height: 69 pixels
Error in width: 3 pixels
Error height: 2 pixels

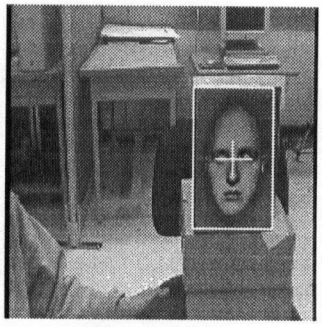
Frame 94



Measured position:
(45,48)
Tracked position:
(48,48)
Error in x: 3 pixels
Error in y: 0 pixels

Measured width: 51 pixels
Measured height: 83 pixels
Tracked width: 48 pixels
Tracked height: 84 pixels
Error in width: 3 pixels
Error in height: 1 pixel

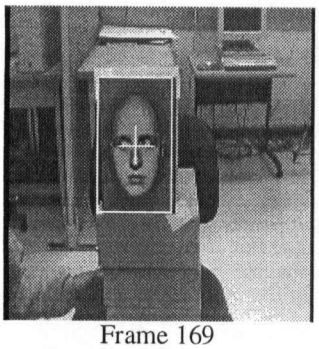
Frame 101



Measured position:
(109,46)
Tracked position:
(108,46)
Error in x: 1pixel
Error in y: 0 pixels

Measured width: 45 pixels
Measured height: 82 pixels
Tracked width: 45 pixels
Tracked height: 83 pixels
Error in width: 0 pixels
Error in height: 1 pixel

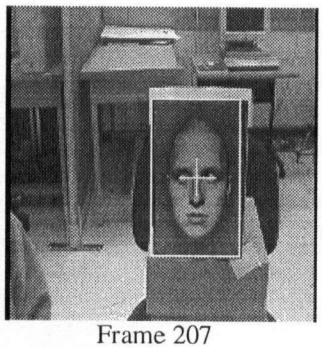
Frame 135



Measured position:
(54,41)
Tracked position:
(54,40)
Error in x: 0 pixels
Error in y: 1 pixel

Measured width: 45 pixels
Measured height: 76 pixels
Tracked width: 45 pixels
Tracked height: 78 pixels
Error in width: 0 pixels
Error in height: 2 pixels

Frame 169



Measured position:
(85,53)
Tracked position:
(84, 52)
Error in x: 1 pixel
Error in y: 1 pixel

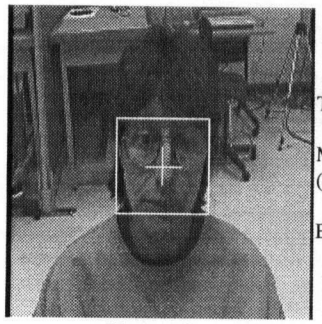
Measured width: 51 pixels
Measured height: 90 pixels
Tracked width: 52 pixels
Tracked height: 92 pixels
Error in width: 1 pixel
Error in height: 2 pixels

Frame 207

Figure 5-7 Tracking Accuracy - Translation

Comparison between tracked and measured positions and sizes
in multiple frames of a video sequence representing
the tracking of the portrait of a head as it translates in the scene.

The captured master template for this tracking sequence has an initial rotation angle of +4°.

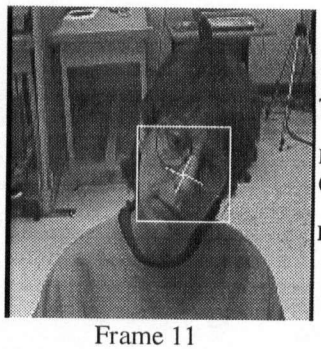


Tracked Angle: -5.1°

Measured Angle: -8°
(-4° - 4°)

Error: 2.9°

Frame 9

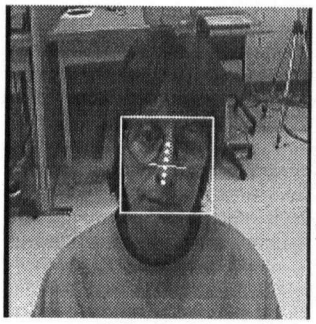


Tracked Angle: -27.8°

Measured Angle: -27°
(-23° - 4°)

Error: 0.8°

Frame 11



Tracked Angle: -10.3°

Measured Angle: -12°
(-8° - 4°)

Error: 1.7°

Frame 14

Tracked Angle: -15.3°

Measured Angle: -17°
(-13° - 4°)

Error: 1.7°

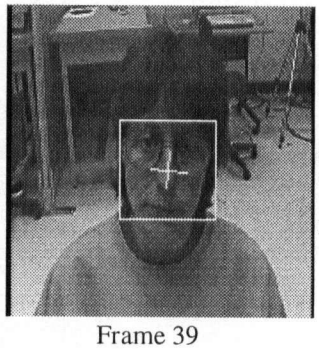
Frame 27

Tracked Angle: -16.5°

Measured Angle: -16°
(-12° - 4°)

Error: 0.5°

Frame 35



Tracked Angle: -9.9°

Measured Angle: -9°
(-5° - 4°)

Error: 0.9°

Frame 39

Figure 5-8 Tracking Accuracy - Rotation

Comparison between tracked and measured rotation angles in multiple frames of a video sequence representing a counterclockwise head rotation followed by a clockwise rotation. The initial rotation angle of +4° (clockwise direction) has been subtracted from the measured counterclockwise angle of the head for each frame. This subtraction of angles is indicated in the parenthesis.

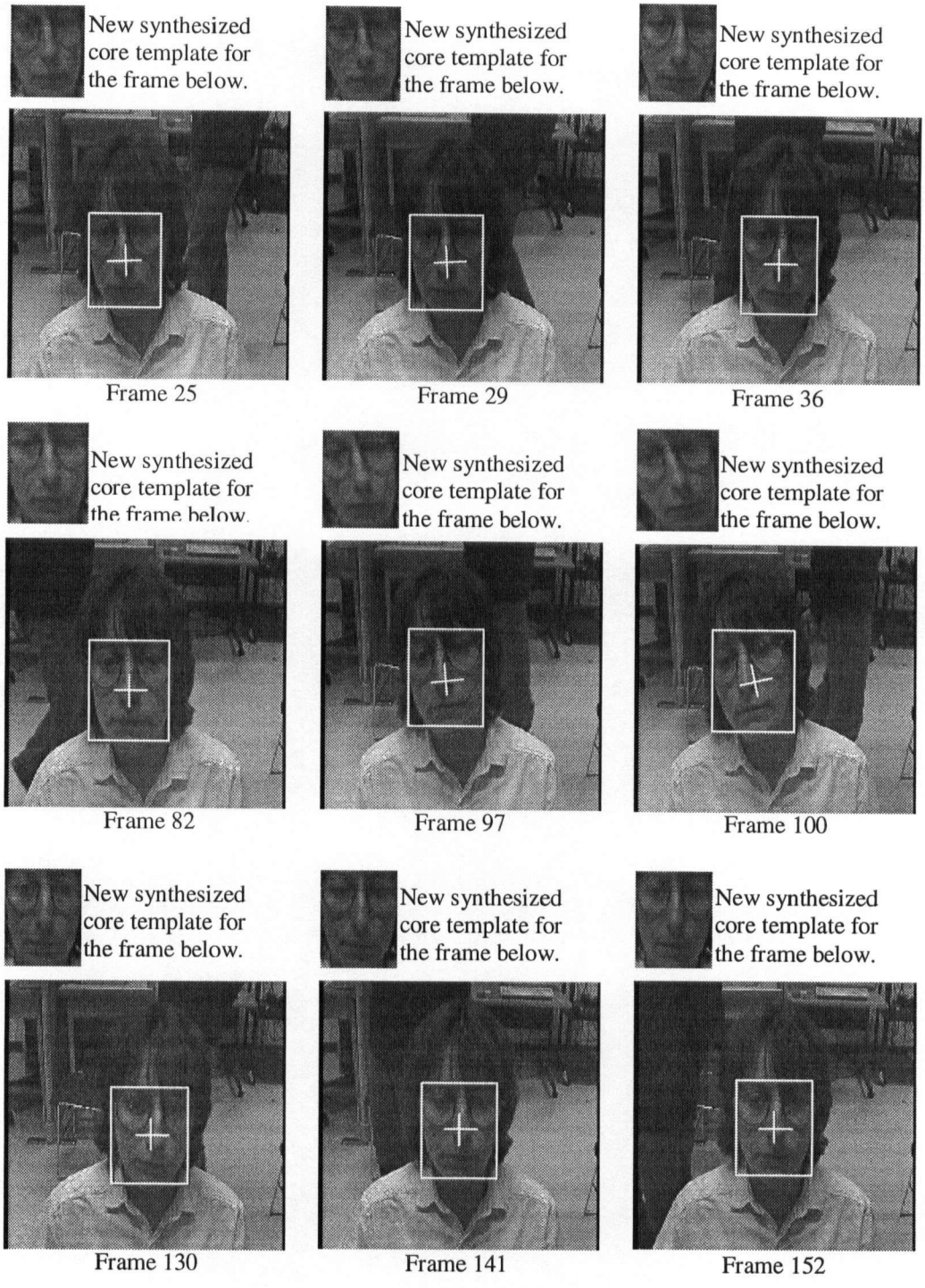


Figure 5-9 Tracking with Motion in Background

Multiple frames of a video sequence representing the tracking of a head while movements are occurring in the background. The newly synthesized core templates, created from the tracking of the head over each frame, are also displayed.

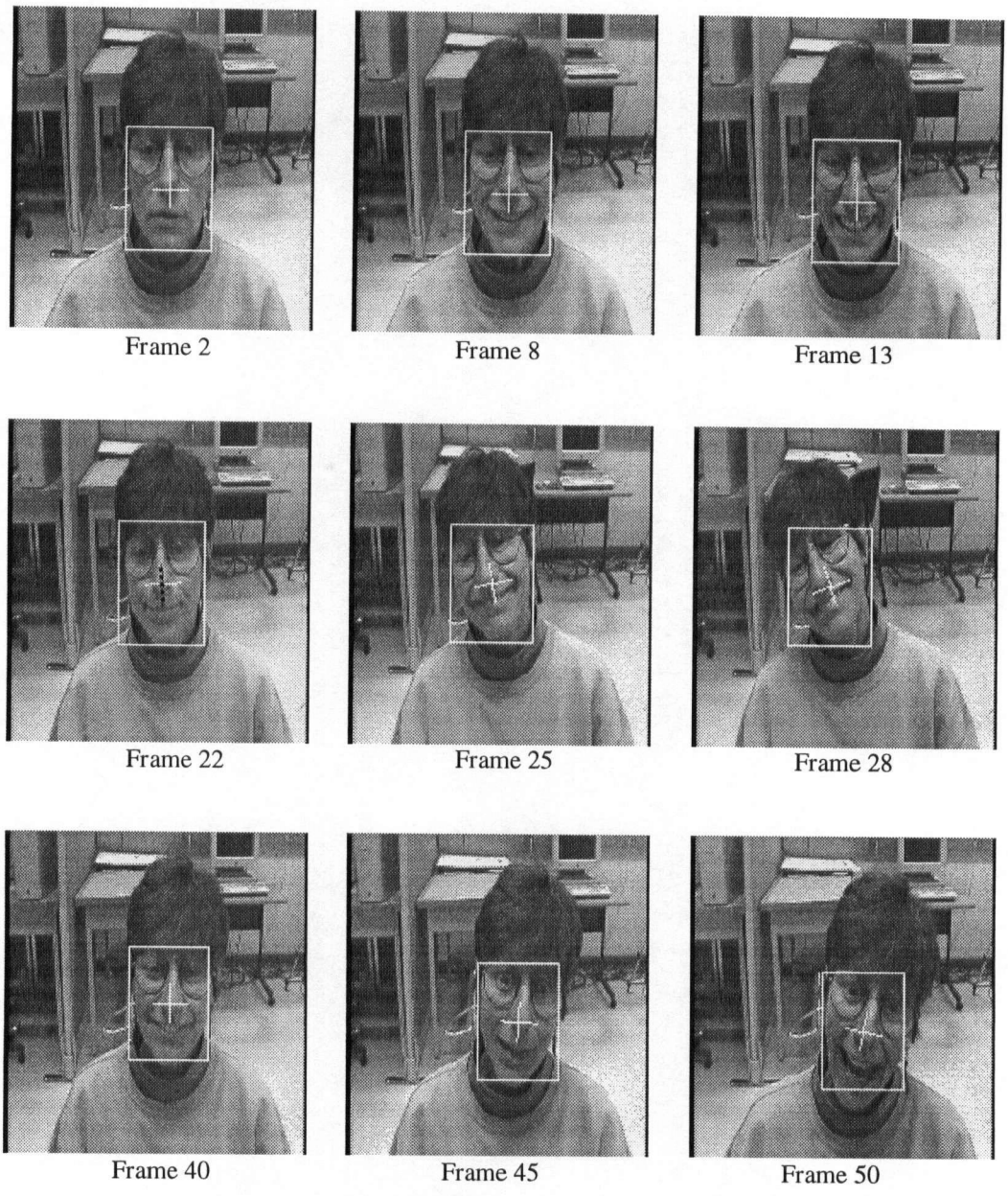


Figure 5-10 Tracking with Varying Facial Expressions

Multiple frames of a video sequence representing
the tracking of a head while facial expression changes occurred.

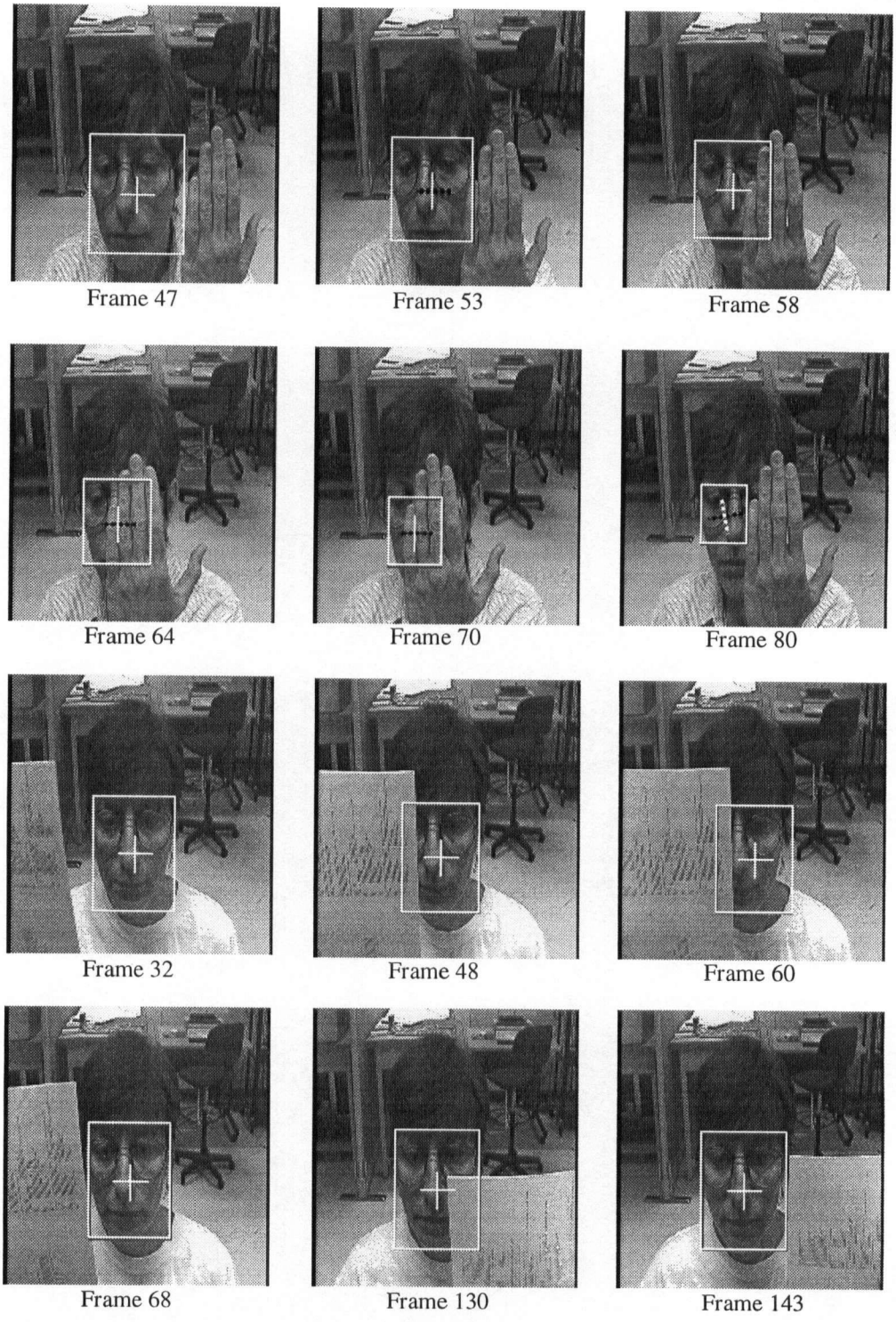


Figure 5-11 Tracking with Occlusion

a) Multiple frames of a video sequence representing tracking failure due to partial occlusion.

b) Multiple frames of a video sequence representing successful tracking while partial occlusion occurs.

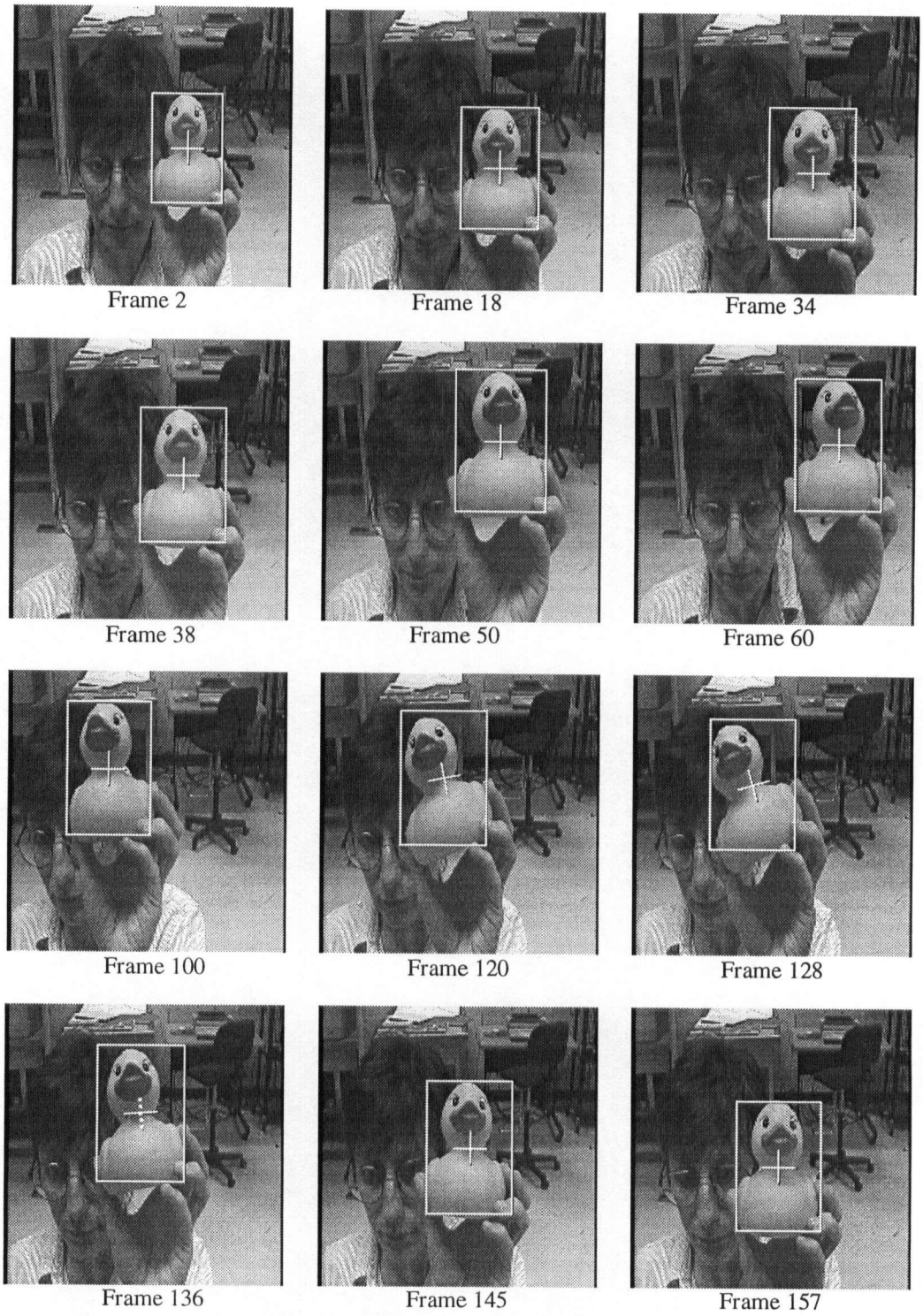


Figure 5-12 Tracking of Object

Multiple frames of a video sequence representing object tracking.

CHAPTER VI CONCLUSIONS AND FUTURE WORK

In this thesis, we investigated the problem of head movement tracking by developing a computer vision approach using correlation-based template matching. We demonstrated the feasibility of our approach by implementing a head movement tracking system that detects the movement of the head of a user seated at a computer workstation, specifically head translation and rotation in the image plane and head translation towards and away from the monitor. A passive camera, positioned on the top of the computer monitor, captures images of the scene that are analyzed by our system. Tracking is performed without the use of special image processing hardware or intrusive tracking devices or markings.

Our head movement tracking system, running on a 400 MHz Pentium II workstation, processes three to eight captured images per second. The time needed for our head movement tracking system to process a frame is a function of the type of head movement occurring in the scene. Since processing time is directly linked to the underlying hardware of the computer on which the head tracking system is executing, new hardware technology will undoubtedly reduce system processing time.

Our results showed that our head movement tracking system reliably tracks head translations. On average, our tracker fails to correctly estimate head position by one pixel in both horizontal and vertical directions and head size, i.e., width and height, by one and two pixels, respectively.

However, the performance of our tracker becomes less reliable, yet still acceptable, when tracking head rotation in the image plane. On average, our tracker estimates the tracked head rotation angle with an error of 1.4°. Our tracking system fails when there is a large amount of head rotation in depth. However, if the angle of rotation in depth is reduced as the head continues its motion, our tracking system often recovers.

Our tracking system is generally robust with respect to movements in the background and small varying facial expressions. Partial occlusion, covering less than 50% of the face with non-skin coloured objects, does not significantly affect the performance of our tracking system. Finally,

the approach we took in developing our tracking system makes it flexible enough to successfully track non-head objects.

We propose two improvements that may enhance the results of our head movement tracking system, namely the detection of head rotation in depth and calibration.

Rotation in Depth

Our model can be extended to include rotation in depth (out of plane rotation) about the x- (yaw) and y-axis (pitch). To remain in accordance with our model, we would represent such head movement using various head poses. We would image these head poses using templates that would be synthesized using a morphing technique. However, since most morphing algorithms require the capture of images representing the head at different angles, this technique would increase the number of inputs required by our tracking system. Additionally, since most morphing techniques are computationally expensive, the performance of our head movement tracking system would suffer greatly.

A simpler and perhaps less computationally expensive approach would be to grab a new master template when the correlation scores reach a certain threshold indicating that the head portrayed by the templates is no longer matching what is present in the captured image of the scene. This approach would facilitate tracking recovery when failure occurs due to head rotation in depth or due to changes in the lighting conditions of the scene.

Calibration

For an application to utilize our head tracking system, the head position, expressed in image pixel coordinates, would need to be transformed into world coordinates and units of distance. Additionally, the detected head size (width and height), expressed in pixels, would need to be mapped onto a corresponding distance, between the user and the monitor, along the z-axis. The head rotation angle, already expressed in degrees, does not require any mapping. However, if we wish our head movement tracking system to report the true head rotation angle, we would need to compute the initial head rotation angle and orientation, as portrayed in the initial head pose.

BIBLIOGRAPHY

[Azarbayejani 93] Azarbayejani, A., Starner, T., Horowitz, B., and Pentland, A., "Visually Controlled Graphics", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 15, Number 6, June 1993, pp. 602-605

[Brunelli 93] Brunelli, R., Poggio, T., "Face Recognition: Features versus Templates", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 15, Number 10, October 1993, pp. 1042-1052

[Catmull 80] Catmull, E., Smith, A. R., "3-D Transformations of Images in Scanline Order", *ACM Computer Graphics* (SIGGRAPH '80), Volume 14, Issue 3, July 1980, pp. 279-285

[Chen 98] Chen, Q., Wu, H., Fukumoto, T., Yachida, M., "3D Head Pose Estimation without Feature Tracking", *Proceedings of the Third International Conference on Automatic Face and Gesture Recognition*, 1998, pp. 88-93

[Cohen 94] Cohen, H. A., Harvey, A. L., "Stochastic Search Approach to Object Location", *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 1994, pp. 2237-2241

[Darrell 98] Darrell, T., Gordon, G., Harville, M., Woodfill, J., "Integrated Person Tracking using Stereo, Color, and Pattern Detection", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 1998, pp. 601-608

[Essa 1996] Essa, I., Basu, S. A., Darrell, T., Pentland, A., "Modeling, Tracking and Interactive Animation of Faces and Heads using Input from Video", *Proceedings of Computer Animation*, 1996, pp. 68-79

[Fua 93] Fua, P., "A Parallel Stereo Algorithm that Produces Dense Depth Maps and Preserves Image Features", *Machine Vision and Applications*, Volume 6, 1993, pp. 35-49

[Graf 96] Graf, H. P., Cosatto, E., Gibbon, D., Kocheisen, M., Petajan, E., "Multi-Modal System for Locating Heads and Faces", *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, 1996, pp. 88-93

[Haralick 93] Haralick, R. M., Shapiro L. G., *Computer and Robot Vision*, Volume II, Addison Wesley, 1993

[Harvey 91] Harvey, A. L., Cohen, H. A., "Software Speedup Techniques for Binary Image Object Recognition", *Proceedings of the International Conference on Industrial Electronics, Controls and Instrumentation*, 1991, pp. 1827-1831

[Heinzmann 98] Heinzmann, J., Zelinsky, A., "3-D Facial Pose and Gaze Point Estimation using a Robust Real-Time Tracking Paradigm", *Proceedings of the Third International Conference on Automatic Face and Gesture Recognition*, 1998, pp.142-147

[Hotz 91] Hotz, B., "Etude de Techniques de Stéréovision par Corrélation", *Rapport des Stage de Dea*, CNES, Toulouse, France, 1991

[Krattenthaler 94] Krattenthaler, W., Mayer, K. J., Zeiller, M., "Point Correlation: a Reduced-Cost Template Matching Technique", *Proceedings of the First International Conference on Image Processing*, 1994, pp. 208-212

[Li 94] Li, H., Forchheimer, R., "Two-View Facial Movement Estimation", *IEEE Transactions on Circuits and Systems for Video Technology*, Volume 4, Number 3, June 1994, pp. 276-287

[Maurer 95] Maurer, T., von der Malsburg, C., "Single-View Based Recognition of Faces in Depth", *Workshop on Automatic Face and Gesture Recognition*, 1995, pp. 248-253

[McKenna 96] McKenna, S., Gong, S., "Tracking Faces," *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, 1996, pp. 271-276

[Paeth 86] Paeth, A., "A Fast Algorithm for General Raster Rotation", *Graphics Interface*, 1986, pp. 77-81

[Pope 94] Pope, A., Ko, D., Lowe, D., Vista Library, Laboratory for Computational Intelligence, University of British Columbia, 1994, http://www.cs.ubc.ca/nest/lci/vista/vista.html

[Rekimoto 95] Rekimoto, J., "A Vision-Based Head Tracker for Fish Tank Virtual Reality: VR without Head Gear", *IEEE Virtual Reality Annual International Symposium*, 1995

[Sobottka 96] Sobottka, K., Pitas, I., "Segmentation and Tracking of Faces in Color Images", *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, 1996, pp 236-241

[Steffens 98] Steffens, J., Elagin, E., Neven, H., "PersonSpotter – Fast and Robust System for Human Detection, Tracking and Recognition", *Proceedings of the Third International Conference on Automatic Face and Gesture Recognition*, 1998, pp. 516-521

[Tang 98] Tang, C.-Y., Hung, Y.-P., Chen, Z., "Automatic Detection and Tracking of Human Heads Using an Active Stereo Vision System", *Asian Conference on Computer Vision*, 1998, Volume 1, pp. 632-639

[Takács 94] Takács, B., Wechsler, H., "Locating Facial Features Using SOFM", *Proceedings of the Twelfth International Conference on Computer Vision and Image Processing*, 1994, pp. 55-60

APPENDIX A TRACKING SYSTEM DEMO

We have developed a demo application that utilizes our head movement tracking system. Our demo application displays a graphical avatar that mimics the head movement of the person using our tracker. Figure A-1 shows the avatar.

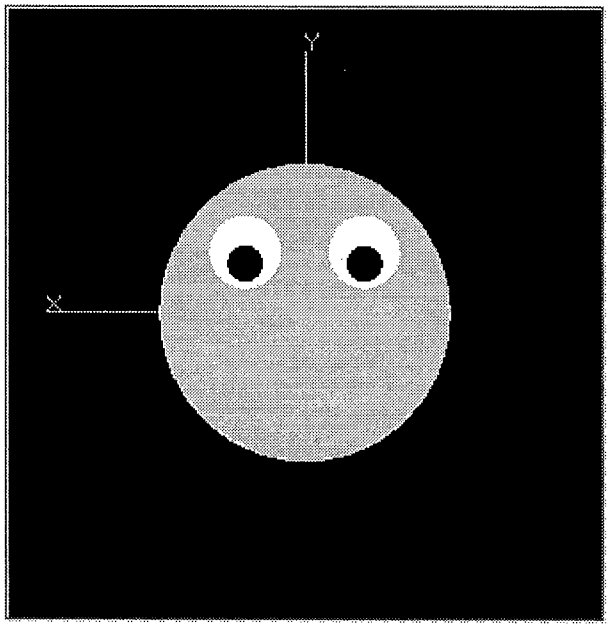


Figure A-1
Avatar of Tracking System Demo

Our tracking system demo, its hardware and software requirements and instructions on how to use it can be found at http://www.cs.ubc.ca/~lavergne/headTracking/ .