FACE DETECTION AND TRACKING USING A BOOSTED ADAPTIVE PARTICLE FILTER

by

WENLONG ZHENG

(Under the Direction of Suchendra M. Bhandarkar)

ABSTRACT

This thesis proposes a novel algorithm for integrated face detection and face tracking based on the synthesis of an adaptive particle filtering algorithm and an AdaBoost face detection algorithm. A novel Adaptive Particle Filter (APF), based on a new sampling technique, is proposed to obtain accurate estimation of the proposal distribution and the posterior distribution for accurate tracking in video sequences. The proposed scheme, termed a Boosted Adaptive Particle Filter (BAPF), combines the APF with the AdaBoost algorithm. The AdaBoost algorithm is used to detect faces in input image frames, while the APF algorithm is designed to track faces in video sequences. The proposed BAPF algorithm is employed for face detection, face verification, and face tracking in video sequences. Experimental results confirm that the proposed BAPF algorithm provides a means for robust face detection and accurate face tracking under various tracking scenarios.

INDEX WORDS: Face detection, face tracking, particle filter, boosted learning

FACE DETECTION AND TRACKING USING A BOOSTED ADAPTIVE PARTICLE FILTER

by

WENLONG ZHENG

B.S., Wuhan Technical University of Surveying and Mapping, China, 1993

M.S., Wuhan Technical University of Surveying and Mapping, China, 1996

Ph.D., Shanghai Institute of Technical Physics, Chinese Academy of Sciences, China, 1999

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2005

FACE DETECTION AND TRACKING USING A BOOSTED ADAPTIVE PARTICLE FILTER

by

WENLONG ZHENG

Major Professor:	Suchendra M. Bhandarkar
Committee:	Eileen T. Kraemer
	Kang Li

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
December 2005

DEDICATION

This thesis is dedicated to my wife, Yanfen Le, and son, Jiale Zheng.

ACKNOWLEDGEMENTS

I am deeply indebted to my major advisor, Dr. Suchendra M. Bhandarkar. It is very hard to adequately express my gratitude for his excellent guidance, sage advice, and boundless encouragement during my research in computer vision and writing of this thesis. Dr. Bhandarkar was always responsive and helpful to my requests. Dr. Bhandarkar advised me not only on research-related issues, but also on professional development. His dedication to research, teaching and service will benefit me lifelong.

I would also like to express my gratitude to my master committee: to Dr. Eileen T. Kraemer, for her invaluable advice on this thesis and endless encouragement; to Dr. Kang Li, for his suggestions on this thesis and great support.

I am also grateful to Dr. Robert Teskey and Dr. Jessica Kissinger for their generous support of my study.

I would also like to thank my fellow graduate students and friends, especially Yinglei Song, Xingzhi Luo, Yanqi Su, Xin Gao, Aura Morris, Ananda Chowdhury, Xiaochuan Yi, Siddhartha Chattopadhyay, for their help on my thesis research and friendship during my study.

I am also grateful to my family: to my wife, Yanfen Le, for her love, sacrifice and endless support; to my son, Jiale Zheng, your love is so important to me; to my parents, for encouraging my study; and for my parents-in-law for your support with family issues.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

1.1 Background

The fast evolution of computer technologies including hardware and software has advanced the state of computing machinery in the past two decades to the point where human life has been significantly improved by machine intelligence. This trend has resulted in an active development in information technology and artificial intelligence, where more friendly and efficient approaches for human computer interaction are developed based on new devices. Computer vision, which is one aspect of machine intelligence, focuses on duplication/emulation of human vision. Traditionally, computer vision systems have been utilized in specific applications such as assembly line inspections and quality control in automated manufacturing. The ever decreasing cost of computing systems and video image acquisition equipment has resulted in computer vision systems advancing towards more generalized vision applications such as face detection and face tracking techniques.

Face detection, which is the first step in any face processing system, attempts to determine whether there are any faces in a single image. If any faces exist, the processing system provides the image location and extent of each face (Yang *et al.*, 2002). Face detection is important in any human face related system, such as any fully automatic face recognition system, warning and surveillance system, or face tracking and human tracking system. Face detection algorithms can be typically extended to generic object detection and recognition (Zhao *et al.*, 2003), which leads to automatic target recognition (ATR). So far, face detection in computer vision is still a challenging task, even though it is easy for humans to perform effortlessly (Hjelmås and Low, 2001). The various face detection related problems include face localization, facial expression recognition, face recognition, face authentication, and face tracking. Traditionally, the solutions to the problems are based on image segmentation, facial feature extraction, and face verification

in the presence of complicated background. The challenges associated with face detection are contributed by changes in scale, location, orientation, pose, facial expression, occlusion, and illumination.

Face tracking aims to keep account of face in a video sequence i.e., determine if there are any faces in a single frame, and continuously estimate the locations and possibly the orientations of the faces in the video sequence in real time (Darrell *et al.*, 2000; Crowley and Berard, 1997; Edwards *et al.*, 1998). Face tracking belongs to the larger area of visual object tracking pursued by the computer vision community, where the object of interest is the face. Object tracking has been studied extensively by researchers in the context of computer vision because of many vision applications such as autonomous robots (Davison and Murray, 1998), video surveillance (Borg et al., 2005), human eye tracking (Hansen and Hammoud, 2005) and human face tracking (Nummiaro et al., 2003). Generally speaking, an image sequence, which is collected in real time, does not change rapidly from one frame to the next frame. This results in a large redundancy of object information over consecutive frames spanning a certain time interval. This redundancy can be utilized to disambiguate the appearances of the visual objects and track the individual objects. Since the human visual system may not distinguish a camouflaged object from a complicated background, the exploitation of the redundancy in a sequence of images is still regarded as a challenging problem in the computer vision community (Isard, 1998).

1.2 Research objectives

The primary objective of this thesis is to incorporate face detection with face tracking in video sequences. This thesis aims to present a new scheme for robust face detection and accurate face tracking, where face detection and face tracking can boost each other in real time. This research

will take a step in moving the conventional face tracking mechanism towards the boosted hybrid face tracking mechanism.

In order to address the general problems of face detection and face tracking, such as low detection rate, variations in lighting conditions, and partial occlusions or complete occlusions, we propose a novel scheme for face detection and tracking in this thesis by combining an AdaBoost algorithm with a new particle filtering scheme, termed an adaptive particle filter (APF). The new APF uses a new sampling technique to obtain much more accurate estimation of the proposal distribution and the posterior distribution, which improves the tracking accuracy in the video sequences. We define the combination of the AdaBoost algorithm and the APF as a boosted adaptive particle filter (BAPF). The AdaBoost algorithm is used to detect faces in the input images, while the APF is used to track the faces in the video sequences. The hybrid system of BAPF is employed for face detection, face verification, and face tracking in the video sequences. Face detection and face tracking will enhance their performance by mutual correlation in the procedure. This BAPF can provide robust face detection and accurate face tracking under some situations that the objects are severely corrupted by the occlusions.

1.3 Thesis structure

This thesis is organized into four chapters in manuscript style. Chapter 1 introduces the background and the research objectives. Chapter 2 provides a comprehensive review of the literature related to face detection and visual object tracking. Chapter 3 proposes a boosted adaptive particle filter (BAPF) for face detection and face tracking by combining an AdaBoost algorithm with a new adaptive particle filter (APF). Chapter 4 presents discussions and conclusions.

The entire thesis is structured as follows.

- *Chapter 1: Introduction*
- *Chapter 2: Literature Review*
- *Chapter 3: Face Detection and Tracking Using a Boosted Adaptive Particle Filter*
- *Chapter 4: Conclusions*

CHAPTER 2

LITERATURE REVIEW

This chapter provides a relatively comprehensive review of the literature related to face detection and tracking. It first briefly introduces the evolution of and the body of literature on face detection and tracking. Next, this chapter reviews the literature on face detection. This is followed by a literature on visual object tracking, which is generalization of face tracking. Finally, it ends with a summary of the various approaches.

2.1 Introduction

The fast evolution of computer technologies including hardware and software has advanced the state of computing machinery in the past two decades, to the point where human life has been significantly improved by machine intelligence. This trend has resulted in an active development in information technology and artificial intelligence, where more friendly and efficient approaches for human computer interaction are developed based on new devices. Computer vision, which is one aspect of machine intelligence, focuses on duplication/emulation of human vision. Traditionally, computer vision systems have been utilized in specific applications such as assembly line inspections and quality control in automated manufacturing. The ever decreasing cost of computing systems and video image acquisition equipment has resulted in computer vision systems advancing towards more generalized vision applications such as face detection and face tracking techniques. For example, computer vision systems, which are deployed in desktop or embedded systems (Pentland, 2000a; Pentland, 2000b; Pentland and Choudhury, 2000), can detect and track the face of the user in real time.

Face detection, which is the first step in any face processing system, attempts to determine whether there are any faces in a single image. If any faces exist, the processing system provides the image location and extent of each face (Yang *et al.*, 2002). Face detection is important in any

human face related system, such as any fully automatic face recognition system, warning and surveillance system, and face tracking and human tracking system. The face detection algorithms can be typically extended to generic object detection and recognition (Zhao *et al.*, 2003), which leads to automatic target recognition (ATR).

Face detection in computer vision is still a challenging task, even though it is easy for humans to perform effortlessly (Hjelmås and Low, 2001). The various face detection related problems are face localization, facial expression recognition, face recognition, face authentication, and face tracking. Traditionally, the solution to the problems is based on image segmentation, facial feature extraction, and face verification in the presence of complicated background. The challenges associated with face detection are contributed by changes in scale, location, orientation, pose, facial expression, occlusion, and illumination. The various factors affecting the images of a human face are described as follows:

1. **Pose**. A change in pose relative to the camera viewpoint affects the appearance of the face in the image.
2. **Facial expression**. The facial expression determines the appearance of the face in the image.
3. **Occlusio**n. In some cases an object may occlude the face partially or completely, thus affecting the appearance of the face in the image.
4. **Orientation**. A relative rotation about the camera's optical axis changes the appearance of the face in the image.
5. **Lighting conditions**. Different illumination conditions, such as the light source distribution and the optical and electronic characteristics of a camera, produce different face images.

Face tracking aims to keep account of face in a video sequence i.e., determine if there are any faces in a single frame, and continuously estimate the locations and possibly the orientations of the faces in the video sequence in real time (Darrell *et al.*, 2000; Crowley and Berard, 1997; Edwards *et al.*, 1998). Face tracking lies within the larger area of visual object tracking pursued by the computer vision community, where the object of interest is the face. Object tracking has been studied extensively by researchers in the context of computer vision because of many vision applications such as autonomous robots (Davison and Murray, 1998), video surveillance (Borg *et al*., 2005), human eye tracking (Hansen and Hammoud, 2005) and human face tracking (Nummiaro *et al*., 2003). Generally speaking, an image sequence, which is collected in real time, does not change rapidly from one frame to the next frame. This results in a large redundancy of object information over consecutive frames spanning a certain time interval. This redundancy can be utilized to disambiguate the appearances of the visual objects and track the individual objects. Since the human visual system may not distinguish a camouflaged object from a complicated background, the exploitation of the redundancy in a sequence of images is still viewed as a challenging problem in the computer vision community (Isard, 1998).

While the human visual system models accurate object tracking as an information-processing problem associated with robust and real-time computation, we are unaware of any current solutions using artificial intelligence which fully understand the human solution. The current solutions have to make some assumptions to simplify the tracking problem and accept less than perfect results to make progress in specific situations. Therefore, we have to segment the images of the real world into the meaningful blocks on the basis of predetermined segmentation criteria. Many approaches to this segmentation problem are proposed such as “layers” (Baker *et al.*, 1998), in which the world contains cardboard cutouts, “textures” (Malik *et al.*, 1999), in which

the world is composed of objects defined by homogeneous textures, “contours” (Blake and Isard, 1998; Li *et al.*, 2003; Rathi *et al.*, 2005), in which the world consists of objects defined by shapes with known geometric properties, and “templates” (Boccignone *et al.*, 2005; Luo and Bhandarkar, 2005), in which the world consists of objects comprising of predefined regions with known properties.

With an aim to present a comprehensive and critical review of face detection and tracking methods, this literature survey is organized as follows: In Section 2.2, we provide a detailed review of various approaches to detect faces in a single image. Section 2.3 presents a detailed survey and discussion of techniques for visual tracking in an image sequence. Finally, the summary and discussion are presented in Section 2.4.

2.2 Face Detection

In this section, we carefully survey existing techniques for face detection in a single image. We classify these techniques into three categories based on how they exploit the knowledge of the face: feature-based methods, template-based methods, and image-based methods. Since image-based methods have demonstrated better results recently compared to the other categories, we present a more detailed review of the image-based methods in this section. Some face detection methods may clearly overlap category boundaries, and hence can be classified into more than one category. For example, template-based methods typically use a face template to extract facial features, and then utilize these features for face detection (Hori *et al.*, 2004; Govindaraju, 1996; Lades *et al.*, 1993); image-based methods also use some specific features to detect a face, such as Haar-like features (Viola and Jones, 2001a; 2001b), and Gabor features (Yang *et al.*, 2004;

Zhang *et al.*, 2004). The three categories of face detection methods are described in the following:

1. **Feature-based methods**. These methods make explicit use of the knowledge of the human face and extract structural features that remain unchanged while the pose, facial expression, or illumination vary. These features can be generated from the results of low-level analysis, such as edges, gray-levels, or color. The facial features can also be obtained from a more global description of the face using information derived from face geometry.
2. **Template-based approaches**. A set of pre-defined standard face patterns or templates is constructed and stored. The templates represent a face as a whole or the facial features separately. These methods use the correlations between an input image and the given patterns to detect faces.
3. **Image-based methods**. These approaches use learning algorithms to detect faces. The learning algorithms can capture the inherent variability in facial appearance within a set of training images. Unlike the methods in category 1 and 2, the image-based methods acquire the knowledge of the human face implicitly through mapping and training schemes.

2.2.1 Feature-based methods

Typically, feature-based face detection methods are proposed to first detect facial features, which are invariant over different poses and lighting conditions. These facial features, which include the eyes, nose, mouth, eyebrows and so on, are then used to determine the existence of a face. Many methods are proposed to extract features for face detection. These methods can be

generally divided into broad categories: low-level feature analysis and high-level feature analysis. The low level feature analysis is based on the segmentation of facial features using pixel properties, such as gray scale, texture, and skin color. However, the features obtained from the low level feature analysis are usually ambiguous and sensitive to changing illumination. The high level feature analysis employs a global model of the face and facial features that incorporates knowledge of face geometry. High level feature analysis can extract better facial features than the low level feature analysis. One common problem of feature-based methods is that the features can be severely affected due to the variations in lighting conditions.

2.2.1.1 Low level feature analysis

Herpers *et al.* (1996) propose a method for facial feature detection and characteristic key-point detection using edges and lines. It first uses a first and second derivative of a Gaussian based edge detector to detect edges and lines in the underlying facial region. The line and edge detection can be performed efficiently at any orientation and scale. Then it uses three basic operations to detect the key-points of the face. The first operation searches the edges or lines in a predefined region with a predefined orientation and scale. The orientation of an edge or line in a given location is determined by the second operation through the evaluation of the maximal response of a rotated filter. The third operation tracks an edge or line by a small step in the known direction. Song *et al.* (2002) propose a method to detect objects in an edge color space (ECDS) instead of the image space. Their method assumes that the uniform-color objects and textured objects have different distribution characteristics in an ECDS. Their method first measures the color of each edge point in the edge detection phase, and then transforms the edge points into the 3D ECDS by quantizing the image space and the color space. Finally the edge

points associated with different objects are segregated spatially in the 3D ECDS rather than being detected as overlapping in the 2D image space. However, this method performs poorly in situations with significant illumination change.

Yang and Huang (1994) propose a face detection method in gray scale pyramid images. Assuming that the face image becomes approximately uniform at lower resolutions, this method searches the uniform regions starting at a lower resolution to obtain face candidates using a set of rules. Then these face candidates are further confirmed based on the prominent facial features corresponding to local minima at higher resolution. Graf *et al.* (1995) explore the gray scale behavior of faces to locate facial features. Their approach first applies morphological operations to enhance regions that have certain shapes. Based on the peak value of the gray scale histogram of the processed image, it then applies the adaptive thresholding algorithm to generate two binary images. Finally, their approach evaluates the combinations of connected components in the binary images to determine the existence of the face.

Huang and Trivedi (2004) develop a framework for face detection and tracking using skin color and elliptical edge contours. It detects skin blobs if the color of the image region is above a predefined threshold and obtains the face candidates. In the meantime, it also detects the face candidates by comparing the extracted edge contours with a predefined ellipse. The final face candidates are generated using a combination of color and edge features. Finally the face candidates are verified using a distance metric in a reduced dimensional feature subspace computed via principal component analysis (PCA) to remove non-faces. However, most skin color models are typically not very robust to significant variations of the lighting conditions. To address this problem, McKenna *et al.* (1998) propose an adaptive color mixture model to track faces in varying lighting conditions. Their approach uses a stochastic model to approximate the

color distribution of an object and adapts the model to the changes in lighting conditions. Naseem and Deriche (2005) present a color-based face detection method that avoids the effect of luminance changes. Using the chromatic or pure color space, a Gaussian distribution model for the skin colors is developed to obtain a skin color likelihood image. This likelihood image is then converted into a binary image using an adaptive thresholding algorithm. Finally, a template matching method is used to estimate the regions with the desired facial properties.

2.2.1.2 High level analysis

Huang *et al.* (2004) propose a face detection method that combines multiple facial features. Four classifiers are designed based on four feature-based representations: intensity, gradient, Gabor (Huang *et al.*, 2003), and 2D Haar wavelet (Tokunaga *et al.*, 2002). The intensity features are obtained after the preprocessing phase consisting of linear illumination correction and histogram equalization. The gradient direction features are extracted from the local images using the Sobel operator. Then the gradient vector is decomposed into its components along the eight chain-code directions. A 2D Gabor filter is used in the image space and the spatial frequency domain to extract the features. Two types of 2D Haar basis functions are used to characterize the changes in intensity along the horizontal and vertical directions resulting in the 2D Haar wavelet features. A polynomial neural network (PNN) is employed for each representative feature model to assign a face likelihood score to each face candidate. The output scores from the four PNNs are averaged to get a final score for each face candidate.

Wang and ertMariani (2000) propose a filter-based method for face detection and facial feature localization. Their approach first uses multi-scale filters to obtain the pre-attentive features of the objects in the image. Three representative models are employed in this method: a

structure model, a texture model, and a feature model. Using the geometric patterns of the underlying facial components, the structure model is used to group the pixels into face candidates. The texture and feature models are used to evaluate the face candidates. The texture model validates the gray scale or color similarities of face candidates using face models. The feature model compares the region features to specific facial features using the eigen-eyes method.

2.2.2 Template-based approaches

Template-based face detection methods use a standard face pattern, which is predefined or parameterized by a function. The similarities between the standard patterns and the local image regions are estimated for the face candidate and its various components. The decision regarding the face candidates are made based upon the values of these similarities. Based on the previous work of Lades *et al.* (1993), Wiskott *et al.* (1997) propose an elastic bunch graph matching (EBGM) method for face recognition. In this method, faces are represented by labeled graphs using a Gabor wavelet transform. A set of M individual model graphs is combined into a stack-like structure, called a face bunch graph (FBG). Once the initial FBG is generated manually, the FBG of new images can be generated automatically by the EBGM procedure. A graph similarity measure between an image graph and the FBG corresponding to an identical pose is computed to match model FBG to a new image. After obtaining model graphs from an image database and image graphs from the probe images, recognition is performed by selecting model FRG corresponding to the highest similarity value resulting from the comparison of an image graph to all the model graphs.

Kwon and Lobo (1994) present a face detection method based on snakes and templates. A modified *n*-pixel snake is used to find and remove small curve segments in the image. An ellipse is used to approximate each face. A Hough transform on the remaining snakelets is employed to search for a predominant ellipse. Each face candidate is evaluated by a method similar to the deformable template matching method. The final decision for each face candidate is provided based on the number of matching facial features found in the image and their proportions. Gunn and Nixon (1996) present a method for face boundary detection using a dual snake configuration based on dynamic programming to locate a global energy minimum. This method uses dynamic programming to extract the inner face boundary, whereas it uses a conventional surface normal-driven technique to extract the outer face boundary. Samal and Iyengar (1995) propose a method for face localization using silhouettes as templates. Principal component analysis (PCA) of the face examples is used to obtain a set of basis face silhouettes. These eigen-silhouettes in combination with a Hough transform are then utilized to localize the faces.

2.2.3 Image-based methods

Image-based face detection methods have demonstrated excellent results recently among all face detection methods. Image-based methods typically depend on techniques from machine learning and statistical analysis to search for the discriminating characteristics of face and non-face images. In general, these characteristics are modeled using known statistical distributions or a combination of known discriminant functions, which are then used for face detection. Much research has been conducted in image-based methods resulting in well known techniques, such as AdaBoost (Viola and Jones, 2001a; Viola and Jones, 2001b; Wang *et al.*, 2004), FloatBoost (Li *et al.*, 2002), S-AdaBoost (Jiang and Loe, 2003), neural networks (Rowley *et al.*, 1996;

Curran *et al.* ,2005), Support Vector Machines (SVM) (Osuna *et al.*, 1997; Shih and Liu, 2004), Hidden Markov Models (Rabiner and Jung, 1993), and the Bayes classifier (Schneiderman and Kanade, 1998; Schneiderman, 2004).

2.2.3.1 Boosting Learning Algorithms

Based on previous work of Tieu *et al.* (2000) and Schneiderman (2000), Viola and Jones (2001a; 2001b) propose a robust face detection algorithm, which can detect faces in a rapid and robust manner with a high detection rate. It presents three contributions for face detection: the integral image, a strong classifier comprising of weak classifiers based on the AdaBoost learning algorithm, and an architecture comprising of a cascade of a number of strong classifiers. The system of Viola and Jones (2001a; 2001b) employs an integral image comprising of Haar-like features for effective feature extraction from a large feature set. Lienhart and Maydt (2002) provide a set of Haar-like features for AdaBoost, as shown in Figure 2.1. In the boosting procedure as shown in Figure 2.2, AdaBoost first learns effective features from a large feature set. Second, it constructs a set of weak classifiers, each of which is composed of a feature, a threshold and a parity. Third, it generates a strong classifier based on the above weak classifiers, as shown in Figure 2.2. Each iteration will generate a weak classifier. After all iterations, it will result in T weak classifiers. These T weak classifiers are combined into a strong classifier using a weighted linear combination. The system of Viola and Jones (2001a; 2001b) uses a cascade of strong classifiers to improve the detection rate with efficient computation, as shown in Figure 2.3. The idea is to construct smaller and efficient classifiers based on the sub-windows within the image. The simpler and faster classifiers will reject the negative sub-windows. A large number of negatives are rejected by the initial classifier with minimal processing. Additional negatives are

eliminated by subsequent layers while requiring additional computation. The number of sub-windows is supposed to be reduced rapidly after several stages of processing.

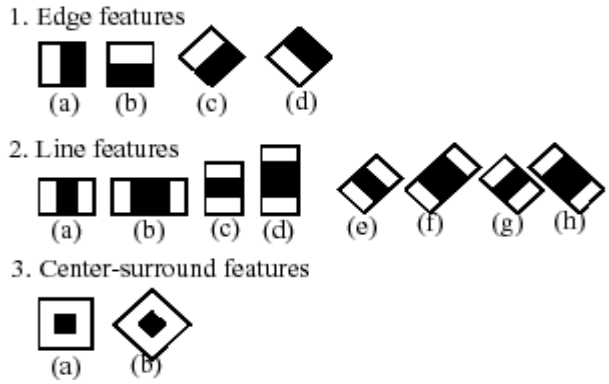


Figure 2.1 Haar-like features used in the AdaBoost algorithm (Lienhart and Maydt, 2002)

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:
  1. Normalize the weights,
     $$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$
     so that w_t is a probability distribution.
  2. For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t, $\epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$.
  3. Choose the classifier, h_t, with the lowest error ϵ_t.
  4. Update the weights:
     $$w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$$
     where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The final strong classifier is:
  $$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$
  where $\alpha_t = \log \frac{1}{\beta_t}$

Figure 2.2 The AdaBoost learning algorithm (Viola and Jones, 2001a)

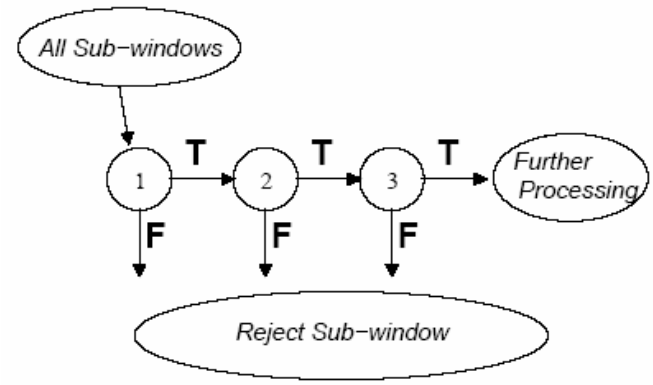


Figure 2.3 The cascade structure of Viola and Jones's system (2001a)

Li *et al.* (2002a) propose the FloatBoost algorithm, an improved version of AdaBoost, for learning a boosted classifier for obtaining the minimum error rate. It uses a backtracking mechanism to improve the detection rate after each iteration of AdaBoost procedure. In the boosting procedure, FloatBoost performs deletions of weak classifiers that are ineffective based on the error rate. Thus a strong classifier containing a set of weak classifiers is used to improve the classification error. But this method needs more training time than AdaBoost since it entails an additional search on the current weak classifiers. Li *et al.* (2002a) also proposed a multi-view face detection system, which is illustrated in Figure 2.4. This structure uses the coarse-to-fine strategy and generalizes the cascade detection system proposed by Viola and Jones (2001a). It consists of three levels. Each level except the top level contains more than one detector. The final result is obtained by merging the combination of the detectors at the bottom level.

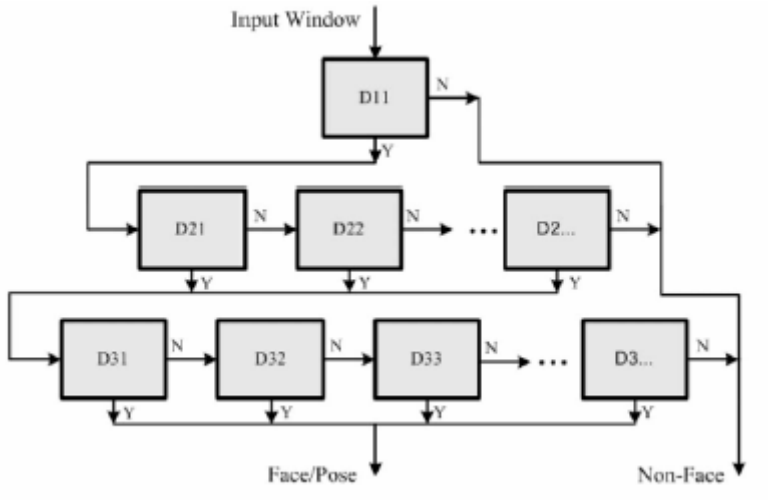


Figure 2.4 The structure of the multi-view face detection system of FloatBoost (Li *et al.*, 2002a)

Jiang and Loe (2003) propose S-AdaBoost, a variant of AdaBoost for handling outliers in pattern detection and classification. S-AdaBoost divides the input space into sub-spaces based on the Divide and Conquer Principle. Dedicated classifiers are used to process the sub-spaces. Finally, a specific classifier handles the combination of the outputs of the dedicated classifiers. Since this method uses different classifiers in different phases, its computation and effectiveness are not satisfactory. Zhang *et al.* (2004b) propose a face detection method based on boosting in hierarchical feature spaces. They assume that global features derived from Principal Component Analysis can be used in the later stages of boosting to further improve the detection rate. However, it needs more computation time for extracting global features.

Wang *et al.* (2004) propose a real-time facial expression recognition system with AdaBoost. In the face detection phase, this system uses the AdaBoost algorithm proposed by Viola and Jones (2001a; 2001b). In the facial expression recognition phase, the expressions are learned from the boosting of Haar-like feature-based look-up-table type weak classifiers. Likewise, Wu *et al.* (2004) propose a rotation invariant multi-view face detection method based on real AdaBoost. The faces are grouped based on the appearance from different views, and then weak

classifiers are learned from the individual groups to construct a confidence-rated look-up-table for Haar-like features. This method uses a view-based detector that can deal with facial profiles and 360-degree rotated faces. A nested-structured cascade is proposed in this method, as illustrated in Figure 2.5. It consists of common weak classifiers of Viola and Jones's system (2001a) and multiple layers of nested weak classifiers. Each layer is a linear network of common weak classifiers and outputs a confidence value for further processing in the following layer. Yang *et al.* (2004) provide a face recognition method with AdaBoosted Gabor features. First, AdaBoost selects a small set of effective Gabor features from a large database of images. Then a strong classifier incorporating a few hundred of weak classifiers with Gabor features can distinguish the difference between two face images. Zhang *et al.* (2004a) also propose a similar method as Yang *et al.* (2004).

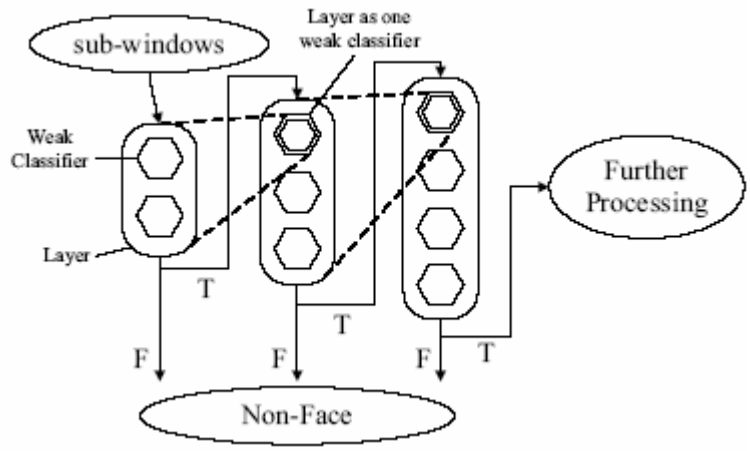


Figure 2.5 View-based detector diagram (Wu *et al.* 2004)

2.2.3.2 Neural Network Learning Algorithms

Rowley *et al.* (1996; 1998) has done the most significant research among all face detection methods based on neural networks, as shown in Figure 2.6. This method consists of two major components: a set of multilayer neural networks and a decision making module. The multilayer

neural networks are used to learn the face and non-face patterns from the training sets consisting of face and non-face images, and then applied to detect faces. The decision making module is used to generate the final decision on the basis of the combination of multiple detection results. The first component receives a 20×20 pixel image region and then outputs a score from -1 to 1, where -1 denotes non-face and 1 denotes face. Using multi-resolution processing, the neural network can detect a face of size larger than 20×20 pixels. The decision making module merges the overlapping detection results from the outputs of the multiple networks and makes a final determination. One drawback of this method is that only upright frontal faces can be detected. Although Rowley further improves the method to detect rotated face images, the result is not promising because of its lower detection rate.

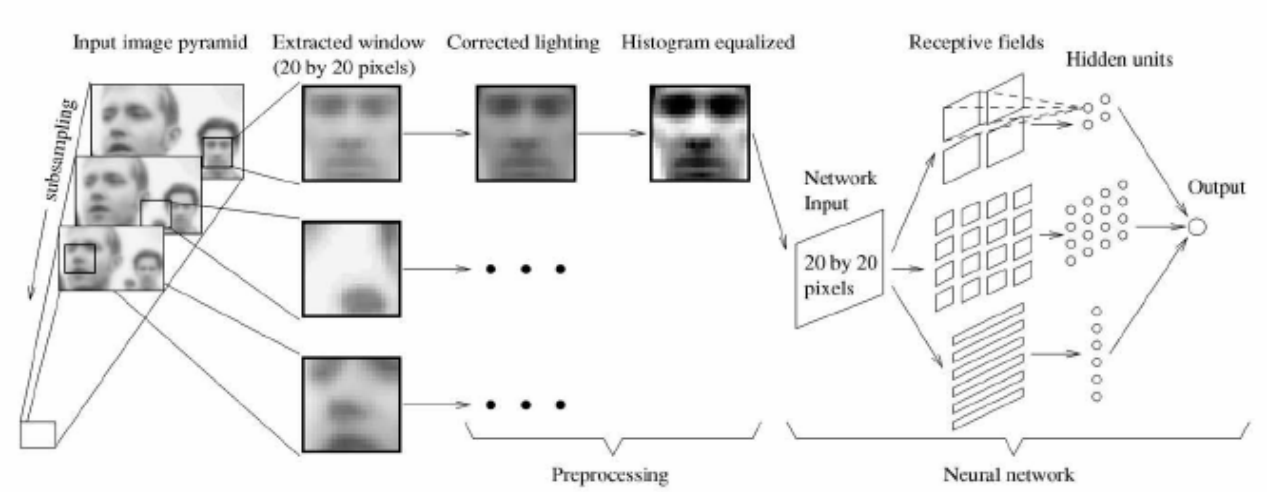


Figure 2.6 Face detection using neural networks (Rowley *et al.*; 1998)

Curran *et al.* (2005) extends the work of Rowley *et al.* (1998) to address the problem of face detection under gross variations. Féraud *et al.* (2001) propose a face detection method based on a neural network model, which is called the Constrained Generative Model (CGM). This approach computes the distance of the input subwindow to the set of faces to estimate the probability of an input subwindow to be a face. The distance is obtained based on a projection of a pixel in the

input image space on the set of faces. The face detector based on CGM and Multilayer Perceptron (MLP) consists of four stages, where the last filter outputs the final decision. The major disadvantage of this method is that it requires non-face samples to model the projection, which entails more computation time.

2.2.3.3 Support Vector Machines

Support Vector Machines (SVMs) use structural risk minimization to minimize the upper bound of the expected generalization error (Osuna *et al.*, 1997), while most other learning methods such as neural networks and Bayesian networks are based on minimizing the training error. The SVM is a linear classifier which computes a separating hyperplane to minimize the expected generalization error. The hyperplane is defined by a weighted combination of a small subset of support vectors. The optimal hyperplane is approximated by solving a linear constrained quadratic programming problem. The major disadvantages of SVMs are its computation time and high memory requirement.

Terrillon *et al.* (2000) analyze the performance of SVMs in static color images and propose a face detection method. Their approach combines the skin color-based image segmentation with the application of SVMs to the invariant features derived from a generalization of the Orthogonal Fourier and Mellin Moments (OFMMs). Shih and Liu (2004) propose a face detection method combining Discriminating Feature Analysis (DFA) and SVMs. This approach uses both temporal and skin color information to locate the regions of interest in the input image. An SVM classifier and Bayesian analysis are applied to the features extracted by DFA for face detection.

2.2.3.4 Other Learning Algorithms

Hidden Markov Models (HMMs) assume that face and non-face patterns can be characterized as a parametric random process. These parameters can be obtained using a well-defined estimation procedure (Rabiner and Jung, 1993). The aim of training an HMM is to estimate the appropriate parameters in an HMM model to maximize the probability of observing the training data. Schneiderman and Kanade (1998) presente a naive Bayes classifier, which exploits the estimation of the joint probability of local appearance and position of a face pattern at multiple scales. However, the performance of a naive Bayes classifier is poor. To solve this problem, Schneiderman (2004) propose a restricted Bayesian network for object detection. This method searches the structure of a Bayesian network-based classifier in the large space of possible network structures. The final structure is computed via constrained optimization of two cost functions where estimates and evaluation are precomputed: a localized error in the log-likelihood ratio function for the structure and a global classification error for the final choice of the structure. Park *et al.* (2005) propose a Face Probability Gradient Ascent (FPGA) method to evaluate the optimal position, scale, and rotation variants of each face. Based on the probability that the partial image corresponds to a face image, the proposed FPFA approach uses a gradient-based iterative search to determine the objective function to model the underlying probability density function.

2.3 Visual Tracking

Object tracking has been studied extensively in the context of computer vision because of many vision applications such as autonomous robots (Davison and Murray, 1998), video surveillance (Borg et al., 2005), human eye tracking (Hansen and Hammoud, 2005) and human face tracking

(Nummiaro et al., 2003) that use tracking algorithms extensively. Object tracking in complex situations needs to deal with uncertainty and error (Intille et al., 1997). Therefore many techniques have been developed to solve the problem of object tracking. We classify visual object tracking methods into three (possibly overlapping) broad categories.

1. **Image-based tracking**. Image-based tracking methods extract the generic features and then group them based on high-level scene information.
2. **Contour-based tracking**. Contour-based tracking assumes that the object is defined by boundaries with some properties. It usually requires shape models (contours), dynamical contour models and other image measurements during the tracking process.
3. **Filtering-based tracking**. The Kalman filter and the particle filter are investigated in this category. Kalman filtering deals with the tracking of shape and location over time in linear dynamic systems. Particle filtering, on the other hand, is not restricted to linear systems. The basic idea of the particle filter is to approximate the posterior density using a recursive Bayesian filter using a set of particles with assigned weights.

2.3.1 Image-based Tracking

Many techniques have been developed in the last decade for visual object tracking. Image-based tracking methods obtain generic features from the images and then combine them based on the high-level scene information. Intille *et al.* (1997) propose a blob-tracker for human tracking in real time. The background is subtracted to extract foreground regions. The foreground regions are then divided into blobs based on color. These blobs are clustered using proximity and velocity into groups such that a single group of blobs belongs to a single person. This approach runs fast, but the major disadvantage is that it merges blobs when the objects in the scene

approach each other. To address this problem, Huang and Trivedi (2004) develop a framework for face detection and tracking using skin color and elliptical edges. This approach detects skin blobs if the color of the area is above a threshold in a color space, and detects the face candidates by comparing the detected edges with a predefined ellipse. The face candidates are verified using a distance measure in a feature space determined via principal component analysis (PCA). A continuous density Markov hidden model (CDHMM) (Rabiner, 1989) is used for face tracking, in which face orientations are estimated via maximum *a posteriori* (MAP) computation in real time. However, skin color models are not effective in the presence of significant variation in the lighting conditions.

Bhandarkar and Luo (2005) present a multi-color model for background updating in surveillance and monitoring systems. It uses multiple color clusters to represent the background at the pixel level. The background updating scheme updates the mean and variance of each color cluster with currently observed color values. The advantage of this method is that it is robust and computationally efficient for real-time monitoring systems. However, this method will give wrong results when the background color does not remain constant for a period of time. Gan *et al.* (2005) propose an object tracking method using the level-set method. This approach explores both local and global features of the image sequences to obtain better tracking results for objects with a non-uniform energy distribution. First, an initial segmentation of the objects is performed using a semi-automatic approach. Second, tracking techniques, which are based on level set methods or geometric partial differential equations (PDEs), are applied to segment the objects in other video sequences. Third, a given image is deformed according to the PDEs, and the desired result is deemed to be the steady state solution of this PDE. The process of solving the PDE can

be regarded as that of minimizing a predefined energy function. However, this method has the limitation of expensive computation.

Chen and Tiddeman (2005) present a facial feature tracking method using skin color filtering. This approach utilizes a 3D facial feature model to estimate the 3D pose of a human head. Skin color filtering is first employed to detect a face in the normalized YCbCr color space and the HSI color space. The Lucas-Kanade (LK) algorithm (Lucas and Kanade, 1981) is then applied to track the feature points. The LK tracking algorithm detects the motion based on the optical flow. However, this method only handles frontal face views, since the disappearance of certain features in a multi-view face scene makes the tracking fail. Thome and Miguet (2005) propose a human tracking method based on the construction of a 2D human appearance model, which provides discriminative features that capture both color and shape properties of the different limbs. This method, however, performs poorly when faced with significant changes in the ambient lighting conditions.

2.3.2 Contour-based Tracking

Contour-based tracking assumes that the object is defined by boundaries with known properties. It relies on shape models (contours), dynamical models and the image measurements. Kass ***et al.*** (1987) present a tracking technique featured as "Snakes" to perform robust segmentation and region tracking by modeling an object using the contour defining the object outline which is insensitive to lighting changes, and applying smoothness constraints on the contour curvature and the object motion. This tracking mechanism is more general than modeling entire objects, and also more clutter-resistant than tracking techniques based on signal-processing or similar low level analysis applied to features such as corners or edges. Many researchers (Blake and Isard,

1998; MacCormick, 2000) have adopted and extended this idea of active contour modeling in the content of tracking.

Blake and Isard (1998) develop a probabilistic active contour framework for visual tracking where objects are represented by B-spline curves in an image sequence. A given object is defined by its contour outline modeled as a B-spline. Specifically, suppose the coordinates of the B-spline control points are $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$, then the B-spline is a parameterized curve $(x(s), y(s))^T$ defined on an interval of the real line: $\begin{pmatrix} x(s) \\ y(s) \end{pmatrix} = B(s) \begin{pmatrix} \vec{x} \\ \vec{y} \end{pmatrix}$, where $B(s)$ is a $2 \times 2n$ matrix whose entries are polynomials in *s*, and $\vec{x}$, $\vec{y}$ are $n \times 1$ column vectors representing the *x*- and *y*-coordinates of the control points respectively. Any such B-spline is called a contour. Figure 2.7 illustrates an example of a face-like contour with 13 control points.

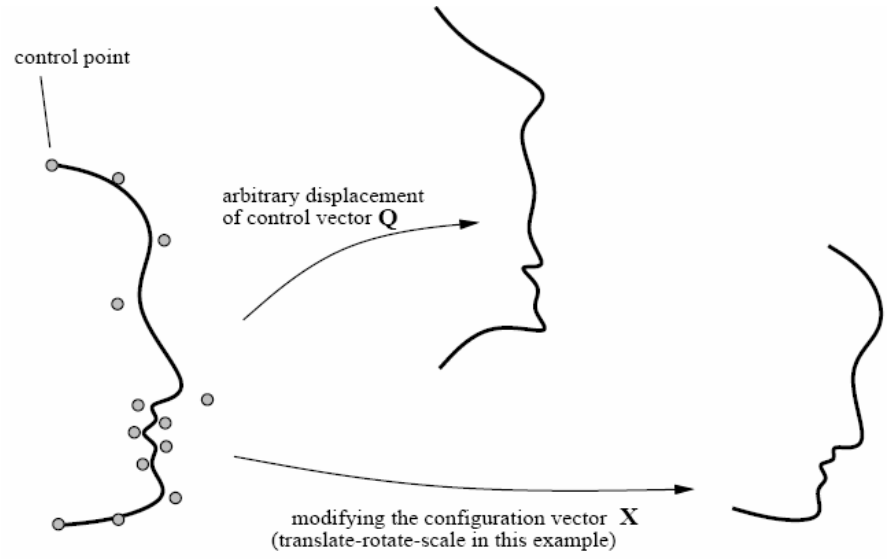


Figure 2.7 A B-spline contour specified by control points (Isard, 1998)

In practice, it is desirable to restrict the configuration of the spline to a shape-vector determined by the configuration vector **X** and described by $\begin{pmatrix} \vec{x} \\ \vec{y} \end{pmatrix} = \mathbf{W} \cdot \mathbf{X} + \mathbf{Q}$, where **W** is the $2n \times d$ shape matrix, configuration vector **X** is a $d \times 1$ column vector, **Q** is a $d \times 1$ vector called

the object template. Generally, the shape space allows affine deformations of the template **Q**, in a space of rigid and non-rigid deformations as shown in Figure 6. Isard and Blake (1998) apply the B-spline representation to contours of objects and present the Condensation algorithm. The Condensation algorithm uses the affine group parameters as the state vector, learns a dynamical model for these parameters, and employs a particle filter to estimate these parameters. However, this approach cannot deal with local deformations of the deforming object because it only tracks the affine parameters. Following the idea of Blake and Isard (1998), Wu *et al.* (2003) propose a generative model approach for contour tracking in the presence of non-stationary clutter. This method uses a proposed dynamic Bayesian network to deal with occlusions via explicit modeling and inference. However, this method is computationally very intensive.

A more recent development in the contour-based tracking is the use of the level set technique (Sethian, 1989), which is an implicit representation of contours. To segment a shape using level sets, this technique deforms an initial guess of the contour shape until it reaches the minimum of an image-based energy functional. Some recent representative research in tracking using level set techniques include the works of Yezzi and Soatto (2003), Jackson *et al.* (2004), and Rathi *et al.* (2005). Yezzi and Soatto (2003) propose a definition for motion and shape deformation for a deforming and moving object. A finite dimensional group action, such as a Euclidean or Affine group is used to parameterize the motion of the object. The shape deformation is defined by the total deformation of the object contour (infinite-dimensional group) modulo the finite-dimensional motion group. Tracking is then described by a trajectory defined on the finite-dimensional motion group. This method depends only on the observed images for tracking and does not make any use of the prior information on the dynamics of the group action or of the deformation. Thus it collapses when there is an outlier observation or when there is occlusion.

To solve this problem, Jackson *et al.* (2004) propose a generic local observer to combine prior knowledge of the system dynamics in the tracking framework, where a constant velocity prior is imposed on the group action and a zero velocity prior is imposed on the contour. A joint minimization of the energy is used to achieve the observed value of the group action and the contour. However, this approach has two disadvantages: intensive computation and instability in the case of a nonlinear system. A joint minimization over the group action and the contour at each time stamp is computationally intensive, and it is hard to choose an observer to guarantee stability for nonlinear system. To address these problems, Rathi *et al.* (2005) formulate geometric active contours as a parameterization technique to deal with the deformable objects. This approach combines a prior system model with an observation model, uses a particle filter to estimate the conditional probability distribution of the group action and the contour at each time step. However, this method still has two major problems. Since this method has to include some kind of predefined shape information, one problem is the difficulty to track highly deformable objects whose shapes are not all predefined. Another problem is the poor performance when the tracked object is completely occluded for many frames.

2.3.3 Filtering-based Tracking

2.3.3.1 Kalman filter-based tracking

The tracking of object shape and location over time is well handled by the Kalman filter in the case of linear dynamic systems (Rehg and Kanade, 1994). The Extended Kalman Filter (EKF) is the extension of the Kalman filter to a nonlinear but unimodal process where non-linear behavior is approximated by local linearization (Jebara *et al.*, 1998). Zhao *et al.* (2004) develop a tracking system using an ellipsoidal model for the gross human shape. The shape parameters are tracked

using a Kalman filter. This method uses an appearance model. The tracking mask of the model is an ellipse rather than a bounding rectangle, however, this model still suffers from the drawback of wrong updates.

Girondel *et al.* (2004) present a method for tracking multiple persons using Kalman filtering and face detection. They use a region-based strategy similar to that of Zhao *et al.* (2004). Face detection restricts the applicability of the method to viewpoints where skin color-based segmentation may be performed. This method uses a Kalman filter to overcome the occlusion problem. However, only partial Kalman filtering is used because several image measurements are likely to result in misses. This approach takes advantage of a Kalman filter only in a predictive mode, thus restricting it to a simple motion model.

Luo and Bhandarkar (2005) propose a multiple object tracking method combining the Kalman filter with elastic matching. A region-based model is used to model the objects in a network of grids. Each grid encodes the color information and the feature points of the object. The grid network contains the contour and the object shape information. This method uses a Kalman filter to predict the velocity of the tracked object, and an elastic matching algorithm to localize the objects defined by the object model. The proposed tracking model consists of three sub-models: the object model, the velocity estimation model, and the velocity measurement model. This approach has the advantage of being able to track both rigid and deformable objects. Another advantage is that the elastic matching algorithm can provide good tracking when the Kalman filter results in wrong prediction. However, this approach is restricted to situations where the occlusion is relatively short, especially if the motion model and object model are simple.

2.3.3.2 Particle filter-based tracking

Various particle filter-based approaches have been developed to improve the tracking performance. It is widely accepted that the particle filter has tracking performance superior to that of Kalman filter (Chang *et al.*, 2005). In this context, particle filtering presents a robust object tracking framework without being restricted to linear systems. Particle filters, also known as sequential Monte Carlo filters, have been widely used in visual tracking to address limitations arising from non-linearity and non-normality of the motion model (Li *et al.*, 2003; Okuma *et al.*, 2004). The basic idea of the particle filter is to approximate the posterior density using a recursive Bayesian filter based on a set of particles with assigned weights. For each frame of an image sequence in the visual tracking framework, a particle filter usually consists of three steps: sampling, weighting, and selection. A set of particles is drawn from a proposal distribution in the sampling step. In the weighting step, each particle is then weighted based on the ratio of its true probability to its approximated probability using the proposal distribution. After outputting the particle states and weights for the posterior density estimation, the particles are selected (re-sampled) according to the estimated posterior density to obtain a uniform weight distribution in the selection step.

The Condensation algorithm, a simple particle filter, proposed by Isard (1998) is designed to solve the tracking problems arising from non-linearity and non-normality of the motion model. In the sampling step, the Condensation algorithm uses a simple proposal distribution to draw a set of particles, which defines the conditional distribution on the particle state in the previous frame. This proposal distribution does not make use of the information from the current frame. The latest observation is only applied in the weighting step rather than in the sampling step. As a result, it generates only a very rough estimation of the posterior distribution and also needs a

large number of particles to represent the posterior distribution. MacCormick and Isard (2000) present a partitioned sampling technique to solve this problem, which requires that the state-space be sliced. Doucet *et al.* (2001) present an optimal proposal distribution (OPD) for state estimation of jump Markov linear systems, which is used to recursively compute optimal state estimates based on the selection of the minimum value of the variance of the weights. However, this approach is computationally very intensive. To address the above problems, Li *et al.* (2003) propose a Kalman particle filter (KPF) and an unscented particle filter (UPF) to improve the particle sampling in the context of visual contour tracking. This approach makes use of a Kalman filter or an unscented Kalman filter to incorporate the current observation. The Kalman filter or the unscented Kalman filter can steer the set of particles to regions of high likelihood in the search space, and thus reduce the number of particles. This approach also uses the local linearization of the OPD with the Gaussian distribution to result in less intensive computation compared to the original OPD. However, this approach does not handle the occlusion problem, and the Kalman filter and the unscented filter may result in wrong updates due to complicated motions of the objects in the scene.

To address the occlusion problem, Wang and Cheong (2005) propose a particle filter with a Markov random field (MRF) based representation of the tracked object within a dynamic Bayesian framework. This method transforms the object into a composite of multiple MRF regions to improve the modeling accuracy. Each MRF region is able to switch labels between foreground or background, thus the occlusion can be accurately modeled by exploiting the flexibility of the observation model. However, this approach has two main problems: one is the intensive computation involved in the MRF modeling; the other is that it is hard to get the stable and compact regions in the MRF implementation. Using the data association techniques, Chang

et al. (2005) present a kernel particle filter to improve sampling efficiency for multiple object tracking. This scheme invokes kernels to continuously approximate the posterior density, where the kernels for object representation and localization are allocated based on the gradient derived from the kernel density. Since this method assumes that the objects being tracked are indistinguishable from each other in terms of the observation model, it is difficult to handle situations in which the motion pattern of objects in one group changes drastically. This is a general problem with most data association techniques. Rathi *et al.* (2005) formulate geometric active contours as a parameterization technique to deal with the deformable objects. This approach first incorporates a prior system model with an observation model, and then uses a particle filter to estimate the conditional probability distribution of the group action and the contour at each time step. However, this method performs poorly when the tracked object is completely occluded for many frames.

Isard and MacCormick (2001) propose a Bayesian multiple-blob tracker (BraMBLe), an early implementation of a particle filter in which the number of tracked objects can vary during tracking. Based on the theory of Bayesian correlation, this approach develops a robust observation model that precisely represents the likelihood of differing numbers of objects being tracked. However, this approach relies on modeling a fixed background to identify foreground objects. To address this problem, Okuma *et al.* (2004) relax the assumption of a fixed background to handle real image sequences, where the background may vary. Based on the work of Vermaak *et al.* (2003), Okuma *et al.* (2004) propose a boosted particle filter (BPF) for multiple object detection and tracking, which interleaves the AdaBoost algorithm with a simple particle filter (the Condensation algorithm). This approach uses the AdaBoost algorithm to learn models of the objects, and these models are then used to steer the particle filter. The proposal

distribution of the particle filter incorporates information from AdaBoost in the current observation, which relieves the sampling/estimation problem in the Condensation algorithm. However, this mixture method does not present a systematic way of incorporating object models to guarantee accurate approximation of the proposal distribution, and also does not address the occlusion problem.

2.4 Summary

We aim to provide a comprehensive review of the literature related to face detection and visual tracking, and to categorize the various approaches proposed in over 90 papers. The face detection methods are also divided into three major categories, and the visual tracking approaches are also divided into three major categories. The face detection methods and the representative research works are summarized in Table 2.1. Table 2.2 summarizes the visual tracking approaches and the representative research works. However, note that some methods can be classified into more than one category.

From Table 2.1 and the survey on face detection in Section 2.2, we can recognize that significant progress has been made in face detection in the last decade. Face detection has evolved from methods that use simple features and heuristics to methods that use multiple complex features, probability analysis and learning algorithms. Due to the variation in lighting conditions, orientation, pose, facial expression, facial hair, and occlusion, face detection is still a challenging problem in the computer vision research community. Although visual object tracking has made large progress in the last decade as seen in Table 2.2 and the literature review in Section 2.3, there is still work to be done to deal with complex motions of scene objects, complex backgrounds, deformable shapes, and cases of complete occlusion. Currently many

researchers have focused on the statistical analysis of the tracked objects, resulting in statistical models for the motion and appearances of the scene objects, to handle the above challenging problems. Interestingly, there are only a few works that deal with the interleaving of face detection and visual tracking. We can expect that the research on robust face detection and tracking will still remain an active research area, since the research addresses several difficult problems dealing with general object detection, tracking and recognition.

To address the problems associated with face detection and visual tracking reviewed in Section 2.2 and Section 2.3, we propose a novel scheme for face detection and tracking in this thesis by combining the AdaBoost algorithm with a new particle filtering scheme, called an adaptive particle filter (APF). The new APF uses a novel sampling technique to obtain much more accurate estimation of the proposal distribution and the posterior distribution. We term the combination of AdaBoost and APF as a boosted adaptive particle filter (BAPF). First, the AdaBoost algorithm is used to detect faces in an input image, and the BAPF algorithm is then used for face verification and tracking in real video sequences. The BAPF algorithm can obtain good tracking results in situations where the objects are severely occluded.

Table 2.1 Face detection methods and their representative works

Category		Characteristics	Works
Feature-based methods		Facial features with edges and lines	Herpers *et al.* (1996) Song *et al.* (2002)
		Gray scale	Yang and Huang (1994) Graf *et al.* (1995)
		Skin color and elliptical edges	Huang and Trivedi (2004) McKenna *et al.* (1998) Naseem and Deriche (2005)
		Multiple facial features	Huang *et al.* (2004) Wang and ertMariani (2000)
Template-based methods		Elastic bunch graph matching	Wiskott *et al.* (1997)
		Snakes and templates	Kwon and Lobo (1994) Gunn and Nixon (1996)
		Silhouettes	Samal and Iyengar (1995)
Image-based methods	Boosting Learning	AdaBoost	Viola and Jones (2001a; 2001b) Lienhart and Maydt (2002) Wang *et al.* (2004)
		FloatBoost	Li *et al.* (2002a; 2002b)
		S-AdaBoost	Jiang and Loe (2003)
		AdaBoost and PCA	Zhang *et al.* (2004b)
		AdaBoost with look-up-table type weak classifiers	Wu *et al.* (2004)
		AdaBoost with Gabor features	Yang *et al.* (2004)
	Neural Network (NN)	Multilayer neural networks	Rowley *et al.* (1996; 1998) Curran *et al.* (2005)
		NN and Constrained Generative Model	Féraud *et al.* (2001)
	Support Vector Machines (SVM)	SVM with polynomial kernel	Osuna *et al.* (1997)
		SVM with Orthogonal Fourier and Mellin Moments (OFMM)	Terrillon *et al.* (2000)
		SVM with Discriminating Feature Analysis	Shih and Liu (2004)
	Other Learning Algorithms	Hidden Markov Model (HMM)	Rabiner and Jung (1993)
		Naive Bayes classifier	Schneiderman and Kanade (1998)
		Restricted Bayesian network	Schneiderman (2004)
		Face Probability Gradient Ascent (FPGA)	Park *et al.* (2005)

Table 2.2 Visual tracking methods and their representative works

Category		Characteristics	Works
Image-based tracking		Blob-tracker	Intille *et al.* (1997
		Skin color and elliptical edges	Huang and Trivedi (2004)
		Continuous density Markov hidden model (CDHMM)	Rabiner (1989)
		Multi-color model	Bhandarkar and Luo (2005)
		Level-set method or geometric partial differential equations (PDE)	Gan *et al.* (2005)
		Skin color filtering	Chen and Tiddeman (2005)
		2D human appearance model	Thome and Miguet (2005)
Contour-based tracking		Snakes	Kass *et al.* (1987)
		Active contour	Blake and Isard (1998) Isard (1998) MacCormick (2000)
		Level set technique	Sethian (1989) Yezzi and Soatto (2003) Jackson *et al.* (2004) Rathi *et al.* (2005)
Filtering-based tracking	Kalman filter-based tracking	Kalman filter (KF)	Rehg and Kanade (1994)
		Extended Kalman Filter (EKF)	Jebara *et al.* (1998)
		KF with ellipse and color	Zhao *et al.* (2004) Girondel *et al.* (2004)
		KF with elastic matching	Luo and Bhandarkar (2005)
	Particle filter-based tracking	Condenstaion algorithm	Isard (1998)
		PF with partitioned sampling	MacCormick and Isard (2000)
		PF with optimal proposal distribution (OPD)	Doucet *et al.* (2001)
		Kalman particle filter (KPF) and unscented particle filter (UPF)	Li *et al.* (2003)
		PF with Markov random field (MRF)	Wang and Cheong (2005)
		Kernel particle filter	Chang *et al.* (2005)
		PF with geometric active contours	Rathi *et al.* (2005)
		Multiple-blob tracker (BraMBLe)	Isard and MacCormick (2001)
		Boosted particle filter (BPF)	Okuma *et al.* (2004)

CHAPTER 3

FACE DETECTION AND TRACKING USING A BOOSTED ADAPTIVE PARICLE FILTER[1]

[1] Zheng, W. and S. Bhandarkar. To be submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Abstract: This paper proposes a novel algorithm for integrated face detection and face tracking based on a combination of a novel adaptive particle filtering algorithm and an AdaBoost face detection algorithm. The proposed method provides a general framework for detecting and tracking faces in video sequences. Using a novel sampling technique, an adaptive particle filter (APF) is introduced to obtain accurate estimation of the proposal distribution and the posterior distribution for accurate tracking in video sequences. The proposed scheme, termed a Boosted Adaptive Particle Filter (BAPF), combines the APF with the AdaBoost algorithm. The AdaBoost algorithm is used to detect faces in the input images, while the APF is used to track faces in video sequences. The proposed BAPF algorithm is employed for face detection, face verification, and face tracking in video sequences. The individual performances of face detection and face tracking can be mutually improved in the proposed tracking procedure. The results of experiments confirm that the proposed BAPF algorithm provides a means for robust face detection and accurate face tracking under various tracking scenarios.

Keywords: Face detection, face tracking, particle filter, boosted learning

3.1 Introduction

Face detection is important in any human face related system, such as any fully automatic face recognition system, a video-based surveillance and warning system, or face tracking and human tracking system. Recently, face detection using machine learning and statistical estimation methods has demonstrated excellent results among all existing face detection methods. Much research has been conducted in the area of face detection techniques, such as AdaBoost (Viola and Jones, 2001a; Viola and Jones, 2001b), FloatBoost (Li *et al.*, 2002), S-AdaBoost (Jiang and Loe, 2003), neural networks (Rowley *et al.*, 1996; Curran *et al.*, 2005), Support Vector Machines (SVM) (Osuna *et al.*, 1997; Shih and Liu, 2004), Hidden Markov Models (Rabiner and Jung, 1993), and the Bayes classifier (Schneiderman and Kanade, 1998; Schneiderman, 2004). Viola and Jones (2001a; 2001b) propose a robust AdaBoost face detection algorithm, which can detect faces in a rapid and robust manner with a high detection rate. Li *et al.* (2002) propose the FloatBoost algorithm, an improved version of AdaBoost, for learning a boosted classifier with minimum error rate. It uses a backtrack mechanism to improve the detection rate after each iteration of the AdaBoost procedure. However this method is computationally more inefficient than the AdaBoost algorithm. Jiang and Loe (2003) propose S-AdaBoost, a variant of AdaBoost for handling outliers in pattern detection and classification. Since this method uses different classifiers in different phases, its computational efficiency and accuracy are not satisfactory. Rowley *et al.* (1996) have done the most significant research among all face detection methods based on neural networks. They employ a multilayer neural network to learn the face and nonface patterns from the training sets consisting of face and nonface images. One drawback of their method is that only upright frontal faces can be detected. Although Rowley *et al.* further improve the method to detect rotated face images, the result is not promising because of low

detection rate. Support Vector Machines (SVMs) uses structural risk minimization to minimize the upper bound of the expected generalization error (Osuna *et al.*, 1997; Shih and Liu, 2004). The major disadvantages of SVMs are intensive computation and high memory requirements. Hidden Markov Models (HMMs) assume that face and non-face patterns can be characterized as a parametric random process. These parameters can be obtained using a well-defined estimation procedure (Rabiner and Jung, 1993). The aim of training an HMM is to estimate the appropriate parameters in an HMM model to maximize the probability of observing the training data. Schneiderman and Kanade (1998) present a naive Bayes classifier, which exploits the estimation of the joint probability of local appearance and position of a face pattern at multiple scales. However, the performance of the naive Bayes classifier is poor. To address this problem, Schneiderman (2004) proposes a restricted Bayesian network for object detection. This method searches the structure of a Bayesian network-based classifier in the large space of possible network structures.

Object tracking has been studied extensively in the context of computer vision because of various vision applications such as autonomous robots (Davison and Murray, 1998), video surveillance (Borg *et al.*, 2005), human eye tracking (Hansen and Hammoud, 2005) and human face tracking (Nummiaro *et al.*, 2003) that use tracking algorithms extensively. Issues of uncertainty and error should be considered in object tracking under complex situations (Intille *et al.*, 1997). Therefore, many techniques have been developed to solve the problem of object tracking.

Many techniques have been developed in the last decade for visual object tracking. Image-based tracking methods obtain generic features from the images and then combine them based on high-level scene information. Intille *et al.* (1997) propose a blob-tracker for human tracking in

real time. The background is subtracted to extract foreground regions. The foreground regions are then divided into blobs based on color. This approach runs fast, but it has a major disadvantage in terms of merging blobs when the objects in the scene approach each other. Contour-based tracking assumes that the object is defined by boundaries with known properties (Blake and Isard, 1998; MacCormick, 2000; Rathi *et al.*, 2005). Contour-based tracking relies on shape models (contours), dynamical models and image measurements. The tracking of object shape and location over time is well handled by the Kalman filter in the case of linear dynamic systems (Rehg and Kanade, 1994). The Extended Kalman Filter (EKF) is an extension of the Kalman filter to a nonlinear but unimodal process where non-linear behavior is approximated by local linearization (Jebara *et al.*, 1998). It is widely accepted that the particle filter is superior in tracking performance to the Kalman filter (Chang *et al.*, 2005), since the particle filter presents a robust object tracking framework without being restricted to linear systems.

Particle filters, also known as sequential Monte Carlo filters, have been widely used in visual tracking to address limitations arising from non-linearity and non-normality of the motion model (Li *et al.*, 2003; Okuma *et al.*, 2004). The basic idea of the particle filter is to approximate the posterior density using a recursive Bayesian filter based on a set of particles with assigned weights. The Condensation algorithm, a simple particle filter, proposed by Isard (1998) is designed to solve the tracking problems arising from non-linearity and non-normality of the motion model. During the sampling step, the Condensation algorithm uses a simple proposal distribution to draw a set of particles, which defines the conditional distribution on the particle state in the previous frame. This proposal distribution does not make use of the information from the current frame.

Various approaches have been developed to improve the tracking performance of a particle filter. Li *et al.* (2003) propose a Kalman particle filter (KPF) and an unscented particle filter (UPF) to improve the particle sampling in the context of visual contour tracking. This approach makes use of a Kalman filter or an unscented Kalman filter to incorporate the current observation. The Kalman filter or the unscented Kalman filter can steer the set of particles to regions of high likelihood in the search space, and thus reduce the number of particles. To address the occlusion problem, Wang and Cheong (2005) propose a particle filter with a Markov random field (MRF) based representation of the tracked object within a dynamic Bayesian framework. This method transforms an object into a composite of multiple MRF regions to improve the modeling accuracy. Using data association techniques, Chang *et al.* (2005) present a kernel particle filter to improve the sampling efficiency for multiple object tracking. This scheme invokes kernels to continuously approximate the posterior density, where the kernels for object representation and localization are allocated based on the gradient derived from the kernel density. However, this method can not handle situations in which the motion pattern of objects in one group changes drastically. Rathi *et al.* (2005) formulate geometric active contours as a parameterization technique to deal with the deformable objects. But the performance of their technique is poor when the tracked object is completely occluded over many frames. Isard and MacCormick (2001) propose a Bayesian multiple-blob tracker (BraMBLe), an early implementation of a particle filter, in which the number of tracked objects can vary during tracking. Nonetheless, this approach relies on modeling a fixed background to identify foreground objects. To address this problem, Okuma *et al.* (2004) relax the assumption of a fixed background to handle real image sequences, where the background may vary. Okuma *et al.* (2004) propose a boosted particle filter (BPF) for multiple object detection and tracking, which

interleaves the AdaBoost algorithm with a simple particle filter (the Condensation algorithm). However, this method does not present a systematic way of incorporating object models to guarantee accurate approximation of the proposal distribution, and also does not address the occlusion problem.

In this paper, we propose a new particle filtering scheme, which is termed as an adaptive particle filter (APF), to enable much more accurate estimation of the proposal distribution and of the posterior distribution. Based on the previous work of Isard (1998), Li *et al.* (2003), Vermaak *et al.* (2003) and Okuma *et al.* (2004), we also propose a novel scheme for face detection and tracking by combining the APF algorithm with the AdaBoost algorithm. We term the combination of the APF algorithm and the AdaBoost algorithm as a boosted adaptive particle filter (BAPF). The AdaBoost algorithm is used to detect faces in an input image, and the BAPF algorithm is designed for face verification and tracking in real video sequences. The BAPF algorithm can obtain good tracking results in situations in which the objects are severely occluded. Experimental results show that the proposed BAPF method provides robust face detection and accurate face tracking under various tracking scenarios.

3.2 Statistical Model

Mathematical notation: The mathematical notation used in the formulation of the statistical model and the particle filtering algorithm is described below:

$\mathbf{x}$, a state vector for an object contour;

$\mathbf{y}$, an observation vector;

$p(\mathbf{y} \mid \mathbf{x})$, observation likelihood (or termed as observation density);

T, length of the measurement line;

x_i , a finite number of sample points on a contour;

s_i , normal to the contour (or termed as measurement line);

m, index of the detected features;

m_i , number of the detected features;

z_i , edge feature;

λ , density for the the Poisson distribution of clutter features on the measurement line;

$p_T(m_i)$, the Poisson distribution of clutter features on the measurement line;

$p_{x_i}(\mathbf{z} \mid \mathbf{v} = \{x_i\})$, generic likelihood function of the observation at a sample point $x_i (i = 1,2,\cdots,n)$;

q_0, probability of undetected features for an object boundary;

q_1, probability of detected features for an object boundary;

$\mathbf{x}_t$, a state vector for an object at time *t*;

$\mathbf{x}_{1:t} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_t\}$, a state vector history up to time *t*;

$\mathbf{y}_t$, an observation vector at time *t*;

$\mathbf{y}_{1:t} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_t\}$, an observation history vector up to time *t*;

$\bar{\mathbf{x}}$, mean value of a state vector;

$\boldsymbol{\omega}_t$, Gaussian noise;

A, matrix describing the deterministic component of the dynamical model;

B, matrix describing the stochastic component;

$p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$, dynamical model (or termed as transition prior);

$p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$, posterior density;

$p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})$, effective prior;

$p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1})$, observation prior;

$f(\mathbf{x}_t)$, function of object state vector;

$E[f(\mathbf{x}_t)]$, estimate of a function $f(\mathbf{x}_t)$;

L, number of iterations of loop l in the adaptive particle filter;

N, number of particles;

$w_t^{(i)}$, particle weight;

$\mathbf{x}_t^{(i)}$, particle state vector;

$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{y}_{1:t})$, proposal distribution;

$\mathcal{N}(\hat{\mathbf{x}}_t^{(i)}, \hat{\mathbf{P}}_t^{(i)})$, Gaussian distribution;

$\mathbf{x}_{t,l}^{(i)}$, particle state vector computed within loop l;

$u_l(\mathbf{x})$, proposal distribution computed within loop l used in adaptive particle filter;

$\xi_1^{(i)}, \xi_2^{(i)}$, two specific values in domain D

m_1, M_1, two specific values of a continuous function in domain D;

m_2, M_2, two specific values of a continuous function in domain D;

Φ, a continuous function in domain D;

max_l, min_l, two specific values in domain D used to impose bounds on Φ within loop l;

K_l, a constant within loop l;

$E[f(\mathbf{x}), \hat{p}(\mathbf{x})]$, sampling error at the iteration step l with respect to $f(\mathbf{x})$;

$E_c(f(\mathbf{x}_t))$, estimate of a sampled point on the contour combining the estimation values from the APF and the AdaBoost algorithm;

γ, weight assigned to the Adaboost detection algorithm;

η, confidence measure for each detected face in the image;

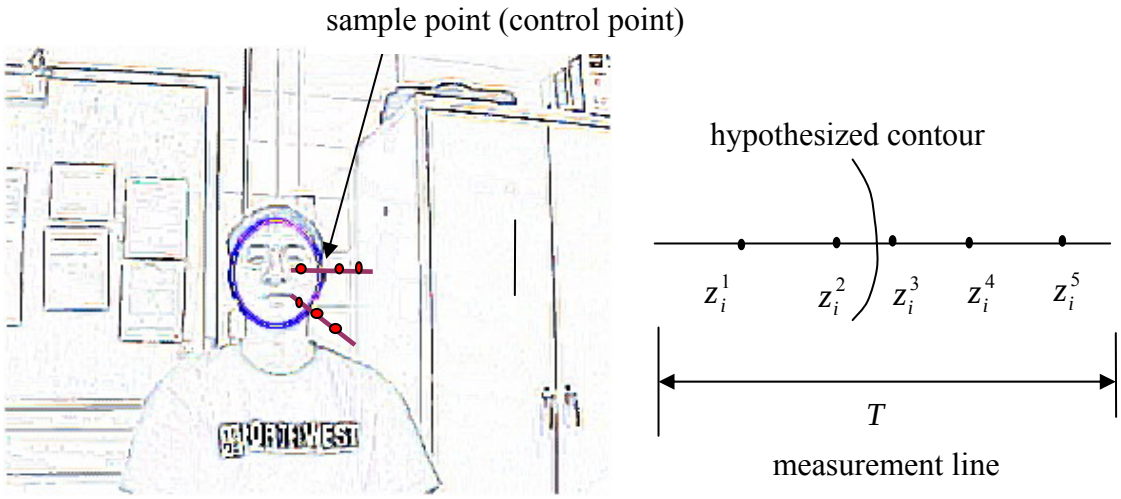
d, distance between the center of a detected face and the center of a sampled template contour;

F, number of the previous frames used for the estimation of an object in the current frame.

3.2.1 Observation Model

We denote a state vector for an object by $\mathbf{x}$, and an observation vector is denoted by $\mathbf{y}$. It is important for contour tracking to obtain accurate estimation of the observation likelihood (or termed as observation density) $p(\mathbf{y} \mid \mathbf{x})$. Blake *et al.* (1998), Isard *et al.* (1998), and MacCormick *et al.* (1998, 2000) introduce statistical models for computing the observation density $p(\mathbf{y} \mid \mathbf{x})$. These models use a set of normals to a hypothesized contour to collect specific image features. A finite number of sample points, called control points, are generated on the hypothesized contour. We follow the general direction of these models for modeling the observation process, but in particular follow the one proposed by MacCormick (2000).

Figure 3.1 shows an observed contour and image features extracted along a measurement line. We denote a finite number of sample points on a hypothesized contour by a set $\{x_i, i = 1,2,\cdots,n\}$, and term the normals to the contour as measurement lines, which are denoted by a set $\{s_i, i = 1,2,\cdots,n\}$. The length of the measurement lines is fixed at a value *T*. A Canny edge detector is applied to the measurement line $s_i (i = 1,2,\cdots,n)$ in order to obtain the positions of the edge features $\{z_i^{(m)}, m = 1,2,\cdots,m_i\}$ (*m* is the index of the detected features, m_i is the number of the detected features). Obviously, each feature is jointly generated by the boundary of an object and the random clutter presented in the image.



(a) (b)

Figure 3.1 (a) Observation process: the ellipse is a hypothesized contour in an image. (b) The image features on the measurement line.

Clutter features on the measurement line $s_i(i=1,2,\cdots,n)$ are assumed to obey the Poisson distribution with density λ :

$$p_T(m_i)=\frac{(\lambda T)^{m_i}}{m_i!}e^{-\lambda T} \tag{3-1}$$

where m_i is the number of detected clutter features. A boundary density function is assumed to obey a Gaussian distribution, thus the generic likelihood function of the observation at a sample point $x_i(i=1,2,\cdots,n)$ can be described by (MacCormick, 2000):

$$p_{x_i}(\mathbf{z}\mid \mathbf{v}=\{x_i\})=\frac{(\lambda T)^{m_i}}{m_i!}e^{-\lambda T}\left(q_0+\frac{q_1}{\lambda}\sum_{i=1}^{m_i}\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(z_i^{(m)}-x_i)^2}{2\sigma^2}\right)\right) \tag{3-2}$$

where q_0 is the probability of undetected features for an object boundary, and q_1 is the probability of detected features for an object boundary. Based on the assumption of independent

and identical distribution of all sample points, the overall likelihood function of the observation $p(\mathbf{y} \mid \mathbf{x})$ can be represented by:

$$p(\mathbf{y} \mid \mathbf{x}) = \prod_{i=1}^{n} \frac{(\lambda T)^{m_i}}{m_i!} e^{-\lambda T} \cdot \prod \left(q_0 + \frac{q_1}{\lambda} \sum_{i=1}^{m_i} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(- \frac{\left(z_i^{(m)} - x_i\right)^2}{2\sigma^2} \right) \right) \tag{3-3}$$

3.2.2 Dynamical Model

Generally, a particle filter algorithm requires a dynamical model to demonstrate how a tracking system evolves over time. An auto-regressive process (ARP) model has been widely used for the purpose of dynamic modeling (Lutkepohl, 1993; Black *et al.*, 1995; MacCormick, 2000). Blake *et al.* (1993, 1995) model object dynamics as a second order process. Isard *et al.* (1998) and Li *et al.* (2003) follow the dynamical model of Blake *et al.* (1993, 1995) for object tracking. Following the previous work of Blake *et al.* (1993, 1995), Isard *et al.* (1998) and Li *et al.* (2003), this paper employs a second-order ARP as the dynamical model for face tracking. It is widely accepted that the second-order ARP captures various motions of interest for visual tracking (MacCormick, 2000). The parameters for the dynamic model in a typical real application can be obtained by learning from the input training data. The second-order ARP presents the state $\mathbf{x}_t$ at time *t* with a linear combination of the previous two states and additive Gaussian noise. The dynamical model can be represented as a second order linear difference equation:

$$\mathbf{x}_t - \bar{\mathbf{x}} = \mathbf{A}\left(\mathbf{x}_{t-1} - \bar{\mathbf{x}}\right) + \mathbf{B}\boldsymbol{\omega}_t \tag{3-4}$$

where $\boldsymbol{\omega}_t$ is Gaussian noise that is independent of the state-vector $\mathbf{x}_t$, and $\bar{\mathbf{x}}$ denotes the mean value of the state vector. **A** and **B** are matrices describing the dynamical model with the deterministic component and the stochastic component, respectively. The state-vector $\mathbf{x}_t$

encodes the knowledge of the object contour in the current state and the previous state. It is represented by:

$$\mathbf{x}_t = \begin{pmatrix} \mathbf{X}_{t-1} \\ \mathbf{X}_t \end{pmatrix}.$$

In most real applications, we set some reasonable default values for the parameters **A, B** and $\bar{\mathbf{x}}$ of the dynamical model. It is effective and straightforward to approximate them through video sequences, in which the object conducts typical motions (Blake *et al.*, 1995; Reynard *et al.*, 1996). The dynamical model can also be represented by a temporal Markov chain (Isard *et al.*, 1998):

$$p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = C \cdot exp\left(-\frac{1}{2} \left| \mathbf{B}^{-1} \left((\mathbf{x}_t - \bar{\mathbf{x}}) - \mathbf{A}(\mathbf{x}_{t-1} - \bar{\mathbf{x}}) \right) \right|^2 \right) \tag{3-5}$$

where *C* is a constant, and $|\cdot|$ denotes the Euclidean norm.

3.3 Face Tracking using Particle Filtering

3.3.1 The Filtering Distribution

We denote a state vector for an object at time *t* by $\mathbf{x}_t$, and its history up to time *t* by $\mathbf{x}_{1:t} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_t\}$. Likewise, an observation vector at time *t* is denoted by $\mathbf{y}_t$ and its history up to time *t* is denoted by $\mathbf{y}_{1:t} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_t\}$. The standard problem of target tracking in statistical pattern recognition terminology is to estimate the state $\mathbf{x}_t$ of the objects at time t, using a set of observations $\mathbf{y}_t$ from a sequence of input images. A posterior density $p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$ demonstrates all the information about $\mathbf{x}_t$ at time *t* that is deducible from the set of observations $\mathbf{y}_t$ up to that time.

We assume that object dynamics form a temporal Markov process and observations $\mathbf{y}_t$ are independent. Therefore, the dynamics are determined by a transition prior $p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$. Given the transition prior $p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ and the observation density $p(\mathbf{y}_t \mid \mathbf{x}_t)$, the posterior density $p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$ can be computed by applying Bayes' rule (Papoulis *et al.*, 1990) for inferring the posterior state density from time-varying observations. The posterior density is estimated recursively via Bayesian filtering (Isard *et al.*, 1998; Doucet *et al.*, 2001):

$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t \mid \mathbf{x}_t, \mathbf{y}_{1:t-1}) p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})}{p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1})} = \frac{p(\mathbf{y}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})}{p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1})} \tag{3-6}$$

where

$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1} \tag{3-7}$$

The posterior density $p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$ is generally evaluated in two steps, namely prediction and updating. First, an effective prior $p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})$ shown in Eq. (3-7) is predicted from the posterior density $p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1})$ via the transition prior $p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$. Second, the posterior density $p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$ is updated based upon new observation $\mathbf{y}_t$ at time t, which is expressed in Eq. (3-6).

The observation prior $p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1})$ which is the denominator in Eq. (3-6) can be represented by:

$$p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1}) = \sum_{\mathbf{x}_t} p(\mathbf{y}_t, \mathbf{x}_t \mid \mathbf{y}_{1:t-1}) = \sum_{\mathbf{x}_t} p(\mathbf{y}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}) \tag{3-8}$$

Furthermore, the observation prior $p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1})$ can be represented by an integration operator:

$$p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1}) = \int p(\mathbf{y}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}) d\mathbf{x}_t \tag{3-9}$$

Thus Eq. (3-6) becomes:

$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t \mid \mathbf{x}_t)p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})}{\int p(\mathbf{y}_t \mid \mathbf{x}_t)p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})d\mathbf{x}_t} \tag{3-10}$$

Based on Eq. (3-7), we substitute the effective prior $p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})$ to obtain:

$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t \mid \mathbf{x}_t)\int p(\mathbf{x}_t \mid \mathbf{x}_{t-1})p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1})d\mathbf{x}_{t-1}}{\int p(\mathbf{y}_t \mid \mathbf{x}_t)\int p(\mathbf{x}_t \mid \mathbf{x}_{t-1})p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1})d\mathbf{x}_{t-1}d\mathbf{x}_t} \tag{3-11}$$

Besides the estimate of the posterior density $p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$, the estimate of a function $f(\mathbf{x}_t)$ of object state vector is also computed under many situations. We can approximate its expected value of the function $f(\mathbf{x}_t)$ by:

$$E[f(\mathbf{x}_t)] = \int f(\mathbf{x}_t)p(\mathbf{x}_t \mid \mathbf{y}_{1:t})d\mathbf{x}_t \tag{3-12}$$

Eq. (3-11) and Eq. (3-12) represent an optimal solution to the standard problem of object tracking. Obviously, this solution involves high-dimensional integrations, non-linearity and non-normality of the motion model under many tracking scenarios. High-dimensional integrations usually can not be computed easily. Thus a particle filter, also known as a sequential Monte Carlo filter, is adopted as a practical solution to the problem of object tracking.

3.3.2 The Standard Particle Filter

A standard particle filter uses Monte Carlo simulation to obtain the posterior probability $p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$ represented by Eq. (3-11). Particle filtering makes use of random sampling strategies in order to model a complex posterior probability $p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$. It uses *N* weighted discrete particles to approximate the posterior probability $p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$ by the observation of the data. Each particle consists of a state vector **x** and a weight *w*. The weighted particle set is given by $\{(\mathbf{x}_t^{(i)}, w_t^{(i)}), i = 1,2,\cdots,N\}$. Particle filtering samples the space spanned by $\mathbf{x}_t$ with *N* discrete

particles and approximates the distribution with the associated weights of the points sampled by the particles. Specifically, we assume that N particles are used for sampling to obtain the posterior probability $p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1})$, and that discrete sample points in the space are given by $\mathbf{x}_{t-1}^1, \mathbf{x}_{t-1}^2, ..., \mathbf{x}_{t-1}^N$ respectively. Thus we have:

$$p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1}) = \sum_{i=1}^{N} w_{t-1}^i \delta(\mathbf{x}_{t-1} - \mathbf{x}_{t-1}^i) \tag{3-13}$$

Since it is infeasible to draw samples directly from the posterior distribution, a proposal distribution $q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{y}_{1:t})$ is used to easily draw the samples for approximation of the posterior probabilities. Based on the proposal distribution $q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{y}_{1:t})$, a particle filter samples $\mathbf{x}_t^{(i)}$ from $\mathbf{x}_{t-1}^{(i)}$ for particle i $(i = 1,2,\cdots,N)$ and computes the weight for $\mathbf{x}_t^{(i)}$ using the following equation:

$$w_t^{(i)} = \frac{p(\mathbf{y}_t \mid \mathbf{x}_t^{(i)}) p(\mathbf{x}_t^{(i)} \mid \mathbf{x}_{t-1}^{(i)})}{q(\mathbf{x}_t^{(i)} \mid \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_{1:t})} w_{t-1}^{(i)} \tag{3-14}$$

The posterior distribution $p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$ can thus be approximated as:

$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(i)}) \tag{3-15}$$

The estimate of the function $f(\mathbf{x}_t)$ of the state vector could be computed as:

$$E[f(\mathbf{x}_t)] \approx \sum_{i=1}^{N} w_t^{(i)} f(\mathbf{x}_t^{(i)}) \tag{3-16}$$

The standard particle filter can be described as consisting of four steps: initialization, sampling, estimation, and selection (Li *et al.*, 2003). In the sampling step, a set of particles is drawn from the proposal distribution, and each particle is weighted based on the ratio of its true probability to its approximated probability using the proposal distribution. In the estimation step,

the standard particle filter approximates the posterior density using the output of the sampling step, namely the particles' states and weights. The particles are selected according to the estimated posterior density to obtain a uniform weight distribution in the selection step.

The standard particle filtering algorithm is illustrated in Figure 3.2.

1. Initialization
   Initialize a set of particles from the prior $p(\mathbf{x}_0)$ to obtain $\left\{\left(\mathbf{x}_0^{(i)}, w_0^{(i)}\right), i = 1,2,\cdots,N\right\}$. Let $t=0$.
2. Sampling step
   a) For $i = 1, 2, \ldots, N$
      Sample $\mathbf{x}_t^{(i)}$ from the proposal distribution $q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{y}_{1:t})$.
   b) Compute the weights of particles
   $$w_t^{(i)} = \frac{p\left(\mathbf{y}_t \mid \mathbf{x}_t^{(i)}\right) p\left(\mathbf{x}_t^{(i)} \mid \mathbf{x}_{t-1}^{(i)}\right)}{q\left(\mathbf{x}_t^{(i)} \mid \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_{1:t}\right)} w_{t-1}^{(i)}, \qquad i = 1, 2, \ldots, N$$
   c) Normalize
   $$w_t^{(i)} = \frac{w_t^{(i)}}{\sum_{i=1}^{N} w_t^{(i)}}, \qquad i = 1, 2, \ldots, N$$
3. Estimation step
   Obtain a set of particles $\left\{\left(\mathbf{x}_t^{(i)}, w_t^{(i)}\right), i = 1,2,\cdots,N\right\}$. The posterior distribution $p(\mathbf{x}_t \mid \mathbf{y}_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta\left(\mathbf{x}_t - \mathbf{x}_t^{(i)}\right)$ can be approximated using the output set of particles, where $\delta(\cdot)$ is the Dirac function. The estimate of $f(\mathbf{x}_t)$ can be computed by:
   $$E[f(\mathbf{x}_t)] \approx \sum_{i=1}^{N} w_t^{(i)} f\left(\mathbf{x}_t^{(i)}\right).$$
4. Selection step
   Resample particles $\mathbf{x}_t^{(i)}$ with probability $w_t^{(i)}$ to obtain N i.i.d random particles $\mathbf{x}_t^{(i)}$, approximately distributed with respect to $p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$.
   Assign $w_t^{(i)} = \frac{1}{N}$, $i = 1, 2, \ldots, N$.
5. Set $t=t+1$, go to step 2.

Figure 3.2 The algorithm of a standard particle filter

3.3.3 The Adaptive Particle Filter

3.3.3.1 The Adaptive Particle Filter Algorithm

Recently, one of the active research areas in particle filtering is to generate a good proposal distribution $q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{y}_{1:t})$ and thus obtain a more accurate estimate of the posterior distribution $p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$. The aim is to obtain a close approximation to the posterior probability distribution. The Condensation (Isard *et al.*, 1998) algorithm makes no use of knowledge obtained from the current image frame, which leads to a rough estimate of the posterior distribution. Doucet *et al.* (2001) provide an optimal proposal distribution (OPD) for state estimation of jump Markov linear systems, and recursively compute optimal state estimates based on the selection of the minimum variance of weights $w_t^{(i)}$ ($i = 1, 2, \ldots, N$). To overcome the problem of inefficient computation of the OPD, Li *et al.* (2003) propose a Kalman particle filter (KPF) and an unscented particle filter (UPF) to drive a set of particles to the regions in the search space with high likelihood. Li *et al.* (2003) employ a local linearization of the OPD to estimate the proposal distribution, which is assumed to be a Gaussian distribution. Therefore, the proposal distribution can be represented as:

$$u_l(\mathbf{x}) = q(\mathbf{x}_t^{(i)} \mid \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_{1:k}) = \mathcal{N}(\hat{\mathbf{x}}_t^{(i)}, \hat{\mathbf{P}}_t^{(i)}) \qquad i = 1, 2, \ldots, N. \tag{3-17}$$

where mean $\hat{\mathbf{x}}_t^{(i)}$ and covariance $\hat{\mathbf{P}}_t^{(i)}$ characterize the Gaussian distribution $\mathcal{N}(\hat{\mathbf{x}}_t^{(i)}, \hat{\mathbf{P}}_t^{(i)})$.

In this paper, we propose a new particle filtering scheme, termed as an Adaptive Particle Filter (APF), to enable much more accurate estimation of the proposal distribution and the posterior distribution. Our method extends the Condensation algorithm and the Kalman particle filter to obtain an accurate approximation of the proposal distribution and the posterior distribution.

In the sampling step of the APF algorithm as shown in Figure 3.3, a new sampling strategy is used to improve the accuracy of the approximation, which is different from the sampling used in other particle filters. The sampling step is the most important step in a particle filtering algorithm. For each discrete particle $\mathbf{x}_{t,l-1}^{(i)}$, the adaptive particle filter generates a new particle $\mathbf{x}_{t,l}^{(i)}$ based on a proposal distribution $u_l(\mathbf{x})$. We use the loop controlled by the parameter *l* in the APF algorithm to implement the new sampling technique. L is the fixed number of iterations of loop *l*. *L* can be adjusted in different real applications. When $L = 1$, the APF is equivalent to the pure standard particle filter. When $L > 1$, the APF performs more sampling iterations than the standard particle filter. We will prove in Section 3.3.3.2 that the additional iterations obtain a lower estimation error of the proposal distribution and posterior distribution.

In order to enable more accurate estimation of the proposal distribution, we iterate the sampling procedure with a constraint, which is called the Adaptive Learning Constraint (ALC). The ALC is described using the following equation which is detailed in a later section.

$$K_l \cdot max_l \le \alpha \cdot K_{l-1} \cdot min_{l-1} \tag{3-18}$$

where

$$max_l = \max_{1 \le i \le N} \left\{ M_1 \left| \xi_1^{(i)} - \mathbf{x}_{t,l}^{(i)} \right|, M_2 \left| \xi_2^{(i)} - \mathbf{x}_{t,l}^{(i)} \right| \right\}$$

$$min_{l-1} = \min_{1 \le i \le N} \left\{ m_1 \left| \xi_1^{(i)} - \mathbf{x}_{t,l-1}^{(i)} \right|, m_2 \left| \xi_2^{(i)} - \mathbf{x}_{t,l-1}^{(i)} \right| \right\}$$

K_l, K_{l-1} are constants, $0 < \alpha < 1$.

If the proposed constraint is satisfied, the iteration for generating new particles in the same image will continue. The iteration in the sampling step will stop when the proposed constraint is not satisfied or the predefined loop threshold is reached. Theoretically and practically, the particles with state vector and weights obtained in the latest iteration will present a better

approximation of the proposal distribution and the posterior distribution. The theoretical analysis and experimental results will be presented in later sections to confirm the superior performance of the APF algorithm.

The other steps of APF are similar to those of other particle filters such as KPF and UPF. The initialization step takes advantage of the information from the results of the AdaBoost face detection algorithm. The adaptive particle filter algorithm is described in Figure 3.3.

1. Initialization
   Initialize a set of particles from the prior $p(\mathbf{x}_0)$ to get $\{(\mathbf{x}_0^{(i)}, w_0^{(i)}), i = 1,2,\cdots,N\}$. Let *t=0*.
2. Sampling step
   1) For *l = 1, 2, … , L*
      a) For *i = 1, 2, … , N*
         Sample $\mathbf{x}_{t,l}^{(i)}$ from $\mathbf{x}_{t,l-1}^{(i)}$ based on the proposal distribution $u_l(\mathbf{x})$, where
         $u_l(\mathbf{x}) = q(\mathbf{x}_t^{(i)} \mid \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_{1:k}) = \mathcal{N}(\hat{\mathbf{x}}_t^{(i)}, \hat{\mathbf{P}}_t^{(i)})$. Construct $p_{l-1}(\mathbf{x}) = \sum_{i=1}^{N} w_{t,l-1}^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_{t,l-1}^{(i)})$,
         where $\delta(\cdot)$ is the Dirac function.
      b) If the Adaptive Learning Constraint is satisfied, where $K_l \cdot max_l \le \alpha \cdot K_{l-1} \cdot min_{l-1}$:
         i) Compute the weights of particles
            $w_{t,l}^{(i)} = p(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}) p(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}) w_{t,l-1}^{(i)}$, *i = 1, 2, … , N*
            $w_{t,0}^{(i)} = w_{t-1}^{(i)}$ when $l = 1$.
         ii) Normalize
            $w_{t,l}^{(i)} = \frac{w_{t,l}^{(i)}}{\sum_{i=1}^{N} w_{t,l}^{(i)}}$, *i = 1, 2, …, N*
         iii) Continue the loop *l*
      b) If the Adaptive Learning Constraint is not met, where $K_l \cdot max_l > \alpha \cdot K_{l-1} \cdot min_{l-1}$:
         i) Let $w_t^{(i)} = w_{t,l-1}^{(i)}$, $\mathbf{x}_t^{(i)} = \mathbf{x}_{t,l-1}^{(i)}$,
         ii) Break the loop *l*
   2) Let $w_t^{(i)} = w_{t,L}^{(i)}$, $\mathbf{x}_t^{(i)} = \mathbf{x}_{t,L}^{(i)}$, *i = 1, 2, … , N.*
   3) $w_t^{(i)} = \frac{w_t^{(i)}}{q(\mathbf{x}_t^{(i)} \mid \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_{1:k})}$, *i = 1, 2, … , N.*
3. Estimation step
   Obtain a set of particles $\{(\mathbf{x}_t^{(i)}, w_t^{(i)}), i = 1,2,\cdots,N\}$. The posterior distribution $p(\mathbf{x}_t \mid \mathbf{y}_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(i)})$ can be approximated using the output set of particles. The estimate value of $f(\mathbf{x}_t)$ can be computed as:
   $E[f(\mathbf{x}_t)] \approx \sum_{i=1}^{N} w_t^{(i)} f(\mathbf{x}_t^{(i)})$.
4. Selection step
   Resample particles $\mathbf{x}_t^{(i)}$ with probability $w_t^{(i)}$ to obtain *N* i.i.d random particles $\mathbf{x}_t^{(i)}$, approximately distributed with respect to $p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$.
   Assign $w_t^{(i)} = \frac{1}{N}$, *i = 1, 2, … , N.*
5. Set *t=t+1*, go to step 2.

Figure 3.3 The algorithm of adaptive particle filter

3.3.3.2 Modeling The Adaptive Learning Constraint

A critical step in the adaptive particle filter (APF) is obtaining a good approximation to to the sampling proposal distribution $u_l(\mathbf{x})$, which is shown in the sampling step in Figure 3.3. The purpose of choosing the proposal distribution $u_l(\mathbf{x})$ recursively in a given state is to reduce the estimation error, which is a result of approximating the posterior distribution $p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$ with a finite number of particles. The design of a single iteration of the estimation of the proposal distribution $u_l(\mathbf{x})$ will be presented in the following derivation. The iteration is described by the loop *l*: *l = 1, 2, … , L* in Figure 3.3, where *L* is a given default value. The additional iterations of loop *l* in the first loop of step 2 could be used to reduce the estimation error adaptively.

We make following analysis to prove this point. First, we prove that the iterations result in the convergence of the estimate of the proposal distribution. The convergence shows that the estimation error of the proposal distribution at loop step *l = k+1* is less than that of the proposal distribution at loop step *l = k*, where $k \in (1,2,\cdots,L-1)$. This results in better approximation of the proposal distribution and the posterior distribution through the iterations of loop *l*. Thus, we can obtain a lower estimation error of the proposal distribution and the posterior distribution. Second, we present the Adaptive Learning Constraint in the derivation, which clarifies the APF described in Figure 3.3.

We define the error of a sampling function $\hat{p}(\mathbf{x})$ with respect to $f(\mathbf{x})$ as:

$$E[f(\mathbf{x}), \hat{p}(\mathbf{x})] = \left| \int f(\mathbf{x})(p(\mathbf{x}) - \hat{p}(\mathbf{x}))d\mathbf{x} \right| \tag{3-19}$$

where

$$p(\mathbf{x}) = p(\mathbf{x}_t \mid \mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t \mid \mathbf{x}_t)\int p(\mathbf{x}_t \mid \mathbf{x}_{t-1})p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1})d\mathbf{x}_{t-1}}{\int p(\mathbf{y}_t \mid \mathbf{x}_t)\int p(\mathbf{x}_t \mid \mathbf{x}_{t-1})p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1})d\mathbf{x}_{t-1}d\mathbf{x}_t},$$

$\hat{p}(\mathbf{x}) = \hat{p}(\mathbf{x}_t \mid \mathbf{y}_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(i)})$, and

$|\cdot|$ denotes the Euclidean norm.

The propagation of errors between the iterations in the adaptive particle filter can be analyzed as follows. Specifically, we consider a single iteration step *l*: $l \in \{1,2,\cdots,L\}$. From the APF algorithm shown in Figure 3.3, the estimate of the proposal distribution in the iteration step *l* is given by:

$$p_l(\mathbf{x}) = \sum_{i=1}^{N} w_{t,l}^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_{t,l}^{(i)}) \tag{3-20}$$

Thus, the sampling error $E[f(\mathbf{x}), \hat{p}(\mathbf{x})]$ at the iteration step *l* with respect to $f(\mathbf{x})$ is computed as:

$$\begin{aligned} & E[f(\mathbf{x}), \hat{p}(\mathbf{x})] \\ & = \left| \int f(\mathbf{x})(p(\mathbf{x}) - p_l(\mathbf{x}))d\mathbf{x} \right| \\ & = \left| f(\mathbf{x}_t) \left(\frac{p(\mathbf{y}_t \mid \mathbf{x}_t) \int p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1}}{\int p(\mathbf{y}_t \mid \mathbf{x}_t) \int p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1} d\mathbf{x}_t} - \sum_{i=1}^{N} w_{t,l}^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_{t,l}^{(i)}) \right) d\mathbf{x}_t \right| \end{aligned} \tag{3-21}$$

Based upon Eq. (3-13), we have $p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1}) = \sum_{i=1}^{N} w_{t-1}^{i} \delta(\mathbf{x}_{t-1} - \mathbf{x}_{t-1}^{i})$. Substituting $p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1})$ with $\sum_{i=1}^{N} w_{t-1}^{i} \delta(\mathbf{x}_{t-1} - \mathbf{x}_{t-1}^{i})$ in Eq. (3-21) and performing the Dirac function computation, we obtain the estimation error as:

$$\begin{aligned} & E[f(\mathbf{x}), p_l(\mathbf{x})] \\ & = \left| f(\mathbf{x}_t) \left(\frac{p(\mathbf{y}_t \mid \mathbf{x}_t) \sum_{i=1}^{N} w_{t-1}^{(i)} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)})}{\int p(\mathbf{y}_t \mid \mathbf{x}_t) \sum_{i=1}^{N} w_{t-1}^{(i)} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)}) d\mathbf{x}_t} - \sum_{i=1}^{N} w_{t,l}^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_{t,l}^{(i)}) \right) d\mathbf{x}_t \right| \end{aligned}$$

$$= \left| f(\mathbf{x}_t) \left(\frac{\sum_{i=1}^{N} p(\mathbf{y}_t \mid \mathbf{x}_t) w_{t-1}^{(i)} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)})}{\sum_{i=1}^{N} \int p(\mathbf{y}_t \mid \mathbf{x}_t) w_{t-1}^{(i)} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)}) d\mathbf{x}_t} - \sum_{i=1}^{N} w_{t,l}^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_{t,l}^{(i)}) \right) d\mathbf{x}_t \right| \tag{3-22}$$

From the APF algorithm shown in Fig. 3-2, we know

$$w_{t,l}^{(i)} = p(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}) p(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}) w_{t,l-1}^{(i)} \tag{3-23}$$

$$w_{t,l}^{(i)} = \frac{w_{t,l}^{(i)}}{\sum_{i=1}^{N} w_{t,l}^{(i)}} = \frac{p(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}) p(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}) w_{t,l-1}^{(i)}}{\sum_{i=1}^{N} p(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}) p(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}) w_{t,l-1}^{(i)}} \tag{3-24}$$

After combining Eq. (3-22), Eq. (3-23), and Eq. (3-24), we obtain:

$E[f(\mathbf{x}), p_l(\mathbf{x})]$

$$= \left| f(\mathbf{x}_t) \left(\frac{\sum_{i=1}^{N} p(\mathbf{y}_t \mid \mathbf{x}_t) w_{t-1}^{(i)} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)})}{\sum_{i=1}^{N} \int p(\mathbf{y}_t \mid \mathbf{x}_t) w_{t-1}^{(i)} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)}) d\mathbf{x}_t} - \sum_{i=1}^{N} \frac{p(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}) p(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}) w_{t,l-1}^{(i)}}{\sum_{i=1}^{N} p(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}) p(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}) w_{t,l-1}^{(i)}} \delta(\mathbf{x}_t - \mathbf{x}_{t,l}^{(i)}) \right) d\mathbf{x}_t \right|$$

$$= \left| \int \left(\frac{f(\mathbf{x}_t) \sum_{i=1}^{N} p(\mathbf{y}_t \mid \mathbf{x}_t) w_{t-1}^{(i)} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)})}{\sum_{i=1}^{N} \int p(\mathbf{y}_t \mid \mathbf{x}_t) w_{t-1}^{(i)} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)}) d\mathbf{x}_t} \right) d\mathbf{x}_t - \frac{\sum_{i=1}^{N} f(\mathbf{x}_{t,l}^{(i)}) p(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}) p(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}) w_{t,l-1}^{(i)}}{\sum_{i=1}^{N} p(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}) p(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}) w_{t,l-1}^{(i)}} \right|$$

$$= \left| \frac{\sum_{i=1}^{N} \int f(\mathbf{x}_t) p(\mathbf{y}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)}) w_{t-1}^{(i)} d\mathbf{x}_t}{\sum_{i=1}^{N} \int p(\mathbf{y}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)}) w_{t-1}^{(i)} d\mathbf{x}_t} - \frac{\sum_{i=1}^{N} f(\mathbf{x}_{t,l}^{(i)}) p(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}) p(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}) w_{t,l-1}^{(i)}}{\sum_{i=1}^{N} p(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}) p(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}) w_{t,l-1}^{(i)}} \right| \tag{3-25}$$

Using the Lagrange theorem, we could obtain specific values $\xi_1^{(i)}$ and $\xi_2^{(i)}$ in domain D: $\xi_1^{(i)} \in D$, $\xi_2^{(i)} \in D$, $(i = 1,2,3,\cdots,N)$ such that

$$\int f(\mathbf{x}_t) p(\mathbf{y}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)}) w_{t-1}^{(i)} d\mathbf{x}_t = f(\xi_1^{(i)}) p(\mathbf{y}_t \mid \xi_1^{(i)}) p(\xi_1^{(i)} \mid \mathbf{x}_{t-1}^{(i)}) w_{t-1}^{(i)} \tag{3-26}$$

$$\int p(\mathbf{y}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)}) w_{t-1}^{(i)} d\mathbf{x}_t = p(\mathbf{y}_t \mid \xi_2^{(i)}) p(\xi_2^{(i)} \mid \mathbf{x}_{t-1}^{(i)}) w_{t-1}^{(i)} \tag{3-27}$$

Therefore we obtain:

$$E\left[f(\mathbf{x}), p_l(\mathbf{x})\right]$$

$$= \left| \frac{\sum_{i=1}^{N} f\left(\xi_1^{(i)}\right) p(\mathbf{y}_t \mid \xi_1^{(i)}) p\left(\xi_1^{(i)} \mid \mathbf{x}_{t-1}^{(i)}\right) w_{t-1}^{(i)}}{\sum_{i=1}^{N} p\left(\mathbf{y}_t \mid \xi_2^{(i)}\right) p\left(\xi_2^{(i)} \mid \mathbf{x}_{t-1}^{(i)}\right) w_{t-1}^{(i)}} - \frac{\sum_{i=1}^{N} f\left(\mathbf{x}_{t,l}^{(i)}\right) p\left(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}\right) p\left(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}\right) w_{t,l-1}^{(i)}}{\sum_{i=1}^{N} p\left(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}\right) p\left(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}\right) w_{t,l-1}^{(i)}} \right|$$

$$= \sum_{i=1}^{N} \left| \frac{f\left(\xi_1^{(i)}\right) p(\mathbf{y}_t \mid \xi_1^{(i)}) p\left(\xi_1^{(i)} \mid \mathbf{x}_{t-1}^{(i)}\right) w_{t-1}^{(i)}}{\sum_{i=1}^{N} p\left(\mathbf{y}_t \mid \xi_2^{(i)}\right) p\left(\xi_2^{(i)} \mid \mathbf{x}_{t-1}^{(i)}\right) w_{t-1}^{(i)}} - \frac{f\left(\mathbf{x}_{t,l}^{(i)}\right) p\left(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}\right) p\left(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}\right) w_{t,l-1}^{(i)}}{\sum_{i=1}^{N} p\left(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}\right) p\left(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}\right) w_{t,l-1}^{(i)}} \right| \tag{3-28}$$

Suppose that $f(\mathbf{x})$, $p(\mathbf{y}_t \mid \mathbf{x}_t)$, $p\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)}\right)$ are continuous functions on domain *D*, hence we have the following equation:

$$\exists m_1, M_1 \in R,$$

$$\begin{aligned} & m_1 \left| \xi_1^{(i)} - \mathbf{x}_{t,l}^{(i)} \right| \\ & \le \left| f\left(\xi_1^{(i)}\right) p(\mathbf{y}_t \mid \xi_1^{(i)}) p\left(\xi_1^{(i)} \mid \mathbf{x}_{t-1}^{(i)}\right) w_{t-1}^{(i)} - f\left(\mathbf{x}_{t,l}^{(i)}\right) p\left(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}\right) p\left(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}\right) w_{t,l-1}^{(i)} \right| \\ & \le M_1 \left| \xi_1^{(i)} - \mathbf{x}_{t,l}^{(i)} \right| \end{aligned} \tag{3-29}$$

Likewise, we have:

$$\exists m_2, M_2 \in R$$

$$\begin{aligned} & m_2 \left| \xi_2^{(i)} - \mathbf{x}_{t,l}^{(i)} \right| \\ & \le \left| \sum_{i=1}^{N} p(\mathbf{y}_t \mid \xi_2^{(i)}) p\left(\xi_2^{(i)} \mid \mathbf{x}_{t-1}^{(i)}\right) w_{t-1}^{(i)} - \sum_{i=1}^{N} p\left(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}\right) p\left(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}\right) w_{t,l-1}^{(i)} \right| \\ & \le M_2 \left| \xi_2^{(i)} - \mathbf{x}_{t,l}^{(i)} \right| \end{aligned} \tag{3-30}$$

Let

$$F_1^{(i)} = f\left(\xi_1^{(i)}\right) p(\mathbf{y}_t \mid \xi_1^{(i)}) p\left(\xi_1^{(i)} \mid \mathbf{x}_{t-1}^{(i)}\right) w_{t-1}^{(i)} \tag{3-31}$$

$$F_2 = \sum_{i=1}^{N} p(\mathbf{y}_t \mid \xi_2^{(i)}) p\left(\xi_2^{(i)} \mid \mathbf{x}_{t-1}^{(i)}\right) w_{t-1}^{(i)} \tag{3-32}$$

$$\Delta F_1^{(i)} = f\left(\mathbf{x}_{t,l}^{(i)}\right) p\left(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}\right) p\left(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}\right) w_{t,l-1}^{(i)} - F_1^{(i)} \tag{3-33}$$

$$\Delta F_2 = \sum_{i=1}^{N} p\left(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}\right) p\left(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}\right) w_{t,l-1}^{(i)} - F_2 \tag{3-34}$$

Thus we obtain:

$$E\left[f(\mathbf{x}), p_l(\mathbf{x})\right]$$

$$= \sum_{i=1}^{N} \left| \frac{F_1^{(i)}}{F_2} - \frac{F_1^{(i)} + \Delta F_1^{(i)}}{F_2 + \Delta F_2} \right|$$

$$= \sum_{i=1}^{N} \left| \frac{F_1^{(i)}\left(F_2 + \Delta F_2\right) - F_2\left(F_1^{(i)} + \Delta F_1^{(i)}\right)}{F_2\left(F_2 + \Delta F_2\right)} \right|$$

$$= \sum_{i=1}^{N} \left| \frac{F_1^{(i)} \Delta F_2 - F_2 \Delta F_1^{(i)}}{F_2\left(F_2 + \Delta F_2\right)} \right|$$

$$\approx \sum_{i=1}^{N} \left| \frac{F_1^{(i)} \Delta F_2 - F_2 \Delta F_1^{(i)}}{F_2^{\,2}} \right| \tag{3-35}$$

Let

$$\Phi = \sum_{i=1}^{N} \left| F_1^{(i)} \Delta F_2 - F_2 \Delta F_1^{(i)} \right| \tag{3-36}$$

Since $f(\mathbf{x})$, $p(\mathbf{y}_t \mid \mathbf{x}_t)$, $p\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)}\right)$ are continuous functions defined on domain *D*, $F_1^{(i)}$, F_2, $\Delta F_1^{(i)}$, ΔF_2 are also continuous functions on domain *D*. Furthermore, Φ is also a continuous function on domain *D*. Based on the properties of continuous functions, Eq. (3-29) and Eq. (3-30), Φ is bounded by two specific values, max_l and min_l.

Let

$$max_l = \max_{1 \le i \le N} \left\{ M_1 \left| \xi_1^{(i)} - \mathbf{x}_{t,l}^{(i)} \right|, M_2 \left| \xi_2^{(i)} - \mathbf{x}_{t,l}^{(i)} \right| \right\} \tag{3-37}$$

$$min_l = \min_{1 \le i \le N} \left\{ m_1 \left| \xi_1^{(i)} - \mathbf{x}_{t,l}^{(i)} \right|, m_2 \left| \xi_2^{(i)} - \mathbf{x}_{t,l}^{(i)} \right| \right\} \tag{3-38}$$

We obtain:

$$K_l \cdot min_l \le E\left(f(X), p_l(X) \right) \le K_l \cdot max_l, \qquad \text{for loop step } l \tag{3-39}$$

where K_l is a constant.

Likewise, we can get the following equation at loop step *l*-1:

$$K_{l-1} \cdot min_{l-1} \le E\left(f(\mathbf{x}), p_{l-1}(\mathbf{x}) \right) \le K_{l-1} \cdot max_{l-1}, \qquad \text{for loop step } l\text{-1} \tag{3-40}$$

Let

$$\frac{K_l \cdot max_l}{K_{l-1} \cdot min_{l-1}} \le \alpha < 1, \text{ where } 0 < \alpha < 1, \tag{3-41}$$

thus we obtain:

$$E\left(f(X), p_l(X) \right) \le \alpha \cdot E\left(f(X), p_{l-1}(X) \right), \qquad 0 < \alpha < 1 \tag{3-42}$$

If Eq. (3-41) is satisfied, then Eq. (3-42) ensures that the estimation error for the proposal distribution and posterior distribution converges during the iterations. Eq. (3-41) is only a necessary condition for the convergence of the estimate, which can be learned from the computation during the iterations. So we name it as the Adaptive Learning Constraint (ALC). Eq. (3-41) can be represented by:

$$K_l \cdot max_l \le \alpha \cdot K_{l-1} \cdot min_{l-1} \tag{3-43}$$

Either Eq. (3-41) or Eq. (3-43) is termed as the ALC.

The ALC can be guaranteed by searching the $\max_l$ and $\min_{l-1}$ from the *N* particles in each iteration as follows:

1. Find $\xi_1^{(i)}$, $\xi_2^{(i)}$, M_1 , M_2, $(i = 1,2,\cdots,N)$ from the N particles in loop l. Determine $\xi_1^{(i)}$, $\xi_2^{(i)}$, M_1 , M_2, $(i = 1,2,\cdots,N)$ from the N particles in loop l-1.
2. Search for $\max_l$ and $\min_{l-1}$ using Eq. (3-37) and Eq. (3-38).
3. Determine whether or not the ALC is satisfied according to Eq. (3-43).

Thus, we prove that the iterations of loop l result in the convergence of the estimate of the proposal distribution. The convergence demonstrates that the estimation error of the proposal distribution at loop step $l = k+1$ is less than that of the proposal distribution at loop step $l = k$, where $k \in (1,2,\cdots,L-1)$. This results in better approximation of the proposal distribution and the posterior distribution through the iterations of loop l. As a result, we can obtain a lower estimation error of the proposal distribution and the posterior distribution. Therefore, we confirm that the APF algorithm with ALC can result in a more accurate estimate of the proposal distribution and posterior distribution. Generally, as more frames are processed during tracking, general particle filters will result in a monotonically increasing tracking error. However, the proposed APF algorithm is designed to improve the estimate of the proposal distribution and the posterior distribution as a tracking system evolves over time.

3.4 The Boosted Adaptive Particle Filter

The boosted adaptive particle filter (BAPF) for face detection and tracking employs two object models: the contour-based model used in the adaptive particle filter (APF) and the region-based model used in face detection. The object models used in the context of tracking lie in three general categories (Luo, 2005): the contour-based models (Li *et al.*, 2003; Koller *et al.*, 1994; Terzopoulos *et al.*, 1993), the region-based models (Isard *et al.*,2001; McKenna *et al.*, 2000;

Nummiaro *et al.*, 2003), and the feature point-based models (Rucklidge, 1995; Malik *et al.*, 2002; Lepetit *et al.*, 2004).

Since the BAPF algorithm uses two models in face detection and tracking, it has advantages over the general particle filter. The incorporation of the AdaBoost algorithm within the APF algorithm substantially improves the robustness of the BAPF algorithm. The AdaBoost algorithm presents a mechanism for maintaining the combined representation, which makes the BAPF algorithm more powerful than the naïve K-means clustering method of Vermaak *et al.* (2003). The BAPF algorithm also performs better than the mixture representation proposed by Okuma *et al.* (2004) since our approach employs a more effective particle filtering algorithm, i.e., the APF algorithm. Specifically, the BAPF algorithm allows us to effectively detect faces leaving and entering the regions of interest, and the BAPF algorithm provides robust face detection and accurate face tracking under various tracking scenarios.

3.4.1 Face Detection through AdaBoost

Among the various face detection methods, the boosted learning-based face detection methods have demonstrated excellent results. Based on the previous work of Tieu *et al.* (2000) and Schneiderman (2000), Viola and Jones (2001a; 2001b) have proposed a robust face detection algorithm, which can detect faces in a rapid and robust manner with a high detection rate. The face detection technique in AdaBoost is comprised of three aspects: the integral image, a strong classifier comprising of weak classifiers based on the AdaBoost learning algorithm, and an architecture comprising of a cascade of a number of strong classifiers.

The system of Viola and Jones (2001a; 2001b) employs an integral image comprising of Haar-like features (Lienhart and Maydt, 2002) for effective feature extraction from a large

feature set. In the boosting procedure, AdaBoost first learns effective features from a large feature set. Second, it constructs a set of weak classifiers, each of which is composed of a feature, a threshold and a parity. Third, it generates a strong classifier based on the above weak classifiers. Each iteration in the AdaBoost algorithm generates a weak classifier. After all iterations are completed, the result is a set of weak classifiers. These weak classifiers are combined into a strong classifier using a weighted linear combination. The system of Viola and Jones (2001a; 2001b) uses a cascade of strong classifiers to improve detection rate with efficient computation. The idea is to construct smaller and efficient classifiers based on the sub-windows within the image. The simpler and faster classifiers will reject the negative sub-windows. A large number of negatives are rejected by the initial classifier with minimal processing. Additional negatives are eliminated by subsequent layers while requiring additional computation. The number of sub-windows to be processed reduces rapidly after several stages of processing.

We employ the system of Viola and Jones (2001a; 2001b) for face detection. A 25 layer cascade of boosted classifiers is trained to detect multiview faces. A set of face and nonface (termed as background) sample images are used for training. Each sample image is cropped and scaled to a resolution of 20×20 pixels. A set of 6230 multiview face images are collected from video sequences with different reflections, illuminations and backgrounds to make face detection more robust in different scenarios. Another set of 6598 nonface examples with the size of 320×240 pixels are collected from video sequences containing no faces. The details of AdaBoost training and AdaBoost face detection results are presented in Section 3.5.

3.4.2 Integrating Adaptive Particle Filter with AdaBoost

The proposed face detection and tracking model consists of two submodels: an AdaBoost face detection model and an adaptive particle filter face tracking model. The AdaBoost face detection model performs multiview face detection based on the trained AdaBoost algorithm. The APF model conducts visual contour tracking using the particle filtering algorithm described in Section 3.3.3. Figure 3.4 shows the structure of the proposed face detection and tracking model.

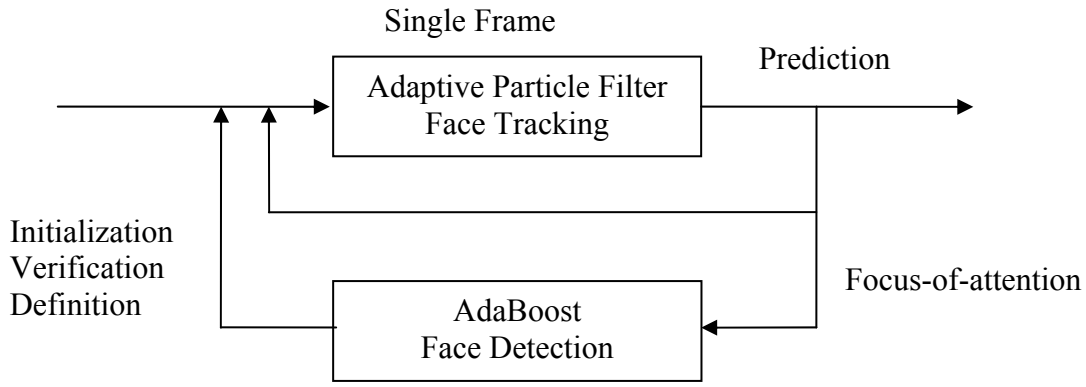


Figure 3.4 Integrating the APF with AdaBoost within a single feedback control system

The process for face detection and tracking contains two phases: an initialization phase and a tracking phase. In the initialization phase of the APF, the AdaBoost face detection model can provide the initial parameters for the APF face tracking model based on the observations of the image sequences during a certain time interval. During the tracking phase, the AdaBoost face detection model and the APF face tracking model improve the tracking performance via mutual interaction. The AdaBoost detection model helps the APF model to find and define new objects, and to verify the current states of the objects being tracked. On the other hand, the APF model provides focus-of-attention regions within the image to speed up the AdaBoost face detection.

After applying AdaBoost face detection to one image, we obtain a confidence measure η for each detected face in the image from the detection procedure. From the APF algorithm, the

estimate of $f(\mathbf{x}_t)$ in the adaptive particle filter at each sample point along the contour is computed as:

$$E[f(\mathbf{x}_t)] \approx \sum_{i=1}^{N} w_t^{(i)} f(\mathbf{x}_t^{(i)}) \tag{3-44}$$

We combine the results of the AdaBoost algorithm and the APF algorithm to obtain new position for a sampled point, which is described by:

$$E_c(f(\mathbf{x}_t)) = (1-\gamma) \cdot E(f(\mathbf{x}_t)) + \gamma \cdot \eta \cdot d \tag{3-45}$$

where E_c represents the estimate of a sampled point in the contour combining the estimation values from the APF and the AdaBoost algorithm, the parameter γ is the weight assigned to the Adaboost detection, the parameter η is a confidence measure for each detected face in the image, and *d* is the distance between the center of a detected face and the center of a sampled template contour. The value of $E_c(f(\mathbf{x}_t))$ is fed back to the APF for further processing.

The parameter γ can be adjusted without affecting the convergence of the adaptive particle filter. When $\gamma = 0$, our approach is equivalent to the pure adaptive particle filter. By increasing γ, we emphasize the AdaBoost face detection. When $\gamma = 1$, our approach is equivalent to the pure AdaBoost algorithm. In reality, we could adjust the value of the parameter γ based on different scene conditions determined by clutter, illumination and occlusions.

3.5 Experimental Results

3.5.1 AdaBoost Face Detection

The system of Viola and Jones (2001a; 2001b) is used to detect faces in input images. In our experiment, we train a 25-layer cascade of strong classifiers to detect multiview faces in video sequences. A data set is composed of face and nonface images of size 20×20. A set of 6230

multiview face images of 8 persons are collected from video sequences with different reflections, illuminations and backgrounds to make face detection more robust in different scenarios. The face images are cropped and scaled to a resolution of 20×20 pixels. Another set of 6598 nonface examples with the size of 320×240 are collected from video sequences containing no faces. The nonface examples use the same size as the one employed by the video camera for real video sequence acquisition, but this is not a requirement. Figure 3.5 shows some random face examples used for the training, and Figure 3.6 shows some random nonface examples used for training. A larger training set of face and nonface examples typically leads to better detection results, although failures still exist in regions of overlap and clutter. Some results of face detection using our trained AdaBoost are illustrated in Figure 3.7. AdaBoost face detection performs well in most cases, but often leads to false positives in complicated sequences consisting of clutter or overlaps.

Figure 3.5 Face examples

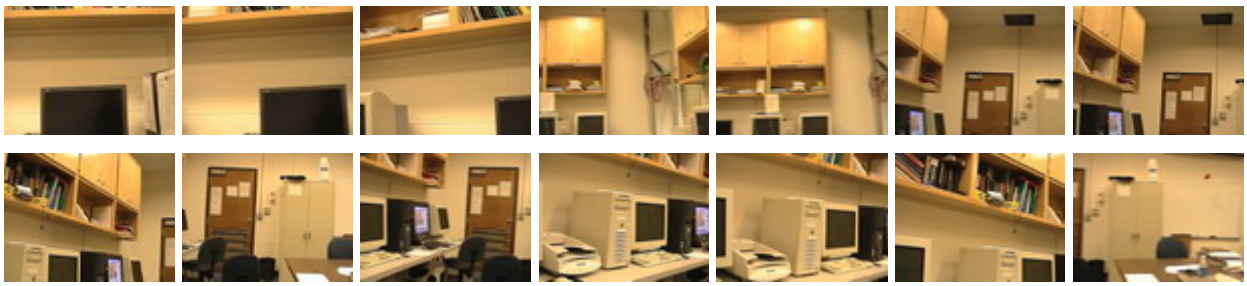


Figure 3.6 Nonface examples

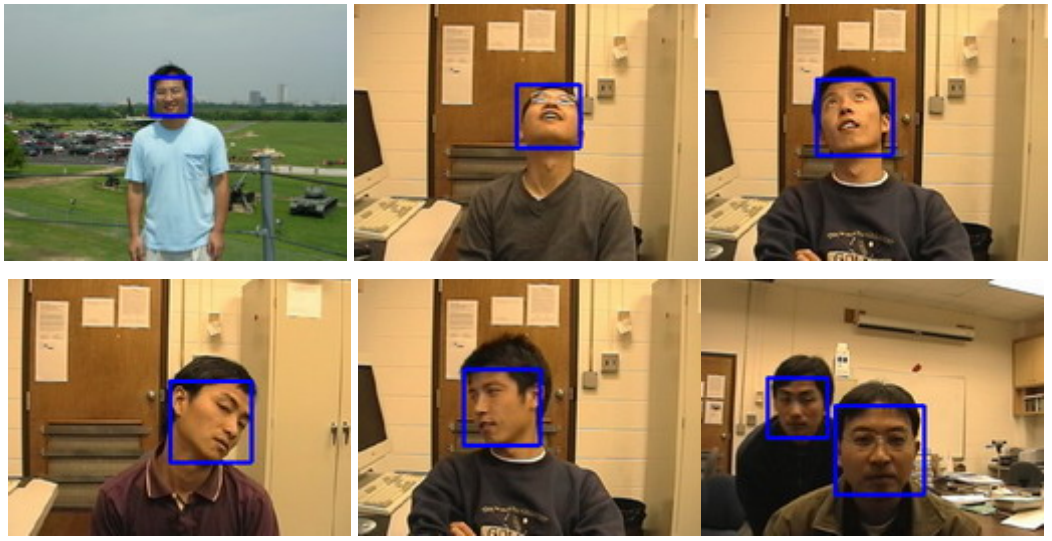


Figure 3.7 Results of frontal face detection and multiview face detection

3.5.2 Boosted Adaptive Particle Filter

The proposed boosted adaptive particle filter (BAPF) is implemented using C++ under the Microsoft Visual C++ .NET environment on a Pentium M 1.6 GHz Computer. Video sequences are of size 320×240 pixels and are sampled at 30 frames per second. In the beginning, the AdaBoost face detection model provides the initial states for the adaptive particle filter (APF) face tracking model for observations of the image sequences during a certain time interval. Since the contour defines the appearance of the face in the video sequences is roughly circular or elliptical, we use a simple parameterized model to represent the contour i.e., $Ax^2 + By^2 + C = 0$.

Of course, our method can also be applied to more complex contours that use B-spline representations. The proposed BAPF algorithm has been applied to various tracking scenarios as shown in Figure 3.8 through Figure 3.13. The tracking results from three test video sequences shown below are captured under various lighting conditions, scales, occlusions, and rotations. Two test videos (test video 1 and 3) comprise of one face in the scene, test video 1 is used for experiments in different tracking scenarios, and test video 3 is used to compare the BAPF algorithm with the Condensation Algorithm. The third test video (test video 2) that comprises two faces in the scene is used for multi-face tracking experiments on different scenarios. All tracking results are obtained using $N = 1000$ particles in the APF.

In Figure 3.8 through Figure 3.13, a yellow ellipse implies the absence of occlusion, whereas a red ellipse means that occlusion has occurred. Figure 3.8 presents the snapshots of single face tracking in test video 1 while the scale of the face changes. It shows that the proposed BAPF tracking algorithm can handle significant scale changes in the object appearance. Figure 3.9 illustrates the snapshots of single face tracking in test video 1 under changing illumination. It demonstrates that the proposed BAPF algorithm is robust to changes in various lighting conditions due to the integration of the AdaBoost statistical learning and the robustness of adaptive particle filtering. Figure 3.10 describes the snapshots of single face tracking in test video 1 under changes in viewpoint and in-plane rotations. It proves that the proposed BAPF algorithm can handle multiview face detection and tracking. Figure 3.11 provides the snapshots of single face tracking in test video 1 with in-plane rotations. It shows that the proposed BAPF algorithm can handle the appearance changes due to object rotations. Figure 3.12 illustrates the snapshots of single face tracking in test video 1 where occlusions happen. It confirms that the proposed BAPF algorithm performs correctly in the presence of occlusions because of the

robustness of adaptive particle filtering. Figure 3.13 presents the snapshots of two-face tracking in test video 2 under various tracking scenarios.

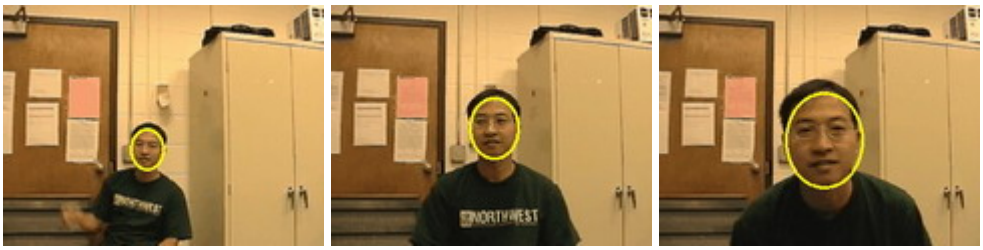


Figure 3.8 Tracking results with scale changes in test video 1. From left to right, the frame numbers are 981, 1043, and 1067.

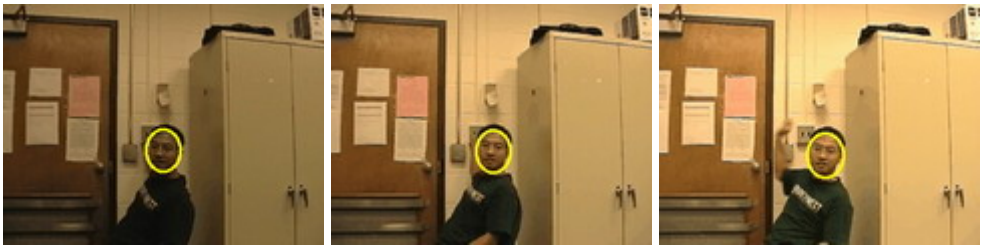


Figure 3.9 Tracking results with illumination changes in test video 1. From left to right, the frame numbers are 866, 954, and 969.

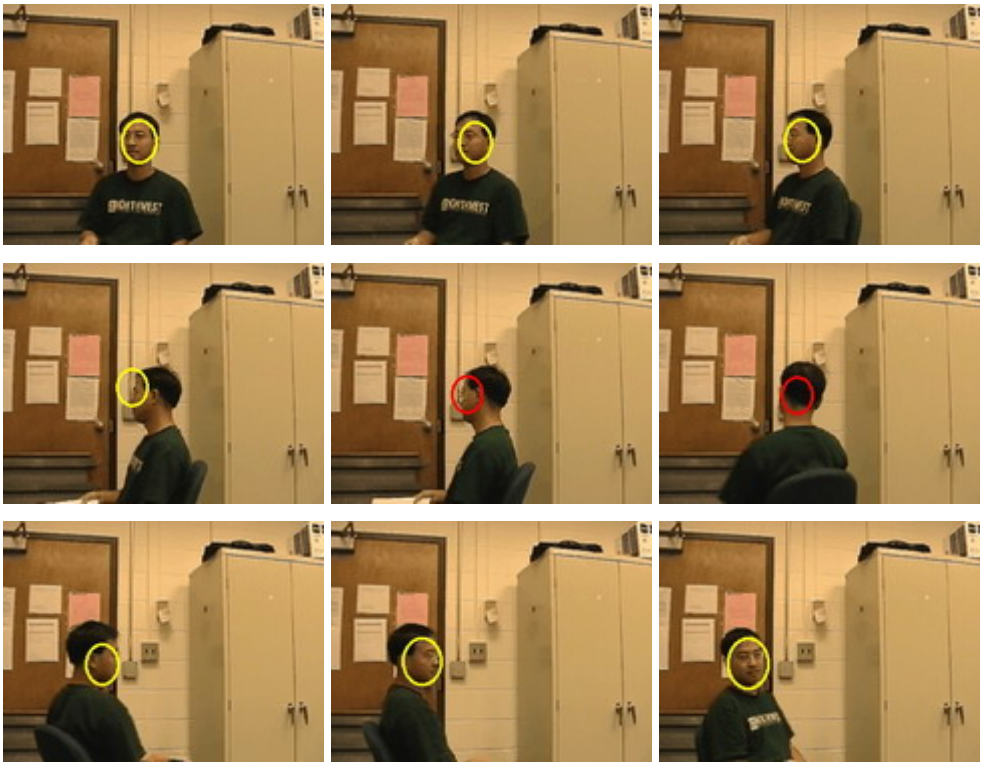


Figure 3.10 Tracking results with multiviews and rotations in test video 1. Yellow ellipse means that no occlusion has occurred, whereas red ellipse means that occlusion has occurred. From top left to bottom right, the frame numbers are 515, 519, 524, 530, 533, 544, 566, 573, and 585.

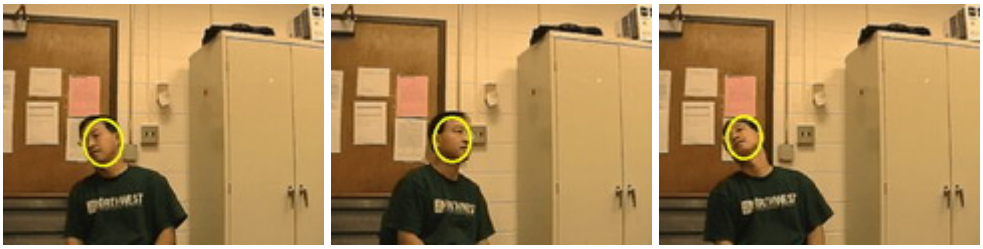


Figure 3.11 Tracking results with out-of-plane rotations in test video 1. From left to right, the frame numbers are 104, 135, and 152.

Figure 3.12 Tracking results with occlusions in test video 1. Yellow ellipse means that no occlusion has occurred, while red ellipse means that occlusion has occurred. From top left to bottom right, the frame numbers are 356, 359, 362, 366, 382, and 397.

Figure 3.13 Tracking results of two faces in test video 2. Yellow ellipse means that no occlusion has occurred, while red ellipse means occlusion has occurred. From top left to bottom right, the frame numbers are 2, 4, 25, 38, 78, and 138.

The performance of the BAPF algorithm has been compared to the Condensation algorithm, a general particle filter. Both algorithms employ $N = 1000$ particles for face tracking in the test video 3. The experimental results show that tracking accuracy of the BAPF algorithm is superior to that of the Condensation algorithm. The BAPF algorithm provides better performance than the Condensation filter. However, better performance does not necessarily mean higher computational efficiency. The BAPF algorithm actually needs more computing time than the Condensation algorithm since the BAPF algorithm performs more computation on account of the additional iterations needed to obtain better nonlinear estimations. Some examples of tracking results are presented in Figure 3.14 for the BAPF algorithm and Figure 3.15 for the Condensation algorithm.

Figure 3.14 Tracking results with the BAPF at six different times in test video 3. From top left to bottom right, the frame numbers are 12, 40, 61, 136, 158, and 180.

Figure 3.15 Tracking results with the Condensation algorithm at same times as in Figure 3.14. From top left to bottom right, the frame numbers are 12, 40, 61, 136, 158, and 180.

Using tracking accuracy and computation time, we quantitatively analyze the performance of the BAPF algorithm, the APF algorithm, and the Condensation in this section. The tracking accuracy is defined by the displacement errors between the centroid of a ground truth face and the centroid of a tracked face in video sequences. All three algorithms are tested on the test video 3, and all algorithms employ $N = 1000$ particles for face tracking. In the following quantitative analysis, the performance of the APF algorithm is compared to the Condensation algorithm, the performance of the BAPF algorithm is compared to the APF algorithm, the performance of the APF algorithm is analyzed with different values of the parameter L, the performance of the BAPF algorithm is analyzed with different values of the parameter F, and the performance of the BAPF algorithm is analyzed with different values of the parameter γ.

We compare the performance of the APF algorithm to the Condensation algorithm. Both algorithms employ $N = 1000$ particles for face tracking in the test video 3. In the APF algorithm, the number of the iterations of the loop l is $L = 3$. The experimental results, as shown in Figure

3.16 and Table 3.1, demonstrate that tracking accuracy of the APF algorithm is better than that of the Condensation algorithm. It can be seen from Table 3.1 that the mean value of the displacement error in the APF algorithm is less than that of the displacement error in the Condensation algorithm, and the computation time of the APF algorithm is greater than the Condensation algorithm. The APF algorithm thus provides better performance than the Condensation algorithm. The computation time of the APF algorithm is comparable but greater than that of the Condensation algorithm, since the APF algorithm performs more computation on account of the additional iterations needed to obtain better estimations of the proposal distribution and the posterior distribution.

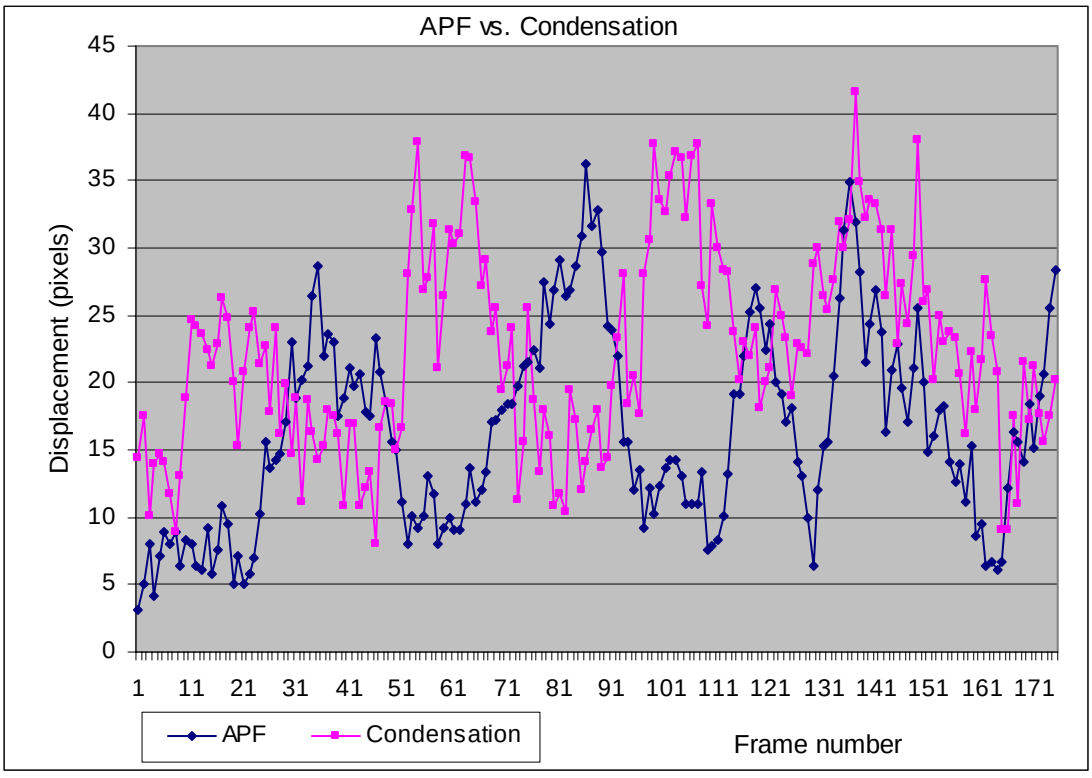


Figure 3.16 Tracking results of the APF algorithm and the Condensation algorithm

Table 3.1 Summary of tracking results of the APF and the Condensation algorithm

	APF	Condensation
Mean displacement error (pixels)	16.3	22.4
Standard deviation (pixels)	7.3	7.3
Speed (frame/sec)	4.7	6.8

The performance of the BAPF algorithm is compared to the APF algorithm. Both algorithms employ $N = 1000$ particles for face tracking in the test video 3. The number of the iterations of the loop l is $L = 3$ for both algorithms. In the BAPF algorithm, the weight assigned to the AdaBoost face detection is $\gamma = 0.8$, and the parameter F for the number of the previous frames is 1. The experimental results, as shown in Figure 3.17 and Table 3.2, demonstrate that the tracking accuracy of the BAPF algorithm is better than that of the APF algorithm. The BAPF algorithm provides better performance than the APF algorithm. It can be seen from Table 3.1 that the mean value of the displacement error in the BAPF algorithm is less than that of the displacement error in the APF algorithm, and the computation time of the BAPF algorithm is larger than the APF algorithm since the BAPF algorithm performs AdaBoost face detection in each frame.

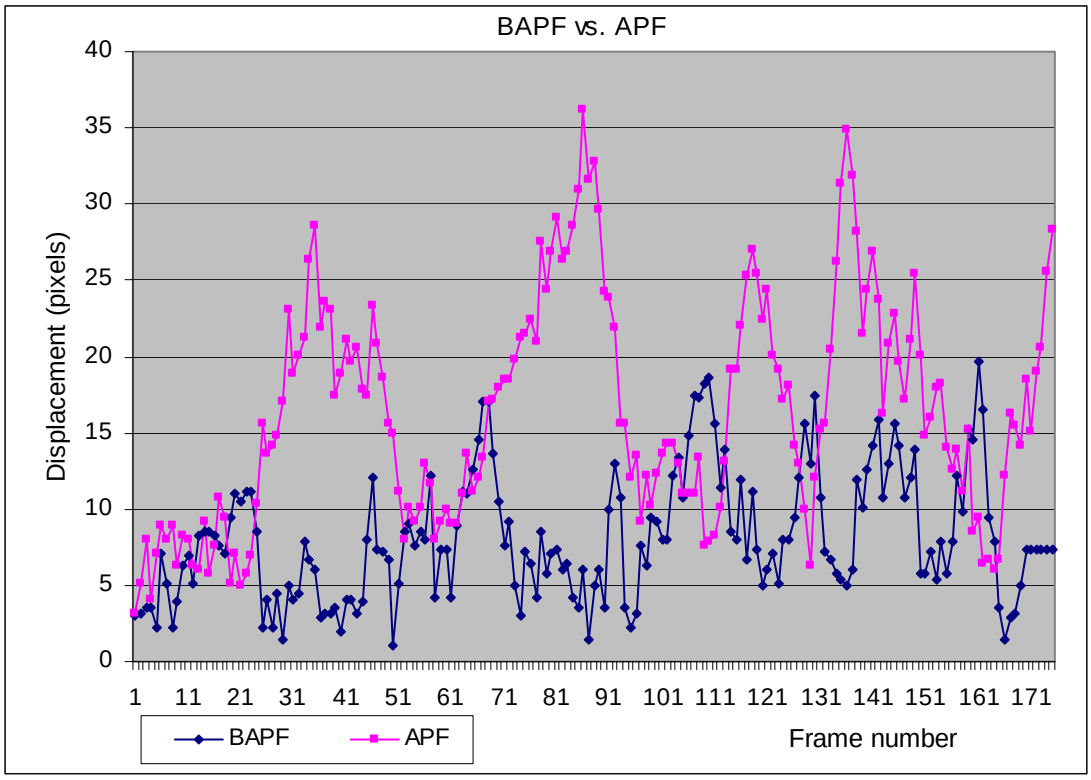


Figure 3.17 Tracking results of the BAPF algorithm and the APF algorithm

Table 3.2 Summary of tracking results of the BAPF and the APF

	BAPF	APF
Mean displacement error (pixels)	8.1	16.3
Standard deviation (pixels)	4.1	7.3
Speed (frame/sec)	4.1	4.7

The performance of the APF algorithm is analyzed using different values of the parameter L. L is the number of iterations of loop l in the APF algorithm. The APF algorithm employs $N = 1000$ particles for face tracking in the test video 3. The number of the iterations L changes from 1 to 4 in the experiments. The experimental results, as shown in Figure 3.18 and Table 3.3, demonstrate that tracking accuracy of the APF algorithm is improved as the number L increases.

It can be seen from Table 3.3 that the mean value of the displacement error in the APF algorithm with large L is less than that with small L. Thus, the APF algorithm with large L provides better performance than with small L. However, the computation time of the APF algorithm increases as the number L increases since the APF algorithm performs additional iteration to estimate the posterior distribution. When L is greater than 3, we can see from Figure 3.18 and Table 3.3 that the estimation accuracy of the posterior distribution is improved but not significantly, whereas the computation time increases significantly. We have to choose a balance between the estimation accuracy and the computation time in real applications. In our experiments, we choose $L = 3$ for the APF algorithm and the BAPF algorithm.

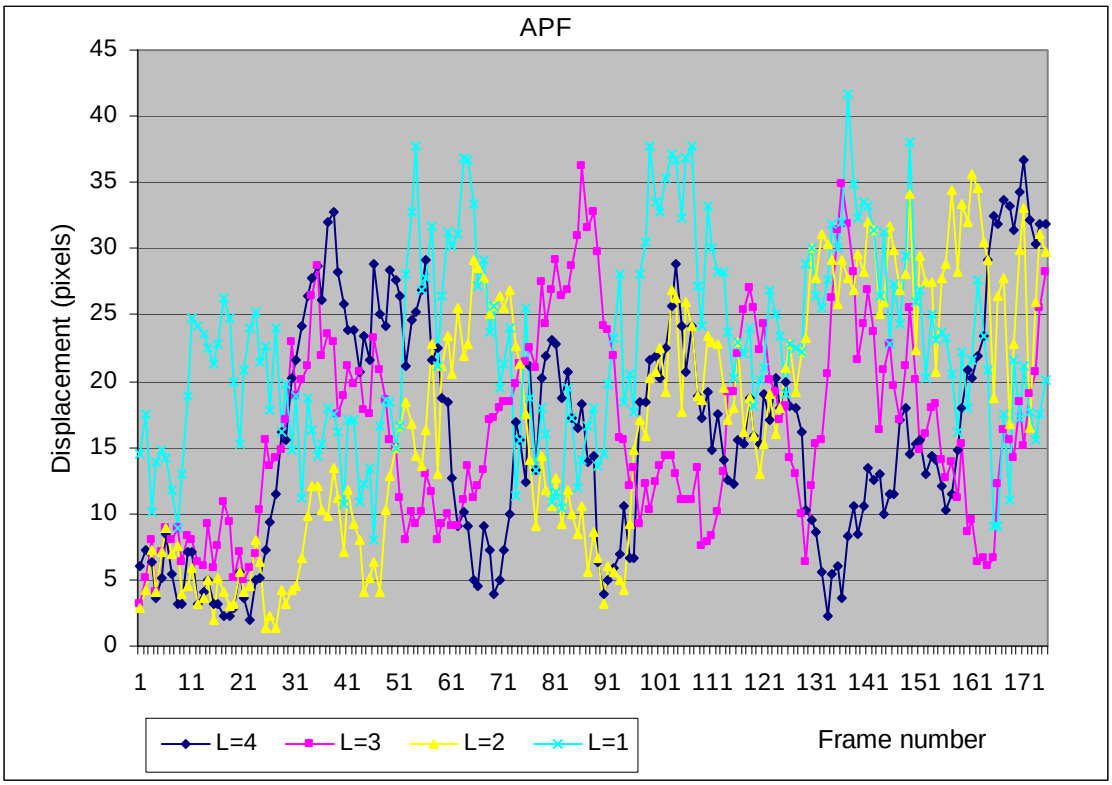


Figure 3.18 Tracking results of the APF algorithm with different values of the parameter L

Table 3.3 Summary of tracking results of the APF with different values of the parameter *L*

	L=4	*L*=3	*L*=2	*L*=1
Mean displacement error (pixels)	16.0	16.3	17.2	22.4
Standard deviation (pixels)	8.7	7.3	9.6	7.3
Speed (frame/sec)	3.2	4.7	5.8	6.8

The performance of the BAPF algorithm is analyzed based on different values of the parameter *F*. In Eq. (3-45), we combine the results of the APF algorithm and the AdaBoost algorithm to obtain the current contour of the tracked face in the current frame. The AdaBoost face detection is performed for each frame in the video sequences used in our experiments. Using AdaBoost face detection, we obtain the estimated position of a detected face based on the results of the previous *F* frames. We define *F* as the number of the previous frames for the estimation of the face in the current frame. For example, $F = 3$, we average the positions of the detected face among the previous 3 frames including the current frame to obtain an estimated position of the face in the current frame. Then we use Eq. (3-45) to combine the results of the APF algorithm and the AdaBoost algorithm.

In this experiment, the weight assigned to the result of AdaBoost face detection is $\gamma = 0.8$ in the BAPF algorithm. The number of the previous frames *F* is varied within a set of values consisting of 1, 3, 5, and 10. The number of the iterations is $L = 3$. The BAPF algorithm employs $N = 1000$ particles for face tracking in the test video 3. The experimental results, as shown in Figure 3.19 and Table 3.4, show that tracking accuracy of the BAPF algorithm is decreased as the number *F* increases. It can be seen from Table 3.4 that the mean value of the displacement error in the BAPF algorithm with large *F* is greater than that with small *F*. Thus, the BAPF algorithm with small *F* provides better performance than that with large *F*. The computation

time of the BAPF algorithm is same for different values of the parameter F. Thus, we choose F = 1 for the BAPF algorithm in our experiments.

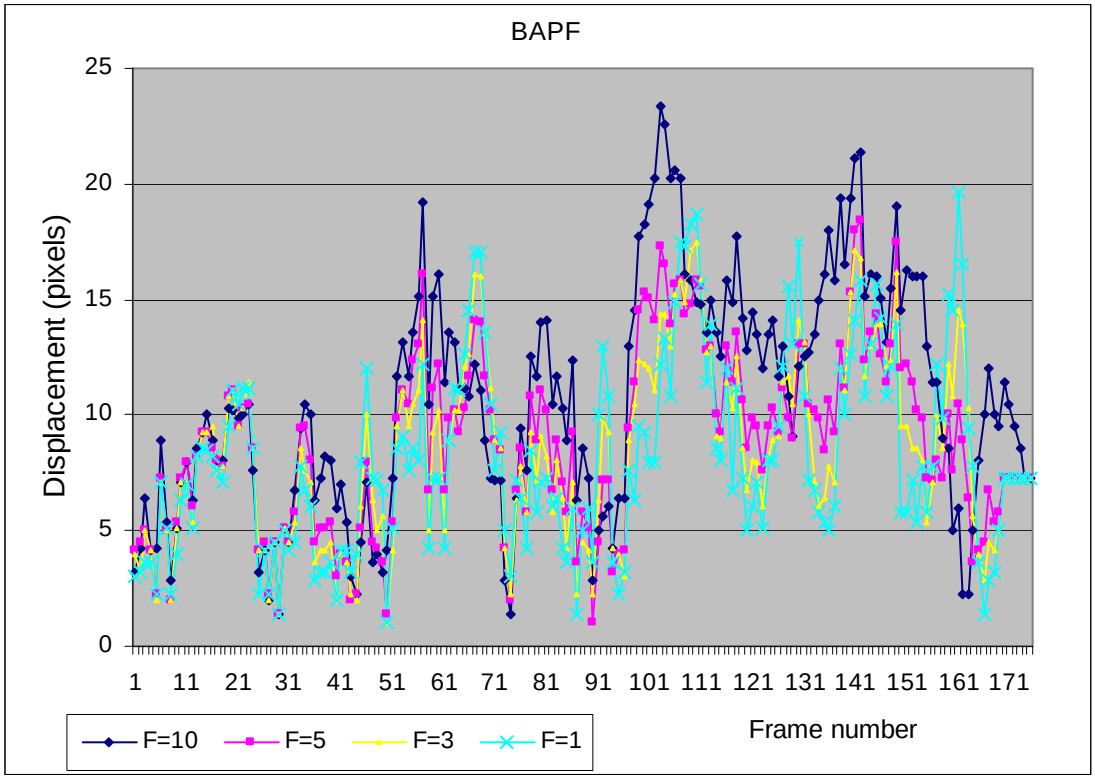


Figure 3.19 Tracking results of the BAPF algorithm with different values of the parameter F

Table 3.4 Summary of tracking results of the BAPF with different values of the parameter F

	F=10	F=5	F=3	F=1
Mean displacement error (pixels)	10.6	8.8	8.4	8.1
Standard deviation (pixels)	4.9	3.8	3.8	4.1
Speed (frame/sec)	4.1	4.1	4.1	4.1

The performance of the BAPF algorithm is analyzed using different values of the parameter γ (termed as gamma in Figure 3.20), where γ is a weight assigned to the AdaBoost face detection in the BAPF algorithm as shown in Eq. (3-45). The weight γ is varied within a set of values consisting of 0, 0.5, 0.8, and 1.0. When $\gamma = 0$, the BAPF algorithm is equivalent to the pure APF algorithm. By increasing γ, we emphasize the AdaBoost face detection. When $\gamma = 1$, the BAPF algorithm is equivalent to the pure AdaBoost algorithm. The parameter F for the number of the previous frames considered is 1. The number of the iterations of the loop l is $L = 3$. The BAPF algorithm employs $N = 1000$ particles for face tracking in the test video 3. The experimental results, as shown in Figure 3.20 and Table 3.5, show that tracking accuracy of the APF algorithm with different weights γ varies. It can be seen from Table 3.5 that the mean value of the displacement error for the BAPF algorithm with $\gamma = 0.8$ is the smallest, and the mean value of the displacement error for the BAPF algorithm with $\gamma = 0$ is the largest. The experimental results show that the performance of the pure AdaBoost algorithm is the worst, since the pure AdaBoost algorithm detects false faces or misses the tracked face in the video sequences. In the BAPF algorithm, the APF algorithm provides regions of interest to the AdaBoost algorithm, and the AdaBoost provides the detected face for the combination function in Eq. (3-45). Thus, the performance of the BAPF algorithm is better than either the APF algorithm or the AdaBoost algorithm used individually. Table 3.5 shows that the computation time of the BAPF algorithm with different γ values is the same since the APF algorithm and the AdaBoost algorithm are performed for each frame of video sequences. In our experiments, we choose $\gamma = 0.8$ for the BAPF algorithm.

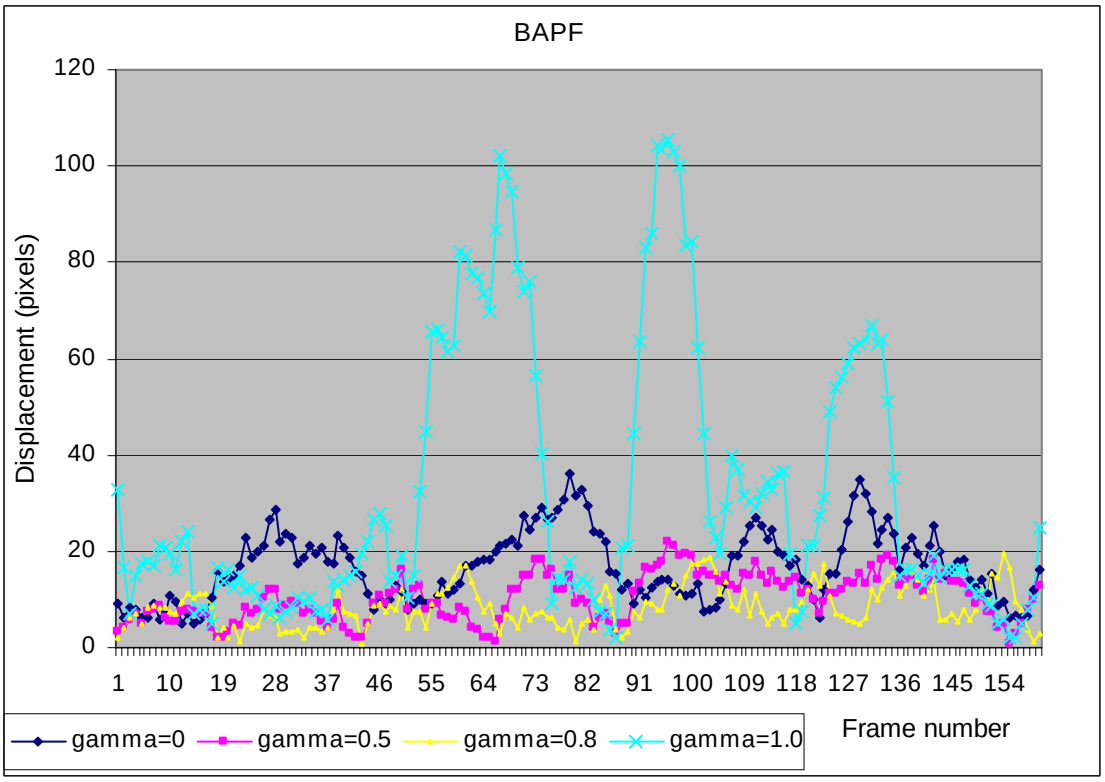


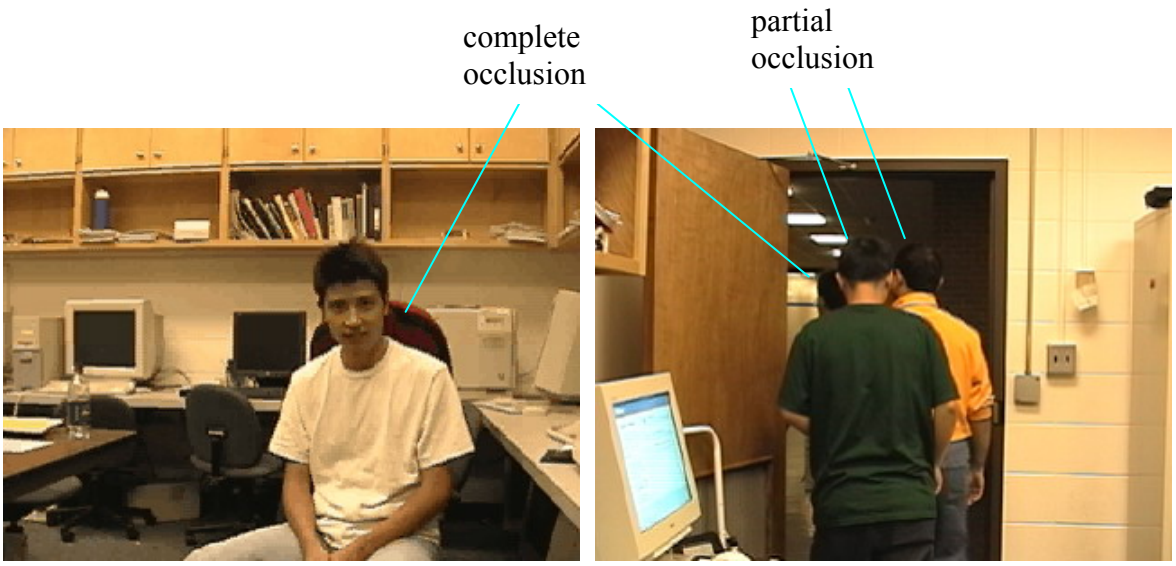
Figure 3.20 Tracking results of the BAPF algorithm with different values of the parameter γ

Table 3.5 Summary of tracking results of the BAPF with different values of the parameter γ

	γ =0	γ =0.5	γ =0.8	γ =1.0
Mean displacement error (pixels)	16.3	10.0	8.1	36.8
Standard deviation (pixels)	7.3	4.7	4.1	32.6
Speed (frame/sec)	4.1	4.1	4.1	4.1

Tracking is successful throughout except when complete occlusion occurs for long time duration, as shown in Figure 3.16. In this case, the occluded face can not be distinguished from the foreground or the background. In case of occlusion for short time duration, we can assume that the occluded face is located at the same place in the image during the occlusion. However, we can not make same assumption for occlusion over a longer time period, since the occluded

person may walk out of the scene during the occlusion. When the faces of three people are occluded and are aligned with the optical axis of the camera as shown in Figure 16 (b), it is hard to detect and track the faces of the two people in the back which results in tracking failure as well. From both cases shown in Figure 16, it is clear that from the appearance of the face alone it is not possible to reliably determine the locations of the occluded face. For correct tracking, we have to exploit more information such as the appearance of the body or the limbs to augment the BAPF algorithm to handle cases of complete occlusion over long time duration. However, the augmented appearance model for accurate tracking will increase the complexity of a dynamical model, which may reduce robustness in other cases.



(a) (b)

Figure 3.21 (a) Tracking failure in case of occlusion for a long time duration (b) Tracking failure in case of three people overlapping

3.6 Conclusions

This paper proposes a novel algorithm for face detection and tracking based on a new adaptive particle filtering algorithm and an AdaBoost algorithm. This method provides a general framework for detecting and tracking of faces in video sequences. It is also applicable to any objects such as deformable and elastic objects if appropriate contour models i.e. B-spline representations are used. Based on a new sampling technique, an adaptive particle filter (APF) is proposed to obtain accurate estimates of the proposal distribution and the posterior distribution for improving the tracking accuracy in the video sequences. The proposed scheme termed as the boosted adaptive particle filter (BAPF) combines the APF with the AdaBoost algorithm. The AdaBoost algorithm is used to detect faces in the input images, whereas the APF is used to track the faces in the video sequences. The proposed BAPF algorithm is employed for face detection, face verification, and face tracking in the video sequences. The performance of face detection and face tracking can be mutually improved in the tracking procedure.

The experimental results confirm that the proposed BAPF algorithm provides robust face detection and accurate face tracking under various scenarios, such as illumination changes, scale changes, occlusions, and rotations. The performance of the BAPF algorithm has been compared to the Condensation algorithm, a general particle filter. The experimental results show that the tracking accuracy of the BAPF algorithm is superior to that of the Condensation algorithm. The BAPF algorithm provides better tracking accuracy than the Condensation algorithm. However, the BAPF algorithm is more computationally intensive compared to the Condensation algorithm since the BAPF algorithm performs more computations on account of additional iterations needed to obtain better nonlinear distribution estimations. The experiments also show that the tracking accuracy of the BAPF algorithm is better than that of the APF algorithm, and that the

tracking accuracy of the APF algorithm is better than that of the Condensation algorithm. It is not surprising that the BAPF algorithm performs better than the linear filtering systems, such as the Kalman filter, because the BAPF algorithm overcomes the limitations of the Gaussian distribution assumption in linear filtering systems.

The problem of intensive computation in the BAPF algorithm can be alleviated by reducing the number of particles. However, the tracking accuracy becomes worse as the number of particles decrease. In a real-time application, we have to choose an appropriate balance between tracking performance and computational cost. However, currently there are no analytical results describing the mathematical relation between the number of the particles and the tracking performance for a given tracking application. In future, we hope to further analyze this tradeoff and reduce the computational cost to make the BAPF algorithm more efficient. We also expect to improve the tracking performance of the BAPF algorithm by augmenting the APF algorithm to deal with occlusions that occur frequently or persist for a long time.

References

Black, M., and Yacoob, Y., 1995, Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. *Proc. 5th International Conference on Computer Vision,* 374-381.

Blake, A., Curwen, R., and Zisserman, 1993, A., A framework for spatio-temporal control in the tracking of visual contours. *International Journal of Computer Vision*, **11** (2), 127–145.

Blake, A., Isard, M., and Reynard, D., 1995, Learning to track the visual motion of contours. *Artificial Intelligence*, **78,** 101–134.

Blake, A. and Isard, M., 1998, *Active Contours*. Springer-Verlag, London.

Borg, M., Thirde, D., Ferryman, J., Fusier, F., Valentin V., Brémond, F., and Thonnat, M., 2005, Video surveillance for aircraft activity monitoring. *International Conference on Advanced Video and Signal based Surveillance*, Como, Italy, 17-21.

Chang, C., Ansari, R., and Khokhar, A., 2005, Multiple object tracking with kernel particle filter. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*.

Crowley, J.L. and Berard, F., 1997, Multi-modal tracking of faces for video communications. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 640-645.

Curran, K., Li, X., and McCaughley, N., 2005, Neural network face detection. *Imaging Science Journal,* **53**(2), 105-115.

Darrell, T., Gordon, G., Harville, M., and Woodfill, J., 2000, Integrated person tracking using stereo, color, and pattern detection. *International Journal of Computer Vision*, **37**(2), 175-185.

Davison, A.J. and Murray, D.W., 1998, Mobile robot localisation using active vision. *The 5th European Conference on Computer Vision*, Freiburg.

Doucet, A., de Freitas, J.F.G., and Gordon N.J., 2001, *Sequential Monte Carlo Methods in Practice*. Springer, Berlin.

Doucet, A., Gordon, N.J., and Krishnamurthy, V., 2001, Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing,* **49** (3), 613–624.

Edwards, G.J., Taylor, C.J., and Cootes, T., 1998, Learning to identify and track faces in image sequences. *Proc. Sixth IEEE International Conference on Computer Vision*, 317-322.

Hansen, D.W., Hammoud, R., Boosting particle filter-based eye tracker performance through adapted likelihood function to reflexions and light changes. *International Conference on Advanced Video and Signal based Surveillance*, Como, Italy, 111-117.

Intille, S., Davis, J., and Bobick, A., 1997, Real-time closed-world tracking. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 697-703.

Isard, M., 1998, *Visual motion analysis by probabilistic propagation of conditional density*, Ph.D. thesis, University of Oxford.

Isard, M. and Blake, A., 1998, Condensation—conditional density propagation for visual tracking. *International Journal of Computer Vision,* **29** (1), 5–28.

Isard, M. and MacCormick, J., 2001, BraMBLe: A Bayesian multiple-blob tracker, *International Conference on Computer Vision*, Vancouver, Canada, **2**, 34-41.

Jebara, T., Russell, K., and Pentland, A., 1998, Mixtures of eigenfeatures for real-time structure from texture. *Proc. 6th Int. Conf. on Computer Vision*, Mumbai, India, 128-135.

Jiang, J.L. and Loe, K., 2003, S-AdaBoost and pattern detection in complex environment. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops*.

Koller, D., Weber, J.W., and Malik, J., 1994, Robust multiple car tracking with occlusion reasoning. *European Conference on Computer Vision*, Stockholm, Sweden, 189-196.

Lades, M., Vorbrüggen, J.C., Buhmann, J., Lange, J., Malsburg, C., Würtz., R., and Konen, W., 1993, Distortion invariant object recognition in the dynamic link architecture. *IEEE Trans. Computer*, **42**(3), 300-310.

Lepetit, V., Pilet, J., and Fua, P., 2004, Point matching as a classification problem for fast and robust object pose estimation. *IEEE International Conference on Computer Vision and Pattern Recognition*, Washington DC, USA, **2**, 244-250.

Li, P., Zhang, T., and Pece, A.E.C., 2003, Visual contour tracking based on particle filters. *Image and Vision Computing*, **21**, 111-123.

Li, S.Z., Zhang, Z.Q., Shum, H.Y., and Zhang, H., 2002, FloatBoost learning and statistical face detection. *IEEE Trans. On Pattern Analysis and Machine Intelligence*. **26**(9), 1112-1123.

Lienhart, R. and Maydt, J., 2002, An extended set of Haar-like features for rapid object detection. *IEEE ICIP 2002*, **1**, 900-903.

Luo, X. and Bhandarkar, S., 2005, Multiple object tracking using elastic matching. *IEEE International Conference on Advanced Video and Signal based Surveillance*, Como, Italy, 123-128.

Luokepohl, H., 1993, *Introduction to Multiple Time Series Analysis.* Spring-Verlag, 2nd edtion.

MacCormick, J. and Blake, A., 1998, A probabilistic contour discriminant for object localization. *Proceeding of Sixth International Conference on Computer Vision*, 390–395.

MacCormick, J., 2000, *Probabilistic models and stochastic algorithms of visual tracking*. Ph.D. thesis, University of Oxford.

Malik, S., Roth, G., and McDonald, C., 2002, Robust corner tracking for real-time augmented reality, *Proc. Vision Interface*, Calgary, Cananda, 399-406.

McKenna, S.J., Jabri, S., Duric, Z., Rosenfeld, A., and Wechsler, H., 2000, Tracking groups of people. *Computer Vision and Image Understanding*, **80**, 42-56.

Nummiaro, K., Koller-Meier, E., and Gool, L.V., 2003, An adaptive color-based particle filter. *Image and Vision Computing*, **21**(1), 99-110.

Okuma, K., Taleghni, A., Freitas, N.D., Little, J.J. and Lowe, D.G., 2004, A boosted particle filter: multitarget detection and tracking. *European Conf. on Computer Vision*, *LNCS 3021*, pp. 28–39.

Osuna, E., Freund, R., and Girosi, F., 1997, Training support vector machines: an application to face detection. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 130-136.

Papoulis, A., 1990, *Probability and Statistics*. Prentice-Hall.

Rabiner, L.R. and Jung, B.H., 1993, *Fundamentals of Speech Recognition*, Prentice Hall.

Rathi Y., Vaswani, N., Tannenbaum, A., Yezzi, A., 2005, Particle Filtering for Geometric Active Contours with Application to Tracking Moving and Deforming Objects. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*.

Rehg, J. and Kanade, T., 1994, Visual tracking of high dof articulated structures: an application to human hand tracking. *Proc. 3rd European Conf. Computer Vision*, Springer-Verlag, 35-46.

Reynard, D., Wildenberg, A., Blake, A., and Marchant, J., 1996, Learning dynamics of complex motions from image sequences. *In Proc. 4th European Conf. on Computer Vision*, Cambridge, England, 357–368.

Rowley, H., Baluja, S., and Kanade, T., 1996, Neural network-based face detection. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 203-208.

Rucklidge, W.J., 1995, Locating Objects Using the Hausdorff Distance, *International Conference on Computer Vision*, Massachusetts, USA, 457-464.

Samal, A. and Iyengar, P.A., 1992, Automatic recognition and analysis of human faces and facial expressions: a survey. *Pattern Recognition*, **25**(1), 65-77.

Schapire, R.E. and Singer, Y., 1998, Improved boosting algorithms using confidence-rated predictions. *Proc. 11th Ann. Conf. Computational Learning Theory*, 80-91.

Schneiderman, H. and Kanade, T., 1998, Probabilistic modeling of local appearance and spatial relationships for object recognition. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 45-51.

Schneiderman, H., 2000, *A statistical approach to 3D object detection applied to faces and cars,* PhD dissertation, R.I.

Shih, P. and Liu, C., 2004, Face detection using discriminating feature analysis and support vector machine in video. *Proceedings of the 17th International Conference on Pattern Recognition*.

Terzopoulos, D. and Szeliski, R., 1993, Tracking with Kalman snakes. *Active Vision*, MIT Press, Cambridge, USA, 3-20.

Tieu, K. and Viola, P., 2000, Boosting image retrieval. *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* **1**, 228-235.

Vermaak, J., Doucet, A., and Perez, P., 2003, Maintaining multi-modality through mixture tracking. *IEEE International Conference on Computer Vision.*

Viola, P. and Jones, M., July 2001, Robust real time object detection. *IEEE ICCV Workshop Statistical and computational Theories of Vision*.

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

Wang, H.L. and Cheong, L.F., 2005, MRF augmented particle filter tracker. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*.

Wu, Y., Hua, G., and Yu, T., 2003, Switching observation models for contour tracking in clutter. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, **I**, 295-302

Yang, M., Kriegman, D., and Ahuja, N., 2002, Detecting faces in images: a survey. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **24**(1), 34-58.

CHAPTER 4

CONCLUSIONS

This thesis provides a review of the literature related to face detection and visual object tracking. In order to address general problems in face detection and tracking, such as low detection rate, variations in lighting conditions, and partial occlusions or complete occlusions, we propose a novel scheme for face detection and tracking in this thesis bases on a novel adaptive particle filter algorithm and the AdaBoost face detection algorithm. We term the combination of the AdaBoost algorithm and the APF as a boosted adaptive particle filter (BAPF).

The proposed BAPF algorithm provides a general framework for detecting and tracking faces in video sequences. It is also applicable to any objects such as deformable and elastic objects if appropriate contour models i.e. B-spline representations are used. Based on a new sampling technique, an adaptive particle filter (APF) is proposed to obtain accurate estimates of the proposal distribution and the posterior distribution for improving the tracking accuracy in the video sequences. The proposed scheme termed as the boosted adaptive particle filter (BAPF) combines the APF with the AdaBoost algorithm. The AdaBoost algorithm is used to detect faces in the input images, whereas the APF is used to track the faces in the video sequences. The proposed BAPF algorithm is employed for face detection, face verification, and face tracking in the video sequences. The performance of face detection and face tracking can be mutually improved in the tracking procedure.

The experimental results confirm that the proposed BAPF algorithm provides robust face detection and accurate face tracking under various scenarios, such as illumination changes, scale changes, occlusions, and rotations. The performance of the BAPF algorithm has been compared to the Condensation algorithm, a general particle filter. The experimental results show that the performance of the BAPF algorithm is superior to that of the Condensation algorithm. The BAPF algorithm provides better tracking accuracy than the Condensation algorithm. However, the

BAPF algorithm is computationally more intensive compared to the Condensation algorithm since the BAPF algorithm performs more computations on account of additional iterations needed to obtain better nonlinear distribution estimations. The experiments also show that the tracking accuracy of the BAPF algorithm is better than that of the APF algorithm, and that the tracking accuracy of the APF algorithm is better than that of the Condensation algorithm. It is not surprising that the BAPF algorithm performs better than the linear filtering systems, such as the Kalman filter, because the BAPF algorithm overcomes the limitations of the Gaussian distribution assumption in linear filtering systems.

The problem of intensive computation in the BAPF algorithm can be alleviated by reducing the number of particles. However, the tracking accuracy becomes worse as the number of particles decrease. In a real-time application, we have to choose an appropriate balance between tracking performance and computational cost. However, currently there are no analytical results describing the mathematical relation between the number of the particles and the tracking performance for a given tracking application. In future, we hope to further analyze this tradeoff and reduce the computational cost to make the BAPF algorithm more efficient. We also expect to improve the tracking performance of the BAPF algorithm by augmenting the APF algorithm to deal with occlusions that occur frequently or persist for a long time.

REFERENCES

Baker, S., Szeliski, R., and Anandan, P., 1998, A layered approach to stereo reconstruction. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* 434-441.

Bhandarkar, S. and Luo, X., 2005, Fast and robust background updating for real-time traffic surveillance and monitoring. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshop*.

Black, M., and Yacoob, Y., 1995, Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. *Proc. 5th International Conference on Computer Vision,* 374-381.

Blake, A., Curwen, R., and Zisserman, 1993, A., A framework for spatio-temporal control in the tracking of visual contours. *International Journal of Computer Vision*, **11** (2), 127–145.

Blake, A., Isard, M., and Reynard, D., 1995, Learning to track the visual motion of contours. *Artificial Intelligence*, **78,** 101–134.

Blake, A. and Isard, M., 1998, *Active Contours*. Springer-Verlag, London.

Borg, M., Thirde, D., Ferryman, J., Fusier, F., Valentin V., Brémond, F., and Thonnat, M., 2005, Video surveillance for aircraft activity monitoring. *International Conference on Advanced Video and Signal based Surveillance*, Como, Italy, 17-21.

Boccignone, G., Caggiano, V., Marcelli, A., and Fiore, G.D., 2005, Interleaved detection and tracking of faces in video. *IEEE Int. Conf. on Image Processing,* **II,** 398-401.

Chang, C., Ansari, R., and Khokhar, A., 2005, Multiple object tracking with kernel particle filter. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*.

Chen, J. and Tiddeman, B., 2005, A robust facial feature tracking system. *IEEE International Conference on Advanced Video and Signal based Surveillance*, Como, Italy, 445-449.

Crowley, J.L. and Berard, F., 1997, Multi-modal tracking of faces for video communications. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 640-645.

Curran, K., Li, X., and McCaughley, N., 2005, Neural network face detection. *Imaging Science Journal,* **53**(2), 105-115.

Darrell, T., Gordon, G., Harville, M., and Woodfill, J., 2000, Integrated person tracking using stereo, color, and pattern detection. *International Journal of Computer Vision*, **37**(2), 175-185.

Davison, A.J. and Murray, D.W., 1998, Mobile robot localisation using active vision. *The 5th European Conference on Computer Vision*, Freiburg.

Doucet, A., de Freitas, J.F.G., and Gordon N.J., 2001, *Sequential Monte Carlo Methods in Practice*. Springer, Berlin.

Doucet, A., Gordon, N.J., and Krishnamurthy, V., 2001, Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing,* **49** (3), 613–624.

Edwards, G.J., Taylor, C.J., and Cootes, T., 1998, Learning to identify and track faces in image sequences. *Proc. Sixth IEEE International Conference on Computer Vision*, 317-322.

Féraud, R., Bernier, O.J., Viallet, J., and Collobert, M., 2001, A fast and accurate face detector based on neural networks. . *IEEE Trans. On Pattern Analysis and Machine Intelligence*. **23**(1), 42-53.

Gan, Z., Chan, S., and Shum, H., 2005, Object tracking and matting for a class of dynamic image-based representations. *IEEE International Conference on Advanced Video and Signal based Surveillance*, Como, Italy, 81-87.

Girondel, V., Caplier, A., et Bonnaud, L., 2004, Real time tracking of multiple persons by kalman filtering and face pursuit for multimedia applications", *IEEE Southwest Symposium on Image Analysis and Interpretation*, 201-205.

Govindaraju, V., 1996, Locating human faces in photographs. *International Journal on Computer Vision*, **19**(2), 129-146.

Graf, H.P., Chen, T., Petajan, E., and Cosatto, E., 1995, Locating faces and facial parts. *Proc. First Int'l Workshop Automatic Face and Gesture Recognition*, 41-46.

Gunn, S.R. and Nixon, M.S., 1996, Snake head boundary extraction using global and local energy minimization. *Proc. Int'l Conf. Pattern Recognition*, 581-585.

Guo, G.D. and Zhang, H.J., 2001, Boosting for fast face recognition. *Proceedings of the IEEE ICCV Workshop on Recognition, Analysis, and Tracking in Real-Time Systems*.

Hansen, D.W., Hammoud, R., Boosting particle filter-based eye tracker performance through adapted likelihood function to reflexions and light changes. *International Conference on Advanced Video and Signal based Surveillance*, Como, Italy, 111-117.

Herpers, R., Michaelis, M., Lichtenauer, K., and Sommer, G., 1996, Edge and keypoint detection in facial regions. *IEEE Proc. of 2nd Int. Conf. on Automatic Face and Gesture Recognition*, 212–217.

Hjelmås, E. and Low, B.K., 2001, Face detection: a survey. *Computer vision and image understanding*, **83**, 236-274.

Hori, Y., Shimizu, K., Nakamura, Y., and Kuroda, T., 2004, A real-time multi face detection technique using positive-negative lines-of-face template. *Proceedings of the 17th International Conference on Pattern Recognition*.

Huang, L., Shimizu, A., and Kobatake, H., 2003, Face Detection from Cluttered Images Using Gabor Filter-Based Features. *Annual Conference of SICE Japan*, WPI-3-1, pp.1150-1153.

Huang, L., Shimizu, A., and Kobatake, H., 2004, A multi-expert approach for robust face detection. *Proceedings of the 17th International Conference on Pattern Recognition*.

Huang, K.S. and Trivedi, M.T., 2004, Robust real-time detection, tracking, and pose estimation of faces in video streams. *Proceedings of the 17th International Conference on Pattern Recognition*.

Intille, S., Davis, J., and Bobick, A., 1997, Real-time closed-world tracking. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 697-703.

Isard, M., 1998, *Visual motion analysis by probabilistic propagation of conditional density*, Ph.D. thesis, University of Oxford.

Isard, M. and Blake, A., 1998, Condensation—conditional density propagation for visual tracking. *International Journal of Computer Vision*, **29** (1), 5–28.

Isard, M. and MacCormick, J., 2001, BraMBLe: A Bayesian multiple-blob tracker, *International Conference on Computer Vision*, Vancouver, Canada, **2**, 34-41.

Ishii, Y., Hongo, H., Yamamoto, K., and Niwa, Y., 2004, Face and head detection for a real-time surveillance system. *Proceedings of the 17th International Conference on Pattern Recognition*.

Jackson, J., Yezzi, A., and Soatto, S., 2004, Tracking deformable moving objects under severe occlusions. *In Conf. Decision and Control*.

Jebara, T., Russell, K., and Pentland, A., 1998, Mixtures of eigenfeatures for real-time structure from texture. *Proc. 6th Int. Conf. on Computer Vision*, Mumbai, India, 128-135.

Jiang, J.L. and Loe, K., 2003, S-AdaBoost and pattern detection in complex environment. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops*.

Koller, D., Weber, J.W., and Malik, J., 1994, Robust multiple car tracking with occlusion reasoning. *European Conference on Computer Vision*, Stockholm, Sweden, 189-196.

Kwon, Y.H. and Lobo, N., 1994, Face detection using templates, *Proc. Int'l Conf. Pattern Recognition*, 764-767.

Lades, M., Vorbrüggen, J.C., Buhmann, J., Lange, J., Malsburg, C., Würtz., R., and Konen, W., 1993, Distortion invariant object recognition in the dynamic link architecture. *IEEE Trans. Computer*, **42**(3), 300-310.

Lepetit, V., Pilet, J., and Fua, P., 2004, Point matching as a classification problem for fast and robust object pose estimation. *IEEE International Conference on Computer Vision and Pattern Recognition*, Washington DC, USA, **2**, 244-250.

Li, P., Zhang, T., and Pece, A.E.C., 2003, Visual contour tracking based on particle filters. *Image and Vision Computing*, **21**, 111-123.

Li, S.Z., Zhang, Z., 2002, FloatBoost learning and statistical face detection. *IEEE Trans. On Pattern Analysis and Machine Intelligence*. **26**(9), 1112-1123.

Li, S.Z., Zhu, L., Zhang, Z., Zhang, H., 2002, Learning to detect multi-view faces in real-time. *Proceedings of the 2nd International Conference on Development and Learning*.

Lienhart, R. and Maydt, J., 2002, An extended set of Haar-like features for rapid object detection. *IEEE ICIP 2002*, **1**, 900-903.

Lucas, B. and Kanade, T., 1981, An interactive image registration technique with an application in stereovision. *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, 674-679.

Luo, X. and Bhandarkar, S., 2005, Multiple object tracking using elastic matching. *IEEE International Conference on Advanced Video and Signal based Surveillance*, Como, Italy, 123-128.

Luokepohl, H., 1993, *Introduction to Multiple Time Series Analysis.* Spring-Verlag, 2nd edtion.

MacCormick, J. and Blake, A., 1998, A probabilistic contour discriminant for object localization. *Proceeding of Sixth International Conference on Computer Vision*, 390–395.

MacCormick, J., 2000, *Probabilistic models and stochastic algorithms of visual tracking*. Ph.D. thesis, University of Oxford.

MacCormick, J., Isard, M., 2000, Partitioned sampling, articulated objects and interface-quality hand tracking. *Proceedings of the Sixth European Conference on Computer Vision*, LNCS 1843, Springer, Berlin, 3–19.

Malik, J., Belongie, S., Shi, J., and Leung, T., 1999, Textons, contours and regions: cue integration in image segmentation. *Proc. 7th Int. Conf. on Computer Vision,* 98-925.

Malik, S., Roth, G., and McDonald, C., 2002, Robust corner tracking for real-time augmented reality, *Proc. Vision Interface*, Calgary, Canada, 399-406.

McKenna, S., Raja, Y., and Gong, S., 1998, Tracking color objects using adaptive mixture models. *Image and Vision Computing*, **17**(3), 223-229.

McKenna, S.J., Jabri, S., Duric, Z., Rosenfeld, A., and Wechsler, H., 2000, Tracking groups of people. *Computer Vision and Image Understanding*, **80**, 42-56.

Naseem, I. and Deriche, M., 2005, Robust human face detection in complex color images. *IEEE Int. Conf. on Image Processing,* **II,** 338-341.

Nummiaro, K., Koller-Meier, E., and Gool, L.V., 2003, An adaptive color-based particle filter. *Image and Vision Computing*, **21**(1), 99-110.

Okuma, K., Taleghni, A., Freitas, N.D., Little, J.J. and Lowe, D.G., 2004, A boosted particle filter: multitarget detection and tracking. *European Conf. on Computer Vision*, *LNCS 3021*, pp. 28–39.

Osuna, E., Freund, R., and Girosi, F., 1997, Training support vector machines: an application to face detection. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 130-136.

Papoulis, A., 1990, *Probability and Statistics*. Prentice-Hall.

Park, J., Choi, H., Kim, S., 2005, Baysian face detection in an image sequence using face probability gradient ascent. *IEEE Int. Conf. on Image Processing,* **II,** 346-349.

Pentland, A., 2000, Looking at people. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **22**(1), 107-119.

Pentland, A., 2000, Perceptual intelligence. *Comm. ACM*, **43**(3), 35-44.

Pentland, A. and Choudhury, T., 2000, Face Recognition for Smart Environments. *IEEE Computer*, 50-55.

Rabiner, L., 1989, A tutorial on hidden Markov models and selected application s in speech recognition. *Proc. IEEE*, **77**(2), 257-286.

Rabiner, L. and Jung, B., 1993, *Fundamentals of Speech Recognition*, Prentice Hall.

Rathi Y., Vaswani, N., Tannenbaum, A., Yezzi, A., 2005, Particle Filtering for Geometric Active Contours with Application to Tracking Moving and Deforming Objects. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*.

Rehg, J. and Kanade, T., 1994, Visual tracking of high dof articulated structures: an application to human hand tracking. *Proc. 3rd European Conf. Computer Vision*, Springer-Verlag, 35-46.

Reynard, D., Wildenberg, A., Blake, A., and Marchant, J., 1996, Learning dynamics of complex motions from image sequences. *In Proc. 4th European Conf. on Computer Vision*, Cambridge, England, 357–368.

Rowley, H., Baluja, S., and Kanade, T., 1996, Neural network-based face detection. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 203-208.

Rowley, H., Baluja, S., and Kanade, T., 1998, Neural network-based face detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **20**(1), 23-38.

Rucklidge, W.J., 1995, Locating Objects Using the Hausdorff Distance, *International Conference on Computer Vision*, Massachusetts, USA, 457-464.

Samal, A. and Iyengar, P.A., 1992, Automatic recognition and analysis of human faces and facial expressions: a survey. *Pattern Recognition*, **25**(1), 65-77.

Samal, A. and Iyengar, P.A., 1995, Human face detection using silhouettes. *Int'l J. Pattern Recognition and Artificial Intelligence*, **9**(6), 845-867.

Schapire, R.E. and Singer, Y., 1998, Improved boosting algorithms using confidence-rated predictions. *Proc. 11th Ann. Conf. Computational Learning Theory*, 80-91.

Schneiderman, H. and Kanade, T., 1998, Probabilistic modeling of local appearance and spatial relationships for object recognition. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 45-51.

Schneiderman, H., 2000, *A statistical approach to 3D object detection applied to faces and cars,* PhD dissertation, R.I.

Schneiderman, H., 2004, Learning a restricted Bayesian network for object detection. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*.

Sethian, J.A., 1989, A review of recent numerical algorithms for hypersurfaces moving with curvature dependent speed. *J. Differential Geometry*, **31**, 131–161.

Shih, P. and Liu, C., 2004, Face detection using discriminating feature analysis and support vector machine in video. *Proceedings of the 17th International Conference on Pattern Recognition*.

Song, J., Cai, M., and Lyu, M., 2002, Edge color distribution transform: an efficient tool for object detection in images. *Proceedings of the 16th International Conference on Pattern Recognition*.

Song, X., Lin, C., and Sun, M., 2004, Cross-modality automatic face model training from large video databases. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops*.

Terrillon, J., Shirazi, M., Sadek, M., Fukamachi, H., and Akamatsu, S., 2000, Invariant face detection with support vector machines. *Proceedings of the 15th International Conference on Pattern Recognition*. **4**, 4210-4217.

Terzopoulos, D. and Szeliski, R., 1993, Tracking with Kalman snakes. *Active Vision*, MIT Press, Cambridge, USA, 3-20.

Thome, N. and Miguet, S., 2005, A robust appearance model for tracking human motions. *IEEE International Conference on Advanced Video and Signal based Surveillance*, Como, Italy, 528-533.

Tieu, K. and Viola, P., 2000, Boosting image retrieval. *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* **1**, 228-235.

Tokunaga, H., Huang, L., Shimizu, A., Hagihara, Y., and Kobatake, H., 2002, Facial Characteristics Extraction Using Wavelet Transform and Its Application to Face Detection. *Proc. Forum on Information Technology Japan*, I-31.

Vermaak, J., Doucet, A., and Perez, P., 2003, Maintaining multi-modality through mixture tracking. *IEEE International Conference on Computer Vision.*

Viola, P. and Jones, M., July 2001, Robust real time object detection. *IEEE ICCV Workshop Statistical and computational Theories of Vision*.

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

Wang, H.L. and Cheong, L.F., 2005, MRF augmented particle filter tracker. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*.

Wang, W. and ertMariani, R., 2000, Face detection and precise eyes location. *Proceedings of International Conference on Pattern Recognition*.

Wang, Y., Ai, H., Wu, B., and Huang, C., 2004, Real time facial expression recognition with Adaboost. *Proceedings of the 17th International Conference on Pattern Recognition*.

Wiskott, L., Fellous, J., Krüger, N., and Malsburg, C., 1997, Face recognition by elastic bunch graph matching. *IEEE Trans. On Pattern Analysis and Machine Intelligence*. **19**(7), 775-779.

Wu, B., Ai, H., Huang, C., and Lao, S., 2004, Fast rotation invariant multi-view face detection based on real AdaBoost. *Proceedings of the 6th International Conference on Automatic Face and Gesture Recognition*.

Wu, Y., Hua, G., and Yu, T., 2003, Switching observation models for contour tracking in clutter. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, **I**, 295-302

Yang, G. and Huang, T. S., 1994, Human face detection in a complex background. *Pattern Recog.* **27**, 53–6.

Yang, M., Kriegman, D., and Ahuja, N., 2002, Detecting faces in images: a survey. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **24**(1), 34-58.

Yang, P., Shang, S., Gao, W., Li, S.Z., Zhang, D., 2004, Face recognition using Ada-Boosted Gabor features. *Proceedings of the 6th International Conference on Automatic Face and Gesture Recognition*.

Yezzi, A. and Soatto, S., 2003, Deformotion: deforming motion, shape average and the joint registration and approximation of structures in images. *Internaitonal Journal of Computer Vision*, **53**(2):153–167.

Zhang, L., Li, S.Z., Qu, Z., Huang, X., 2004, Boosting local feature based classifiers for face recognition. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops*.

Zhang, D., Li., S.Z., and Perez, D., 2004, Real-time face detection using boosting in hierarchical feature spaces. *Proceedings of the 17th International Conference on Pattern Recognition*.

Zhao, T., Nevatia, R., Lv, F., 2004, Segmentation and tracking of multiple humans in complex situations". *IEEE Trans. On Patt. Anal. and Machine Intelligence.*

Zhao, W., Chellappa, R., Phillips, P.J., and Rosenfeld, A., 2003, Face recognition: a literature survey. *ACM Computing Surveys*, **35**(4), 399-458.