UNIVERSITY OF CALIFORNIA, SAN DIEGO

Analysis of Head Pose, Faces, and Eye Dynamics in Images and Videos: A Multilevel Framework and Algorithms

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in
Electrical Engineering (Signal and Image Processing)

by

Junwen Wu

Committee in charge:

Professor Mohan M. Trivedi, Chair
Professor Sanjoy Dasgupta
Professor David Kriegman
Professor Bhaskar Rao
Professor Nuno Vasconcelos

2007

The dissertation of Junwen Wu is approved, and it is acceptable in quality and form for publication on microfilm:

Chair

University of California, San Diego

2007

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to acknowledge many people for helping me during my doctoral work. Without their persistent help and support, I would not be able to complete this work.

First, I would like to give my special thank to my advisor, Professor Mohan M. Trivedi, for his valuable time, guidance, and most importantly his generous understanding and encouragements all the time. Throughout my doctoral work he advised and encouraged me to develop independent research and problem solving skills from both systematical and scientific aspects.

My research was supported in part by grants from the UC Discovery Program and the Technical Support Working Group of the US Department of Defense. I would like to thank our great sponsors for providing me such good opportunities to be able to work freely in this area.

I am also very grateful for having an exceptional doctoral committee and wish to thank Professor Sanjoy Dasgupta, Professor David Kriegman, Professor Bhaskar Rao and Professor Nuno Vasconcelos for their inputs, valuable discussions and accessibility.

I'd like to thank my colleagues and friends from CVRR lab for the continuous support, especially Shinko Cheng, Anup Doshi, Dr. Tarak Gandhi, Dr. Kohsia Huang, Stephen Krotosky, Dr. Joel McCall, Brendan Morris, Erik Murphy-Chutorian, Dr. Sangho Park and Shankar Shivappa, for the friendship and assistance in the past several years.

I owe a special note of gratitude to Shinko Cheng for supporting me with miscellaneous tasks such as data collection and experimental evaluation environment setup.

During the last five and half years, I got to know so many friends in San Diego. I am particularly thankful to my friends Dashan Gao, Dan Liu, Zhou Lu, Duangmanee Putthividhya, Dr. Bing Shao, Dr. Yushi Shen, Dr. Deqiang Song, Haichang Sui, Zheng Wu and Dr. Hongtao Zhang. Their friendship is the most

precious gift during my PhD studies.

Finally, I thank my parents for being supportive and loving all the time. I would also like to thank my husband, Dr. Junan Zhang, for his magnanimous encouragement, continual patience and unwavering love. I would like to dedicate my dissertation to my father, mother, brother, sisters and my husband. Your love is the very foundation of my life.

The text of Chapter III, in part, is a reprint of the material as it appears in: Junwen Wu and Mohan M. Trivedi, “A Binary Tree for Probability Learning in Eye Detection.” in *Proceedings of the IEEE FRGC Workshop, in Conjunction with CVPR 2005*, San Diego, CA, 2005 and Junwen Wu and Mohan M. Trivedi, “A Binary Tree Based Probability Learning in Eye Detection: Framework and Evaluations.” Under Review, *IEEE Transactions on System, Man and Cybernetics.* I was the primary researcher of the cited material and the co-author listed in these publication directed and supervised the research which forms a basis for this chapter.

The text of Chapter IV, in part, is a reprint of the material as it appears in: Junwen Wu and Mohan M. Trivedi, “Simultaneous Eye Tracking and Blink Detection with Interactive Particle Filters.” Submitted. I was the primary researcher of the cited material and the co-author listed in this publication directed and supervised the research which forms a basis for this chapter.

The text of Chapter V, in part, is a reprint of the material as it appears in: J. Wu and M. M. Trivedi, “An Integrated Two-stage Framework for Robust Head Pose Estimation”, in *the Lecture Notes of Computer Science, IEEE International Workshop on Analysis and Modeling of Faces and Gestures (AMFG), China, in Conjunction with ICCV 2005*, 2005, Beijing and Junwen Wu and Mohan M. Trivedi, “A Two-stage Head Pose Estimation: Framework and Evaluations.” to Appear, *Pattern Recognition.* I was the primary researcher of the cited material and the co-author listed in these publication directed and supervised of the research which forms a basis for this chapter.

The text of Chapter VI, in part, is a reprint of the material as it appears in: Junwen Wu, Mohan M. Trivedi and Bhaskar Rao, “High Frequency Component Compensation based Super-Resolution Algorithm for Face Video Enhancement.” in *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, pp598-601, Aug. 2004, and Junwen Wu and Mohan M. Trivedi, “Enhancement for Face Video from Omni-directional Video Camera”, in *Proceedings of the 38th Asilomar Conference on Signals, Systems and Computers*, Asilomar, CA, November, 2004, and Junwen Wu and Mohan M. Trivedi. “Resolution Enhancement by Inter-Pixel Interference Elimination”, In Press, *Journal of Electronic Imaging*, 16(1), 2007, and Junwen Wu and Mohan M. Trivedi, “A Regression Model in TensorPCA Subspace for Face Image Super-Resolution Reconstruction”, in *Proceedings of the IEEE International Conference of Pattern Recognition (ICPR)*, August, 2006. I was the primary researcher of the cited material and the co-author listed in these publication directed and supervised of the research which forms a basis for this chapter.

VITA

1999	Bachelor of Engineering Automation, Tsinghua University, Beijing
2001	Master of Science Pattern Recognition and Artificial Intelligence, Tsinghua University, Beijing
2001–2007	Research Assistant Computer Vision and Robotics Research Laboratory Department of Electrical and Computer Engineering University of California, San Diego
2007	Doctor of Philosophy Electrical and Computer Engineering, University of California, San Diego

PUBLICATIONS

J. Wu and M. M. Trivedi. Resolution Enhancement by Inter-Pixel Interference Elimination. In Press, *Journal of Electronic Imaging*, 16(1), 2007.

J. Wu and M. M. Trivedi, A Two-stage Head Pose Estimation: Framework and Evaluations. to Appear, *Pattern Recognition.*

J. Wu and M. M. Trivedi, A Binary Tree Based Probability Learning in Eye Detection: Framework and Evaluations. Under Review, *IEEE Transactions on System, Man and Cybernetics.*

J. Wu and M. M. Trivedi, A Face Tracking and Head Pose Change Estimation Algorithm for Driver's Assistant System, Under Review, *IEEE Transactions on Intelligent Transportation Systems.*

J. Wu and M. M. Trivedi, Simultaneous Eye Tracking and Blink Detection with Interactive Particle Filters. Submitted, *IEEE Transactions on System, Man and Cybernetics.*

J. Wu and M. M. Trivedi, A Regression Model in TensorPCA Subspace for Face Image Super-Resolution Reconstruction, *in Proceedings of the IEEE International Conference of Pattern Recognition (ICPR)*, August, 2006.

J. Wu and M. M. Trivedi, A Coarse-to-Fine System Framework for Head Gesture Analysis in Intelligent Vehicle Systems, *in Proceedings of the IEEE Intelligent Vehicle Symposium (IV)*, 2006.

J. Wu and M. M. Trivedi, Robust Facial Landmark Detection for Intelligent Vehicle System, *in the Lecture Notes of Computer Science, IEEE International Workshop on Analysis and Modeling of Faces and Gestures (AMFG)*, 2005, Beijing, China, in Conjunction with ICCV 2005.

J. Wu and M. M. Trivedi, An Integrated Two-stage Framework for Robust Head Pose Estimation, *in the Lecture Notes of Computer Science, IEEE International Workshop on Analysis and Modeling of Faces and Gestures (AMFG)*, 2005, Beijing, China, in Conjunction with ICCV 2005.

J. Wu and M. M. Trivedi, Performance Characterization for Gaussian Mixture Model Based Motion Detection, *in the IEEE International Conference on Image Processing (ICIP)*, 2005, Italy.

J. Wu and M. M. Trivedi, A Binary Tree for Probability Learning in Eye Detection, *in Proceedings of the IEEE FRGC Workshop, in Conjunction with CVPR 2005*, San Diego, CA, 2005.

J. Wu and M. M. Trivedi, Enhancement for Face Video from Omni-directional Video Camera, *the 38th Asilomar Conference on Signals, Systems and Computers*, Asilomar, CA, November, 2004.

J. Wu, M. M. Trivedi and B. Rao, High Frequency Component Compensation based Super-Resolution Algorithm for Face Video Enhancement. *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, pp598-601, Aug. 2004.

J. Wu, M. M. Trivedi and B. Rao, Resolution Enhancement by AdaBoost. *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, pp893-896, Aug. 2004.

J. Wu, J. M. Pedersen, P. Putthividhya, D. Norgaard and M. M. Trivedi, A Two-level Pose Estimation Framework Using Majority Voting of Gabor Wavelets and Bunch Graph Analysis. *FG Net Workshop on Visual Observation of Deictic Gestures, in conjunction with the IEEE International Conference on Pattern Recognition (ICPR)*, Aug., 2004.

ABSTRACT OF THE DISSERTATION

Analysis of Head Pose, Faces, and Eye Dynamics in Images and Videos: A Multilevel Framework and Algorithms

by

Junwen Wu
Doctor of Philosophy in Electrical Engineering
(Signal and Image Processing)
University of California, San Diego, 2007
Professor Mohan M. Trivedi, Chair

This study is to investigate the fundamental problems of extracting and analyzing head and face related visual cues in multi-level. In the coarse level, the problem of head pose estimation is studied; and in the fine level, the problems of 1) facial feature detection and localization, especially eye features; and 2) eye dynamics, including tracking and blink detection, are studied. Algorithms frameworks for solving these problems and the experimental evaluations are presented. We first describe our contribution in the detailed level visual cue analysis, including facial feature detection, eye tracking and blink detection. Following that, the head pose estimation for images are discussed. Face super-resolution algorithms as a potential solution for obtaining more visual details are also presented.

Facial feature detection is solved in a general object detection framework and the performance over eye localization is presented. The dependency distance based on the features' empirical mutual information is used to cluster the features, such that a binary tree can be formed to represent the structure of the object feature space. The binary tree representation partitions the object feature space into compact feature subspace in a coarse to fine manner. In each compact feature subspace, independent component analysis (ICA) is used to get the independent sources, whose probability density functions (PDFs) are modeled by Gaussian mixtures. When applying this representation for the task of object detection, a sub-window is used to scan the entire image and each obtained image patch is examined using Bayesian criteria to determine the presence of an object.

After the eyes are automatically located with the binary tree-based probability learning, interactive particle filters are used for simultaneously tracking the eyes and detecting the blinks. Eye blink pattern as an important visual cue for both attentiveness analysis and fatigue indication can thereby be obtained. The particle filters use classification based observation models, in which the posterior probabilities are evaluated by logistic regressions in tensor subspaces. Extensive experiments are used to evaluate the performance from two aspects, 1) blink detection rate and the accuracy of blink duration in terms of the frame numbers; 2) eye tracking accuracy. Experimental setup for obtaining the benchmark data in tracking accuracy evaluation is also presented. A marker based commercial motion capturing system is used to provide the ground-truth. The experimental evaluation demonstrates the capability of this approach.

Besides detailed facial feature analysis, the coarser level analysis, head pose estimations, also plays an important role, such as in human-computer interaction systems. In this work the problem is formulated as a multi-class classification problem. We propose using a subspace analysis in wavelet space followed by a geometric structural analysis to solve the problem of classification with non-perfectly aligned face images. Different subspace techniques are compared for a fundamental understanding of head pose space structure.

Most head and face visual cue analysis approaches require that the image resolution is high enough to extract sufficient visual details. However, due to the physical constraints, the resolution of the input videos might be limited. Therefore, super-resolution is proposed to reconstruct more visual details about facial features. We first use an inter-pixel interference elimination approach, which is a general approach to arbitrary images. Although reasonable reconstruction can be obtained with low aliasing artifacts, the magnification factors are still limited.

An identity dependent regression model in subspaces is proposed as an alternative. High magnification factors can be obtained. The relevance model between the low-resolution face images and their high-resolution counterparts is obtained, which is used to inversely reconstruct the high-resolution face images. Occluded low-resolution images can also be reconstructed using un-occluded training samples.

A multi-level analysis of the head and face related visual cues are very important for a smart human-interface system. We present possible solutions in this work, and extensive experimental trials are done to evaluate and validate the proposed approaches. On top of the knowledge we learned using such approaches, the recognition of attentions and behaviors can be solved by a semantic interpretation of such visual cues, which indicates the further study direction.

Chapter I

Introduction

I.A Motivation

Advances in computer vision techniques boost the applications of machine understanding of human behaviors, such as human computer interface (HCI) systems [5, 6], driver's safety assistance systems [7, 8, 9, 10], and video surveillance systems [11, 12, 13, 14]. Variant application scenarios impose different types of tasks and challenges, however, some fundamental vision problems are similar. Such basic visual modules can be used in different applications. For example, in HCI systems, the challenge is to develop non-invasive vision modules that have the ability to interact with computerized equipment without need for special external equipment. Figure I.1 gives an example application of the HCI systems, which utilizes head gesture to control the computers. In the fine level, facial features are defined, detected and tracked so that higher level descriptions can be obtained. Such descriptions include the finer level visual cues, such as the blinks and eye gaze directions; as well as the coarser level visual cues, such as the head poses and the facial expressions. These visual cues can be further analyzed to formulate a semantic description of the human subject's behavior. In this way, a better perceptual interface between the computers and the human subjects can be obtained. The similar idea can also be used for driver's safety assistance systems. Such multi-level visual cues as 1) the finer level detail like eye blinks; and 2) the coarser level analysis like head poses, play an important role for driver's attentiveness analysis. Eye blink patterns can be used for evaluating the driver's alertness, and the head pose estimates can be used for determining the driver's focus of attention. These visual modules can be shared by different applications, such as video surveillance systems and human computer interaction systems, and thus attract broader interest. In this work, we focus on the study of general head and face related visual modules that can be easily adapted into variant application contexts. The study includes both the detailed level analysis: 1) facial feature detection and localization, especially eye localization; 2) eye blink detection; and

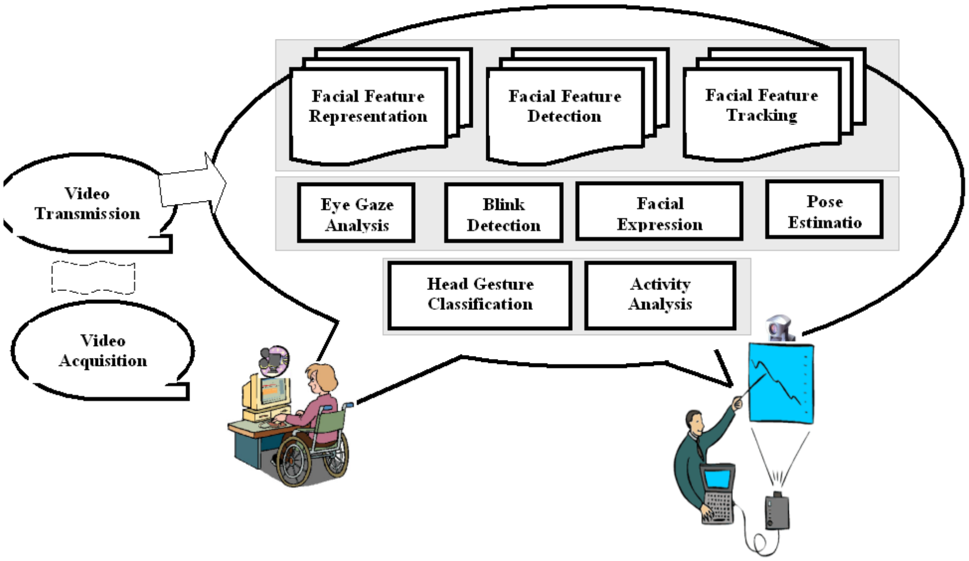


Figure I.1: An example application of the HCI system

the coarser level analysis: 3) head pose estimations. Also, face super-resolution reconstruction algorithms are discussed. This provides a possible solution when image resolutions are limited by the physical constraints.

Eye blink pattern is an interesting indicator of a human subject's attentiveness. Studies show that eye blink duration has a close relation with a subject's drowsiness [7]. The openness of eyes, as well as the frequency of eye blinks, implies the level of a person's consciousness [15]. Also, eye blinks can be used as a method of communication for people with severe disabilities, in which blink patterns are interpreted as semiotic messages [6, 16, 17]. This provides an alternate input modality to control a computerized equipment. The duration of eye closure determines whether the blink is voluntary or involuntary. Long voluntary blinks are interpreted according to the predefined semiotics dictionary, while short involuntary blinks are ignored. From the computer vision point of view, there are two challenges for a blink detection system:

1. the problem of automatically locating the eyes;

2. the problem of eye tracking and blink detection.

We solve the automatic eye localization task in a more general context, such that it can be easily generalized to other systems. A binary tree representation is used to describe the object's feature space structure, such that its probability distribution can be learned accordingly. The learned probability can be used for object detection, including the facial feature detection and localization. Specifically, we use it for the task of automatic eye localization. After eyes are automatically located, eye tracking and blink detection are simultaneously realized by interactive particle filters.

Many "smart" systems are devoted to determining the human's identity and activities across different scenarios. Head pose is an important visual cue for scene interpretation and human-computer interaction [10, 18, 19]. In most applications, head pose is determined by both the pan angle β and the tilt angle α as shown in the top right image of Figure I.2. It can be used for the subject's attentiveness analysis. Figure I.2 gives some examples of the potential applications. Besides that, head pose estimation is also very useful to provide a front-end processing for analyzing face images from different views. Many appearance-based face detection, recognition and facial expression classification systems are designed for a single view, and these algorithms require the input face images well-aligned and properly normalized. The performance degrades drastically under the appearance changes caused by head pose changes. Accurate head pose estimation can provide necessary information to build a mapping between the side-view faces and the frontal view faces. One approach is to reconstruct the frontal view faces with face images of other views. In [20], an example in multi-view facial expression recognition is given. This requires a continuous pose estimation. Another approach is to select the best view-model for detection and recognition [21, 22]. We focus on providing a subject-independent pose estimator which covers a wide range of pose angles and is capable of classifying head poses at a fine resolution. The main challenge is the robustness to face alignment and background. A two-stage frame-

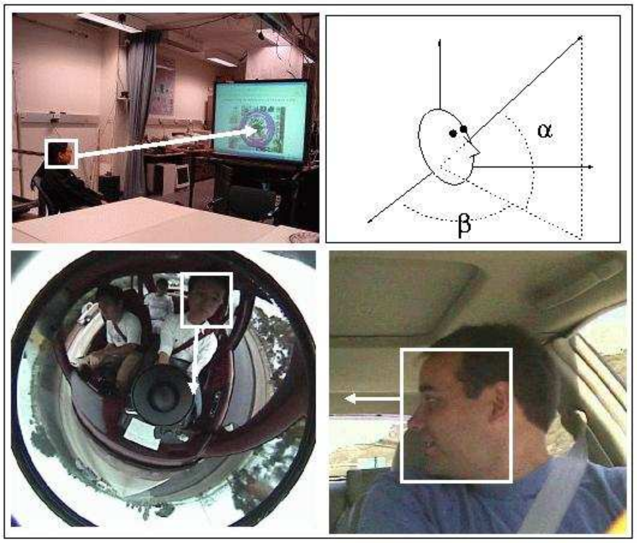


Figure I.2: Illustration of the head pose determination. Top right: illustrate the pose estimation problem (to determine the angles α and β); the other three figures: examples of the typical applications of the head pose estimation.

work is presented and evaluated, which combines the holistic statistical subspace analysis together with the geometric structure analysis for more robustness.

State of the art techniques improve the performance of such visual modules to a large extent, however, for most techniques, the performance also relies on the quality of the input videos. In general, the spatial resolution of images is limited by the optical imaging systems as well as the transmission media. Instead of challenging the limit of the hardware, super-resolution reconstruction techniques provide an alternative way to meet the capability of the display devices as well as the requirements of the successive visual computing processes. For low-resolution video input, appropriate resolution enhancement algorithms can provide more details, which attracted considerable attention from the computer vision researchers during the past several years [23].

I.B Outline

In chapter Chapter II, the related works are reviewed and a short survey is given. In chapter Chapter III, we present the framework for probability learning with the proposed binary-tree based representation. The motivation, algorithm framework and experimental evaluations are described in details. The feature dependencies are studied to form a tree representation that is capable to extract the statistical parts in a coarse-to-fine manner. At any level, the probability of the feature subset can be recursively estimated from its subtrees. The tree is built in a top-down fashion and the feature distribution is learned bottom-up. A general object representation is obtained by this way, which can be used for object detection. Automatic eye detection and localization problem can be regarded as a special application.

In chapter Chapter IV, we present a system that simultaneously tracks eyes and detects eye blinks. Two interactive particle filters are used for this purpose, one for the closed eyes and the other one for the open eyes. Each particle filter is used to track the eye locations as well as the scales of the eye image patches. The performance is carefully evaluated from two aspects: the blink detection rate and the tracking accuracy. The blink detection rate is evaluated using videos from varying scenarios, and the tracking accuracy is given by comparing with the benchmark data obtained using the Vicon motion capturing system. The set-up for obtaining benchmark data for tracking accuracy evaluation is presented and experimental results are shown.

In chapter Chapter V, a two-stage approach is proposed to address the issue of head pose estimation with un-perfectly aligned face images. The two-stage approach combines the subspace analysis together with the topography method. The first stage is based on the subspace analysis of Gabor wavelets responses. Different subspace techniques were compared for better exploring the underlying data structure. In the second stage, the pose estimate is refined by analyzing finer geometrical structure details captured by bunch graphs. We examined 86 poses, with the pan angle spanning from -90^o to 90^o and the tilt angle spanning from -60^o to 45^o. Detailed experimental evaluations are presented.

In chapter Chapter VI, two resolution enhancement algorithms are proposed: one is a general resolution enhancement algorithm, which is realized by iteratively estimating and eliminating inter-pixel interference from neighboring pixels; and the other one is realized by learning a regression model in tensorPCA subspace to get a relevance model between high-resolution face images and their counterparts. Experimental evaluation on varying inputs, including faces, synthetic text subjects, as well as license plates, validates the algorithm. Chapter Chapter VII summarizes the work and gives the concluding remarks.

Chapter II

Related Work

II.A Automatic Eye Localization

Robust and reliable eye localization is an important preprocessing step towards the development of user interfaces that are capable of analyzing user's attentiveness, such as analyzing the focus of attention for intelligent rooms [12, 24], driver's fatigue analysis for intelligent vehicle systems [8, 25, 26], and analyzing the awareness of surrounding environments in [9, 27]. Many efforts have been sought to capture the essential physical and emotional information obtainable from analyzing the eyes, such as eye gaze analysis [18, 26] and blink pattern classification [6, 7, 15, 17, 28, 29, 30]. In the other hand, numerous existing face detection and recognition algorithms also require accurate face alignment. Eye locations are usually used to provide useful information for face localization and alignment [31, 32]. Study shows that eye localization has a noticeable impact on face recognition accuracy [32]. Accurate eye localization is important for a successful face detection and recognition system. General eye detection algorithms can be categorized into two classes: traditional image appearance based approaches [8, 24, 25, 31, 32] and active illumination based approach [26, 33, 34]. Traditional image appearance based approaches usually use visual appearance features, for instance, textures, color features and geometrical structures; while active illumination based approaches use the physiological properties of eye pupils under infra-red lightings. For IR illumination based approaches, additional hardware settings are required at the image acquisition step, which limits the potential applications. The central problem is the robustness to the external illumination variations for outdoor applications, which is still an open challenge. In this work, the statistical structure of eyes is studied to get a general object representation model for the task of automatic eye detection and localization, which is an appearance based model.

Applying machine learning techniques to study the visual statistical structure of objects is a research topic that has received considerable interest since the late 70's. The work is built upon studies from two closely related communities: computer vision and machine learning. Techniques presented in [35, 36, 37] summarize the earliest work in this area. Enormous efforts have continued since then.

The research efforts start from the point of *what makes a good feature* [38]. Efforts are made towards finding the *best* feature descriptor for object detection. Here, *best* is defined with respect to the feature's ability to discriminate object image patches from background image patches. There are descriptive features designed for the specified target object, such as shape features for pedestrian detection [39], skin color features for face detection [31], and statistical features like eigen-face for face recognition [40]. In the past decade, more research interests have been directed towards the study of generic feature descriptors for general object representation, such as the the scale-invariant salient feature detectors [41][42]. On the contrary, another stream of recent research focuses on the discrimination information between objects and background. Features are defined as the statistical discriminative information between the object and background. For example, in [43], the *best* set of wavelet coefficients extracted from the SVM training procedure was used for object detection. In [44], integral image features found by a cascade of AdaBoost classifiers were proposed for a fast object detection.

The individual features can be grouped together so as to get a semantic structure. Instead of studying individual contributions of each element, features are grouped to form feature cliques. The statistics of the feature cliques as well as their relationships are studied to get a global concept, or *part-based* object representation [44, 45, 46, 47, 48]. In Figure II.1 we use an eye example to illustrate the idea. From the empirical distributions of the pixels, we can see that some pixels have closer statistical similarities while others do not. This implies that the features may have varying statistical correlations and can be studied in cliques. In [45] and [48], concept of *parts* was proposed to represent local structures that have explicit physical correspondences, such as arms, heads, torsos etc. In relation to this, others have shown interest in defining parts based on the statistical dependencies of features [44, 46, 47]. We refer to such parts as *statistical parts*. Objects can hence be represented by the relationship between statistical parts together with the part's statistics, which is referred to as statistical structure of the object.

Statistical structure of an object can be characterized using its probability density function. Bayesian criterion states that given the true distributions, Bayesian criterion can provide us an optimal classifier. Accordingly, the object detection/classification problem can be reformulated as a problem of finding the ratio of posterior probabilities [49]. This motivates us to find the statistical structure that best organize the low-level features, such that a more accurate PDF estimator for the object can be obtained. In [47] a restricted Bayesian network was proposed that clearly stated the concept of learning feature dependencies. This approach demonstrated how component based detection and recognition algorithms can achieve good performance. However, the proposed restricted Bayesian network assumes that object parts only exhibit a first order dependency, which implies an assumption that all parts have the same scale and the high order dependencies between features are ignored. In this work, we propose using a binary tree representation, which is capable to analyze the object feature space in a coarse to fine fashion.

II.B Eye Tracking and Blink Detection

Eye blink detection has attracted considerable research interest from the computer vision community. In literature, most existing techniques use two separate steps for eye tracking and blink detection [6, 15, 17, 28, 29, 50]. For blink detection systems, there are three types of dynamic information involved: 1) the global motion of eyes; 2) the local motion of eye pupils; and 3) the eye openness/closure. Accordingly, an effective eye tracking algorithm for blink detection purposes needs to satisfy the following constraints:

- Track the global motion of eyes;

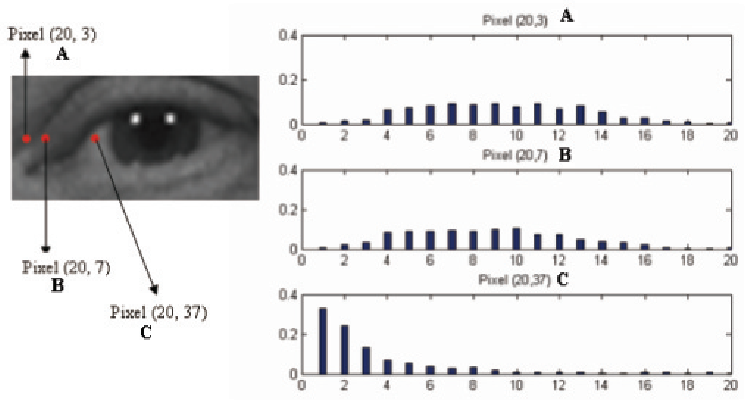


Figure II.1: Examples of the feature's empirical distributions: pixel intensity for the eye samples.

- Maintain invariance to local motion of eye pupils;
- Classify the closed-eye frames from the open-eye frames.

Once the eyes' locations are estimated by the tracking algorithm, the differences in image appearance between the open eyes and the closed eyes can be used to find the frames in which the subjects' eyes are closed, such that eye blinking can be determined. In [15], template matching is used to track the eyes and color features are used to determine the openness of eyes. Detected blinks are then used together with pose and gaze estimates to monitor the driver's alertness. In [28, 30], blink detection is implemented as part of a large facial expression classification system. Differences in intensity values between the upper eye and lower eye are used for eye openness/closure classification, such that closed-eye frames can be detected. The use of such low-level features makes the real-time implementation of the blink detection systems feasible. However, for videos with large variations, such as the typical videos collected from in-car cameras, the acquired images are usually noisy and with low-resolution. In such scenarios, simple low-level features, like color and image differences, are not sufficient. Also, temporal information is used by some other researchers for blinking detection purposes. For example, in [6, 17, 50], a human-computer interaction system exploiting eye blinks is presented to provide a possible new solution that can be used by highly disabled people. The image difference between neighboring frames is used to locate the eyes, and the temporal image correlation is used thereafter to determine whether the eyes are open or closed. In [29], the dense motion field estimated from dense optical flow describes the motion patterns, in which the eye lid movements can be separated to detect eye blinks. The ability to differentiate the motion related to blinks from the global head motion is essential. Since face subjects are non-rigid and non-planar, it is not a trivial work.

Such two-step based blink detection system requires that the tracking algorithms are capable of handling the appearance change between the open eyes and the closed eyes. In this work, we propose an alternative way that simultaneously tracks eyes and detects eye blinks.

II.C Head Pose Estimation

Over the past several years, head pose estimation remains as an active research topic. Good head pose estimation algorithms should be independent with the subjects' identity as well as the surrounding environments. If there are multiple images available, pose position in the 3D space can be recovered using the face geometry. The input could be video sequences from single camera [10, 20, 51, 52] as well as images from multiple cameras [39, 53]. Following techniques have been proposed: 1) feature tracking, and 2) multi-modal information fusion.

Tracking-based Approach

Feature tracking, including tracking the local salient features [20, 52] or the geometric features [10, 51] usually can be used to evaluate the relative deformation from a reference frame, such that the relative 3D head pose change can be recovered. In [52], Horprasert et al. show that 5 points are sufficient to recover the head pose by tracking, with the anthropometric data given. A first stage was used to recover a subjects gender, race and age from the appropriate table of anthropometric data, and then a second stage is performed to estimate the pose by salient facial point tracking. The 5 points they used are the 4 eye corners as well as the nose point. In [20], Braathen et al. proposed to use multiple particle filters to track the pre-defined facial features so that the head pose can be estimated. In [10, 51], the image differencing and ellipse fitting were suggested as the geometric feature for tracking the head pose.

Multi-modal Information Fusion

In [39, 53], the joint statistical property of the image intensity and the depth information were studied. View-based eigen-spaces from both the intensity images and the depth images are computed to reconstruct all available views of a new subject. This is used as a prior model. The pose-change from the prior is computed using a Kalman filter and then used to estimate the head pose.

With only static images available, the 2D head pose estimation problem has presented a different challenge. There is no temporal dynamic information, so that the focus of the research is on what is the best data space to describe the pose information and how to effectively estimate the pose. 2D pose estimate can be used as the front-end procedure for multi-view face analysis [21, 54] (such as detection and recognition); as well as providing the initial reference frame for 3D head pose tracking. In [55], the author investigated the dissimilarity between poses in transformed feature space such as Gabor wavelet coefficients and Principal Component Analysis (PCA). This study indicates that identity-independent pose can be discriminated by prototype matching with suitable filters. Various efforts have been put to investigate the 2D pose estimation problem [21, 22, 54, 56, 57, 58] and they are mainly focused on the use of statistical learning techniques, such as the support vector classification (SVC) in [21], kernel Principal Component Analysis (KPCA) in [54], multi-view eigen-space in [57], eigen-space from *best* Gabor filters in [56], active appearance model (AAM) in [59], manifold learning in [22] and graph embedded analysis in [58] etc. In table Table II.1, a detailed comparison of the literatures is summarized. All these algorithms are based on the features from entire faces. The use of holistic face features requires that the face samples are well-aligned and properly scaled. However, the output from face detectors usually cannot satisfy such requirement, which will cause the deterioration in the accuracy. Some researchers also explored the problem by utilizing the geometric structure constrained by representative local features [60, 61]. In [60], the authors extended the bunch graph work from [62] to pose estimation. The technique provides the idea to incorporate the geometric configuration to 2D head pose estimation. In [61], Gabor wavelet network (GWN), which is constructed from the Gabor wavelet coefficients of local facial features, was used to estimate the head poses. However, the use of geometric configuration encounters two problem: 1) how to be identity independent; 2) the computational cost from the exhaustive search. In this work, a two state head pose estimation framework is presented, which combines the holistic statistical analysis and the geometric configuration analysis.

II.D Face Video High Resolution Reconstruction

Image super-resolution is defined as the process to estimate the higher-resolution images from images with lower-resolutions. Traditionally, the terminology is only used for the problem of recovering a single image from multiple low-resolution input, which are different samples of the same scene [3, 4, 63, 64, 65, 66]. Non-redundant information from the given input is used to reconstruct the higher-resolution image. The concept has been extended to incorporate the work of recovering a higher resolution image with single image input [2, 67, 68, 69, 70, 71, 72, 73, 74]. Since the high resolution images that can produce the same low-resolution image are not unique, the super-resolution reconstruction problem is ill-conditioned. Super-resolution problem has been criticized as impossible because it seeks to recover the information that has been lost, however, the successes of recent super-resolution algorithms proved that it is solvable.

Table II.1: A comparative view of existing head pose estimation approaches.

References	Input Data	Approach	Comments
Kohsia et.al [10]	Video sequence	Tracking head outline	Real-time Omni-directional camera Requires good outline detection and tracking
Braathen et.al [20]	Video sequence	Particle filtering facial features	Easy to generate 3D face model Manual initialization
Y. Li et.al [21]	Static images	SVC regression	Actual angles Sensitive to Illumination and alignment
S. T. Li et.al [22]	Static images	Kernel Machine learning	Accurate classification Pan angle only Requires good alignment
Cordea et.al [51]	Video sequence	Tracking head outline	Real-time Requires good outline detection and tracking
Horprasert et.al [52]	Video sequence	Tracking 5 facial features	Real time Requires anthropometric data
Morency et.al [53]	Intensity images Depth images	3D view-based eigen-spaces	User independent Requires additional depth information
Chen et.al [54]	static images	Manifold learning	Manifold representation Limited to $\pm 10^{o}$ Requires good alignment
Wei et.al [56]	Static images	Eigen-space for Gabor features	Good generalization Pan angle only Requires good alignment
Fu et.al [58]	Static images	Graph Embedded Analysis	Continuous pan angle estimate Tilt angle not considered Requires good alignment
Kruger et.al [61]	Static images	Gabor Wavelet Network	Precise pan and tilt angels Pose range limited
Two-stage (subspace+ topography)	Static images	Subspace + Geometric constraint	Good generalization Both pan and tilt angles Tolerant small mis-alignment Higher computational cost

To recover a higher-resolution image from multiple low-resolution samples, most existing techniques solve the problem in a two-step manner: motion estimation followed by the back projection from the low-resolution image pixels onto the high-resolution grid. Some techniques concentrate on the re-sampling procedure during image degradation [63, 66]; while some others are focused on finding an appropriate regularization for the ill-conditioned problem [3, 4, 65, 75, 76, 77].

In [63], Baker et al. proposed to use dense optical flow for estimating the sub-pixel motion, so as to find the corresponding pixel values residing on the uniform high-resolution lattices. The reconstructed higher-resolution images are used again to refine the motion estimate, accordingly, the higher-resolution image is repeatedly estimated. However, accurate dense optical flow estimation requires sufficient textures. The lack of the texture will cause ambiguity in motion, so that aliasing will occur. In [66], non-uniform sampling was considered and reconstruction was based on a convex set constraint. Sub-pixel motion estimation is still necessary for re-sampling over a denser grid, and for regions without sufficient texture, aliasing will be a problem. However, most algorithms for re-sampling on the denser high-resolution grid only require local operation for both motion estimation and back-projection, hence they are very computationally efficient.

Different from the re-sampling algorithms, the forward model defines super-resolution as an inverse problem, which can be represented in both the frequency domain and the spatial domain [75]. Correspondingly, algorithms in both frequency domain and spatial domain were studied. The earliest significant work on frequency domain algorithms can be dated back to 1984 [64]. In [64], pure global translational motion of the camera was considered as the only source for variations between input frames. Before quantization, the t-th observations $I_o(\mathbf{x};t)$ were related to the source I_s by:

$$I_o(\mathbf{x};t) = I_s(\mathbf{x}+\mathbf{\Delta}_t), t = 1, \cdots, R; \tag{II.1}$$

where $\mathbf{\Delta}_t$ describes the translational motion for the t-th observation. It can also be represented in the frequency domain as:

$$\mathcal{F}\{I_o(\mathbf{x};t)\} = e^{j2\pi(\mathbf{\Delta}_t^{\mathrm{T}}\mathbf{w})}\mathcal{F}\{I_s(\mathbf{x})\}. \tag{II.2}$$

Considering impulse sampling, the *Continuous Fourier Transform* (CFT) of the scene $\mathcal{I}_o = \mathcal{F}\{I_o\}$ is related to the *Discrete Fourier transform* (DFT) of the shifted and sampled low-resolution images, which is $\mathcal{I}_l(\mathbf{w};t)$, via aliasing:

$$\mathcal{I}_l(\mathbf{w};t) = \frac{1}{T_1T_2}\sum_{i_1=-\infty}^{+\infty}\sum_{i_2=-\infty}^{+\infty}\mathcal{I}_o(\frac{w_1}{N_1T_1}+\frac{i_1}{T_1},\frac{w_2}{N_2T_2}+\frac{i_2}{T_2};t); \mathbf{w} = (w_1, w_2); \tag{II.3}$$

where T_1 and T_2 are sampling intervals; N_1 and N_2 are respectively the magnification factor in each direction. Equation (II.2) and (II.3) are combined together to solve the discrete Fourier transform (DFT) of the observed images. Inverse DFT is then applied to reconstruct the high-resolution images. This computationally efficient model, although sensitive to the modeling error and hard to include local motion, has brought further researches into the frequency domain techniques.

Spatial domain approaches describe the generative deterioration models directly in the image domain. If $\mathbf{I}_h$ and $\mathbf{I}_l$ are, respectively, lexicographically ordered high-resolution images and its corresponding low-resolution images, the commonly used forward model can be described as follows:

$$\mathbf{L} = \mathbf{A}_1 \times \cdots \times \mathbf{A}_a\mathbf{H} + \mathbf{n};$$

$$\mathbf{H} = [\mathbf{I}_h(t-p), \cdots, \mathbf{I}_h(t+p)];$$

$$\mathbf{L} = [\mathbf{I}_l(t-p), \cdots, \mathbf{I}_l(t+p)]. \tag{II.4}$$

$\mathbf{A}_1, \cdots, \mathbf{A}_a$ separately model the different deterioration procedure, such as optical blur, sampling error etc. Noise $\mathbf{n}$ is assumed as additive. This spatial domain representation is able to incorporate non-global motion, as well as spatially varying deformation. Additional constraints are needed for solving this ill-conditioned inverse problem. Different *a-priories* have been proposed [3, 4, 65, 76, 77]. Borman et al. [3] and Capel et al. [77] proposed to solve it in a Bayesian framework, with

MAP estimators using Huber edge-penalty function [78] as the *a-priori*. Uniform additive Gaussian noise model is assumed. In [4], Zomet et al. proposed to use a robust regularization as an additional constraint. It is a variant of the back-projection method. The median vector was used to update the gradient vector for more robustness. The performance of such algorithms relies on the consistency between the apriority assumption and the data. In [76], Farsiu et al. proposed an alternate approach, which uses $L-1$ norm minimization and robust regularization based on a bilateral prior, to accommodate different types of data and noise models. These forward algorithms are batch algorithms, which process the entire image simultaneously, which requires expensive large matrix operations and the complexity is in the order of the square of the image size. A large memory buffer is also required for such computations.

Recently, researchers, mostly from the computer vision society, have extended the super-resolution concept to reconstruct high-resolution image from single image input. Different from interpolation using different kernels, such as replication, bilinear, bicubic and other edge-preserving kernels [79, 80], these algorithms introduce additional prior knowledge learned from other available images with similar structure. The prior knowledge adds more compatible details so as to reduce the perceived loss [2, 67, 68, 69] as well as to facilitate the image representation ability. For example, in [70, 71, 72, 73, 74], face super-resolution is used to improve the recognition ability. In general, these generative algorithms can do a better job than deconvolution or interpolation. In [67], Freeman et al. proposed to use Markov Random Field (MRF) to model local image structures; and the idea of high-frequency prediction for super-resolution reconstruction was proposed. In [2], Baker and Kanade proposed to solve the super-resolution problem from the view of recognition. Pixel-wise matching is used to add more details. In [81], a combination of these two was presented for improvement. In [82], the spatial and temporal constraints are used to regularize the learned priors for a better reconstruction.

In this work we present a novel super-resolution framework. We adopt the similar idea of high-frequency loss compensation for multi-frame based super-resolution reconstruction. The rationale is based on the observation that low-resolution deterioration comes from the inter-pixel interference between neighboring points, due to the low-pass filtering nature of the image acquisition device as well as the sampling process. For each pixel, the interference from its neighboring pixels is estimated and subtracted, so that a refined estimate of the scene can be obtained. Experimental evaluation shows that the algorithm is effective when the magnification factor is low.

In case of large magnification factor is desired, learning-based super-resolution reconstruction algorithms utilizing the structure of face images should be considered. The problem of face super-resolution has certain uniqueness. Different from the other super-resolution problems, face images have a nice statistical structure. Many learning based face super-resolution algorithms have exploited such structural similarities [23, 65, 70, 71, 72, 73, 74, 82]. Essentially, these learning algorithms are used to find out the most appropriate model to describe the statistics between the high-resolution image and its low-resolution counterpart. Local patch based descriptor [23, 70, 73, 74, 82] as well as global template-type model are used [65, 71, 72]. Although local patch based models have their advantages, they will be problematic when occlusion occurs. In [65], Capel etc. proposed to use the PCA subspace as the feature representation and MAP estimation framework is incorporated to explore the underlying statistical structure of the face images. In [71], the idea is extended to use tensor face representation [83] to explicitly accommodate the multiple modes of variations in facial shape, expression, pose and illumination. However, as pointed in [84], the multi-linear tensor face representation treats each image as a single feature vector, by which the spatial localization ability is lost. Also, the high-dimensionality of the tensor space makes the algorithm still suffers from the *curse of dimensionality dilemma*. In this work, we propose to use tensorPCA subspace as the face representation, which is a specific case of the Concurrent Subspace Analysis described in [85]. The relevance model between the projected tensors from the low-resolution images and their counterpart from the high-resolution images is studied. A regression model is proposed based on a Maximum-likelihood estimation framework. This is still based on a global face representation, which makes the algorithm capable for partially occluded images.

Chapter III

Probability Learning in Automatic Eye Detection with A Binary Tree Representation

III.A Algorithm Overview

Object detection can be viewed as a binary classification problem that exhaustively classifies all image patches as either object or background. The image patches with sufficient high probability of being object indicate the presence of the object at the current locations. Each image patch can be represented by either pixels' intensity values or the transformed domain features. We use $\mathbf{X}$ to denote the feature vector for an image patch. The feature vector takes value $\vec{x}$. Knowing the probabilities $\Pr(\text{object}|\mathbf{X} = \vec{x})$ and $\Pr(\text{background}|\mathbf{X} = \vec{x})$, the optimal class label of $\mathbf{X}$, denoted as $\mathcal{L}(\mathbf{X})$, is determined by Bayesian criteria using the following classification rule:

$$\mathcal{L}(\mathbf{X}) = \begin{cases} \text{object} & \text{if } \frac{\Pr(\text{object}|\mathbf{X}=\vec{x})}{\Pr(\text{background}|\mathbf{X}=\vec{x})} > 1; \\ \text{background} & \text{otherwise.} \end{cases} \tag{III.1}$$

According to the *curse of dimensionality* coined by Bellman [86], the complexity of estimating PDFs grows exponentially with the increase of the dimensionality of the space. This rapid growth in complexity outpaces the computational and memory storage capabilities of computers. Furthermore, if the density function needs to be estimated for a given set of high-dimensional samples, the number of samples required for accurate density function estimation also grows exponentially. With a fixed number of training samples, the capability to accurately estimate the density is limited. Therefore, it is essential to reduce the dimensionality for accurate PDF estimation. There are many research efforts to address this issue, such as subspace projections like PCA [87, 88], LDA [88] and ICA [89]. However, most objects are constituted by variant substructures which have unique appearances and statistical structures. For example, human faces are composed of variant parts such as eyes, nose, lip and cheek, etc. These substructures may appear to have different distributions, hence it is unclear whether a single subspace transformation is sufficient to describe the object and extract the discriminant information. In this work, we focus on studying the feature space structure, such that the original feature space can be partitioned into meaningful low-dimensional subspaces, or parts, for more accurate probability density estimation.

Given feature sets $\mathcal{X}_1 = \{X_{1_1}, \cdots, X_{1_n}\}$ and $\mathcal{X}_2 = \{X_{2_1}, \cdots, X_{2_n}\}$, we can obtain two feature vectors $\mathbf{X}_1 = (X_{1_1}, \cdots, X_{1_n})$ and $\mathbf{X}_2 = (X_{2_1}, \cdots, X_{2_n})$. If $\mathbf{X}_1$ and $\mathbf{X}_2$ are independent, the joint PDF $\Pr(\mathbf{X}_1, \mathbf{X}_2)$ can be obtained by computing two marginal PDFs $\Pr(\mathbf{X}_1)$ and $\Pr(\mathbf{X}_2)$ separately, which is:

$$\Pr(\mathbf{X}_1, \mathbf{X}_2) = \Pr(\mathbf{X}_1)\Pr(\mathbf{X}_2). \tag{III.2}$$

According to previous discussion, to avoid the curse of dimensionality, it is always preferable to model the probability in a lower-dimensional space spanned by the feature vectors, especially when the sample size is limited. Hence, computing $\Pr(\mathbf{X}_1)$ and $\Pr(\mathbf{X}_2)$ separately is preferred to estimating $\Pr(\mathbf{X}_1, \mathbf{X}_2)$ directly. This motivates us to find the partition that the obtained feature cliques are as independent as possible. The feature dependencies are used to find such a partition.

Finding a partition of the feature space that best describes the feature space structure is equivalent to learn the statistical parts. A tree model is proposed for this purpose. Specifically, a binary tree is used. This tree-type model extracts statistical parts of the object from coarse to fine. Each subtree represents a statistical part of the object at a given scale. The leaf nodes model the finest detail we study. Also, the tree structure shows the connection between different parts. We use a clustering method based on the dependency distance, which is evaluated from features pairwise mutual information, to help partition the feature space. For each set of features, cliques with the least inter-dependency are found. The tree grows by separating the most independent feature cliques into different subtrees. The procedure is repeated, until the obtained feature subspaces are compact enough, which means every pair of features in the obtained subspace are highly dependent. Consequently, as the tree grows, the dependency between features from the same subspace increases. At any level, the PDF of the entire feature space can be recursively estimated from feature cliques in its subtrees, whose dimensionality is much lower. For each obtained low dimensional feature clique, its PDF is modeled by Gaussian mixtures of the independent components obtained from independent component analysis (ICA) [89]. The tree is built in a *top-down* fashion and the feature distribution is learned *bottom-up*. Figure III.1 gives an diagram of the object representation and posterior estimation. The DCT transformation is used to get image features that are robust to the global illumination variations. We use a window size of 8 pixels to compute the DCT coefficients and each block is normalized using the corresponding DC component. While other features can surely be used, in this work, *features* refers to the normalized DCT coefficients unless otherwise specified.

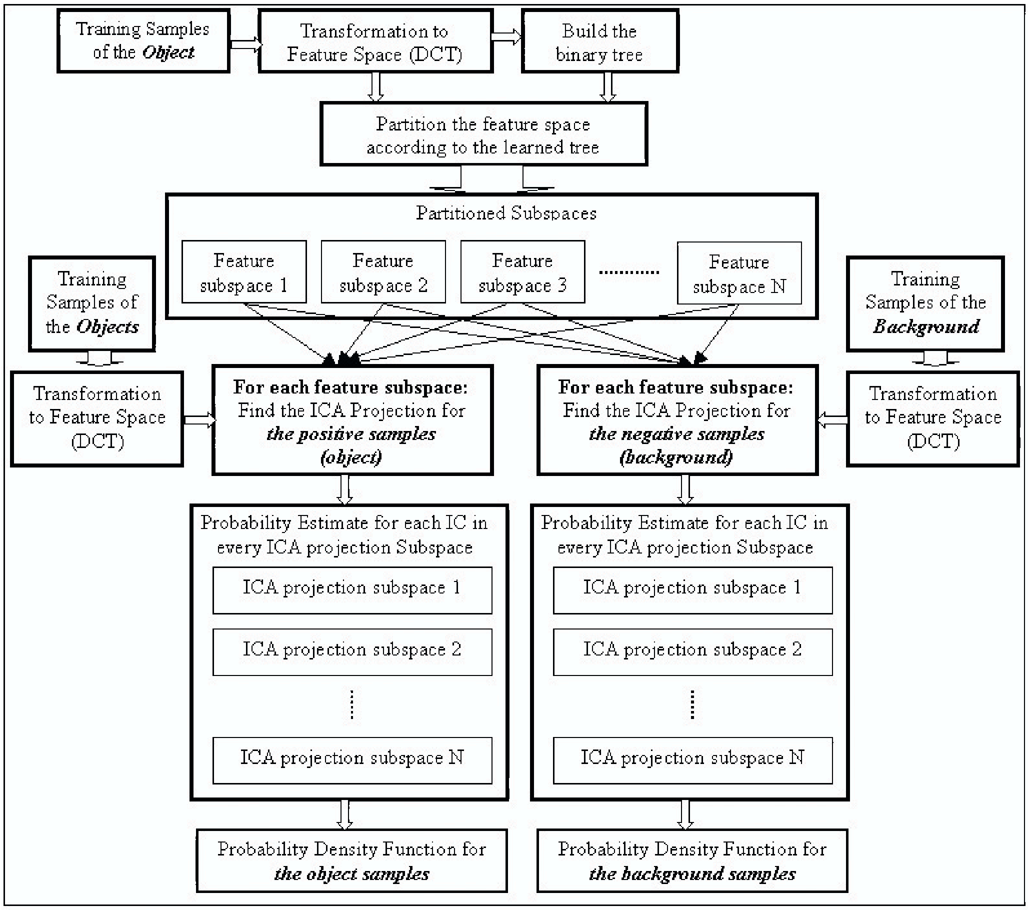

Training Samples of the *Object*
Transformation to Feature Space (DCT)
Build the binary tree
Partition the feature space according to the learned tree
Partitioned Subspaces
Feature subspace 1
Feature subspace 2
Feature subspace 3
....
Feature subspace N
Training Samples of the *Objects*
Training Samples of the *Background*
Transformation to Feature Space (DCT)
For each feature subspace: Find the ICA Projection for ***the positive samples (object)***
For each feature subspace: Find the ICA Projection for ***the negative samples (background)***
Transformation to Feature Space (DCT)
Probability Estimate for each IC in every ICA projection Subspace
ICA projection subspace 1
ICA projection subspace 2
ICA projection subspace N
Probability Estimate for each IC in every ICA projection Subspace
ICA projection subspace 1
ICA projection subspace 2
ICA projection subspace N
Probability Density Function for ***the object samples***
Probability Density Function for ***the background samples***

Figure III.1: Diagram of the object detection framework.

III.B Finding the Least Inter-dependent Feature Cliques

Each feature clique formed by closely correlated features can be explained as a statistical part. Features from the same statistical part have higher dependency; while features from different statistical parts are less correlated. We focus on learning the object's statistical structure that can represent an object by connected statistical parts from different scales.

Mutual information is a metric for evaluating the co-information between random variables. It measures the amount of information one random variable contains about the other, or mutual dependency. We use mutual information estimated from empirical distributions to evaluate the feature dependency. If two features are completely independent, we cannot derive any information about one feature from the other. In such case there is no co-information between these two features and the mutual information reaches the minimum, which is 0. Mutual information is defined in terms of entropies [90]:

$$I(X_1; X_2) = H(X_1) + H(X_2) - H(X_1, X_2). \tag{III.3}$$

$H(X_i)$ is the entropy of two features $X_i, i = 1, 2$, and $H(X_1, X_2)$ is the joint entropy. They are defined as follows [90]:

$$H(X_i) = -\sum \Pr(X_i) \log \Pr(X_i); \tag{III.4}$$

$$H(X_1, X_2) = -\sum \sum \Pr(X_1, X_2) \log \Pr(X_1, X_2). \tag{III.5}$$

Mutual information is actually the KL divergence [90] between the joint probability mass function $\Pr(X_1, X_2)$ and the product of the marginal probability mass function $\Pr(X_1)$ and $\Pr(X_2)$:

$$I(X_1; X_2) = \sum_{X_1} \sum_{X_2} \Pr(X_1, X_2) \log \frac{\Pr(X_1, X_2)}{\Pr(X_1)\Pr(X_2)}. \tag{III.6}$$

Its maximal value is attained if there exists a bijective mapping between the values of X_1 and X_2. On the other extreme, $I(X_1; X_2)$ drops to zero if X_1 and X_2 are statistically independent. Knowing the empirical distribution of the features, the pairwise mutual information can be estimated individually.

In order to evaluate whether the features in a feature clique have sufficient dependency, we define a *compact feature space* as follows:

$\diamond$ *Compact feature space*: Given a feature space $\mathcal{X} = \{X_1, \cdots, X_N\}$, where X_i are component features. Subset $\mathcal{X}' = (X_{s_1}, \cdots, X_{s_M})$ forms a compact feature space (with respect to θ) if for any $1 \leq i, j \leq M (i \neq j)$, the mutual information between components X_{s_i} and X_{s_j} satisfies:

$$I(X_{s_i}; X_{s_j}) > \theta;$$

where θ is a predefined threshold.

By working towards obtaining compact feature spaces, the pursued partition should tend to separate the independent features into different feature subspaces while preserving more correlated features in the same subspace. This definition contains ambiguity since subsets of a *compact feature space* also satisfy the definition. This may cause over-partitioning of the feature space. However, as we will see later, this relaxed definition only considers the within-set information, which simplifies the problem of identifying the compact feature space to a large extent. Also, over-partitioning can be controlled by an additional constraint: once a feature subset is identified as a compact feature subset, no further partition will be done. For such partition purpose, we define the following two variables:

Between-set Mutual Information

With the mutual information estimated from empirical distributions, if two feature subsets $\mathcal{X}_1$ and $\mathcal{X}_2$ are close to independent, we have:

$$I(X_1; X_2) \simeq 0; \quad \forall X_1 \in \mathcal{X}_1 \quad \text{and} \quad \forall X_2 \in \mathcal{X}_2.$$

$I(X_1; X_2)$ is the empirical pairwise mutual information between X_1 and X_2. Therefore the average pair-wise mutual information satisfies:

$$\mathcal{M}(\mathcal{X}_1; \mathcal{X}_2) = \frac{1}{|\mathcal{X}_1||\mathcal{X}_2|} \sum_{X_i \in \mathcal{X}_1} \sum_{X_j \in \mathcal{X}_2} I(X_i; X_j) \simeq 0; \tag{III.7}$$

where $| \bullet |$ is the number of features in the set. $\mathcal{M}(\mathcal{X}_1; \mathcal{X}_2)$ is called "between-set mutual information". For a given partition of the feature set, if we can find the subsets that have the between-set mutual information lower than a given threshold, these two subsets can be regarded as conditionally independent.

Within-set Mutual Information

Since the definition of the compact feature space only considers within-set information, the *compactness* of a feature subset can be easily examined to evaluate whether the features constitute a compact feature space. In other words, compactness determines whether the feature subspace needs further decomposition. We use "within-set mutual information" to evaluate the compactness, $\mathcal{C}(\mathcal{X}_1)$, which is defined as the average pairwise mutual information between features from the same subset.

$$\mathcal{C}(\mathcal{X}_1) = \frac{1}{|\mathcal{X}_1|(|\mathcal{X}_1| - 1)} \sum_{X_i, X_j \in \mathcal{X}_1; X_i \neq X_j} I(X_i; X_j). \tag{III.8}$$

With the features' pairwise mutual information, between-set mutual information and within-set mutual information, we use a clustering scheme adapted from traditional K-means clustering to find the feature cliques with the least inter-dependency. Instead of using Euclidean distance, we use mutual information to measure the dependency distance. Two features are "closer" if they have higher dependency. In other words, highly dependent features have smaller dependency distances. Since the mutual information increases with feature dependency, we use a monotonically decreasing function of the mutual information as the dependency distance measure:

$$D(X_i; X_j) = \frac{1}{1 + I(X_i; X_j)}. \tag{III.9}$$

Other monotonically decreasing functions can also be used.

Traditional K-means clustering groups the continuous data points into clusters, whose centroids are used as the prototypes for each cluster. In our case each data point is an image feature and the data space under investigation is discrete. This requires that the prototypes of each cluster be redefined. We use the feature that has the smallest average dependency distance to all the other features from the same cluster as the prototype. The stopping criterion is modified accordingly so that the clustering procedure stops when no feature changes cluster label. This clustering procedure guarantees the decreasing nature of the between-set mutual information, such that less dependent features are separated into different clusters.

The same as in K-means clustering, the number of the clusters, K, should be determined. K is selected in the clustering process by iteratively increasing K until there exists at least one set $\mathcal{X}_1$, such that:

$$C(\mathcal{X}_1) > \delta_S.$$

δ_S is a linear function of θ (the threshold to determine the "compact feature set"). Larger δ_S generates more parts while smaller δ_S leads to fewer partitions. We determine δ_S according to the prior knowledge about the subject. For eye subjects, we use the mean of the first 500 greatest features' (out of 2048 DCT coefficients for a 32×64 image patch) pairwise mutual information as δ_S. Experiments show that the algorithm is relatively stable with respect to δ_S. When we change δ_S from the mean of the first 500 greatest features' pairwise mutual information to the mean of the first 1000 greatest features' pairwise mutual information, output from the following clustering process does not have significant change.

The clustering procedure can be summarized in Algorithm 1. δ_W is a threshold determined from the prior knowledge of the subject in a similar way to δ_S. This procedure gives a way to find the feature cliques with the least inter-dependency. Suppose the current feature set can be partitioned into clusters $\mathcal{X}_1, \mathcal{X}_2, \cdots, \mathcal{X}_K$. Without loss of generality, we denote the two clusters that have the minimal between-set mutual information as $\mathcal{X}_1$ and $\mathcal{X}_2$.

To further avoid the over-partitioning problem mentioned above, we use an additional merging step. Suppose the size of $\mathcal{X}_1$ is greater than the size of $\mathcal{X}_2$. Let $A = \mathcal{X}_1$. If there are multiple subsets whose between-set mutual information

Algorithm 1 Clustering with dependency distance.

```
Step 1: Evaluate the compactness of the current feature set $\mathcal{X}$, $\mathcal{C}(\mathcal{X})$. If
            $\mathcal{C}(\mathcal{X}) > \delta_S$,
    no further partitioning is needed for $\mathcal{X}$ and the procedure stops. The current feature space is compact enough. Otherwise continue to the following steps.
Step 2: Estimate the pairwise mutual information for features in $\mathcal{X}$ by their empirical distribution, using Eq. III.6. Calculate feature dependency distance defined by Eq. III.9.
Step 3: For $k = 3, \cdots, K$, perform clustering by the following steps:
    1. Randomly select $k$ points $\hat{X}_l \quad (l = 1, \cdots, k)$ from $\mathcal{X}$ as the prototypes for $k$ clusters;
    2. Initialize error $\varepsilon$ with a large number;
    3. While $\varepsilon > 0$:
        (a) Assign each point $X_i$ into cluster $\zeta_i$ by evaluating $X_i$'s dependency distance to the cluster prototypes $\hat{X}_l$, $D(X_i; \hat{X}_l)$:
                $\zeta_i = \arg\min_l D(X_i; \hat{X}_l)$.    (III.10)
            $k$ clusters $\mathcal{X}_l$ $(l = 1, \cdots, k)$ can be obtained;
        (b) Let $\varepsilon$ equal to the number of features that change cluster labels;
        (c) For each cluster, update the prototype as follows:
                $\hat{X}_l \leftarrow \arg\min_{\hat{X}_l \in \mathcal{X}_l} \frac{1}{|\mathcal{X}_l|} \sum_{X_i \in \mathcal{X}_l} D(X_i; \hat{X}_l)$;    (III.11)
    4. Compute the compactness for each feature cluster $\mathcal{C}(\mathcal{X}_l)$, $l = 1, \cdots, k$.
        (a) If $\forall \mathcal{C}(\mathcal{X}_l) < \delta_S$, let $k = k + 1$ and repeat from step 1;
        (b) If for all clusters $\mathcal{X}_l$, we have $\mathcal{C}(\mathcal{X}_l) > \delta_S$;, proceed to step 2;
        (c) Otherwise we only concentrate on feature subsets that require further partition: for each cluster with $\mathcal{C}(\mathcal{X}_l) < \delta_S$, let $\mathcal{X} = \mathcal{X}_l$ and repeat from step 1.
```

Algorithm 2 Continue of the Algorithm 1.

```
**Step 4:** Calculate the between-set mutual information for each pair of clusters;
**Step 5:** Find the two clusters with the minimal between-set mutual information $\mathcal{M}_0 = \min_{i,j} \mathcal{M}(\mathcal{X}_i, \mathcal{X}_j)$ and the maximal between-set mutual information $\mathcal{M}_1 = \max_{i,j} \mathcal{M}(\mathcal{X}_i, \mathcal{X}_j)$;
    1. If the maximal between-set mutual information $\mathcal{M}_1 < \delta_W$, where $\delta_W$ is a given threshold, this implies that any two clusters are sufficiently independent. It indicates that the substructures represented by each cluster are not correlated and each of them can be handled independently;
    2. Otherwise go to the tree construction step described in the next section.
```

with A is smaller than δ_W, which means $\mathcal{M}(A; \mathcal{X}_q) < \delta_W$, we merge these subsets and get

$$B = \bigcup_{\mathcal{M}(A;\mathcal{X}_q)<\delta_W} \mathcal{X}_q.$$

A and B are the pursued independent feature cliques. This partition is recursively done for each feature subset. The details for constructing the tree are introduced in the next section.

III.C Tree Construction and Likelihood Learning

The above process generates two feature cliques A and B whose interdependency is sufficiently low. However, both subsets are still highly correlated with the remaining subsets. This means A and B can be considered as conditionally independent with respect to the union of the remaining subsets $C = \mathcal{X} \setminus (A \cup B)$. Suppose $A = \{X_1, \cdots, X_n\}$, for simplicity, from now on we use $\Pr(A)$ to represent the probability density distribution of the feature vector $(X_1, \cdots, X_n)$. Similarly, we have $\Pr(B)$ and $\Pr(C)$. This gives the following equation:

$$\begin{aligned} \Pr(A, B, C) &= \Pr(A, B|C)\Pr(C) \\ &\approx \Pr(A|C)\Pr(B|C)\Pr(C); \end{aligned} \qquad \text{(III.12)}$$

where $\Pr(A|C)$ and $\Pr(B|C)$ can be written as:

$$\Pr(A|C) = \frac{\Pr(A, C)}{\Pr(C)}$$

and

$$\Pr(B|C) = \frac{\Pr(B, C)}{\Pr(C)}. \tag{III.13}$$

From Eq. III.12 and III.13, the joint probability for $\mathcal{X} = A \cup B \cup C$ is:

$$\begin{aligned} \Pr(\mathcal{X}) &\approx \Pr(A|C)\Pr(B|C)\Pr(C) \\ &= \frac{\Pr(A, C)\Pr(B, C)}{\Pr(C)}. \end{aligned} \tag{III.14}$$

The equation Eq.III.14 shows that, as desired, for each set of features, the probability can be computed in several low-dimensional subspaces instead of in the original feature space. Also, this naturally leads to a binary tree representation for describing the feature space's statistical structure. We keep C as the parent node. Subsets $C \cup A$ and $C \cup B$ are features passed to the subtrees for further tree construction. Each subtree is recursively divided in a similar manner by identifying the most independent feature cliques. The procedure is illustrated in Figure III.2. In Figure III.3, we use a color diagram to show the procedure of the tree construction for human eyes. Only the first two levels are shown for the left eye. For illustration purposes, this color diagram is based on the pixel intensity features instead of the DCT features we used. The tree structure describes the feature space's statistical structure and each subtree represents a statistical part at a certain scale. As we traverse deeper towards leaves of the tree, the statistical part representation reveals more local details.

Although object detection problems can be solved using binary classification techniques, it is different from the traditional binary classification in the sense that the negative samples are not well defined, especially for object detection in unconstrained scenes. That is to say, unlike positive samples, there is no such explicit representative structure for negative samples. One way to solve this problem is to evaluate how far the sample is deviated from the positive samples in

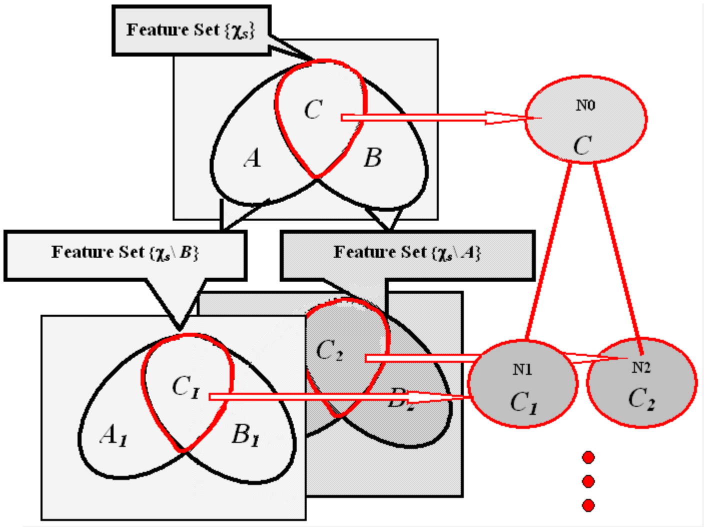


Figure III.2: Illustration for constructing the binary tree (first two levels). The first level feature set: $A \cup B \cup C$. The second level feature sets: $A_1 \cup B_1 \cup C_1$ and $A_2 \cup B_2 \cup C_2$ ($A_1 \cup B_1 \cup C_1 = \{\mathcal{X}\} \setminus B$ and $A_2 \cup B_2 \cup C_2 = \{\mathcal{X}\} \setminus A$).

partitioned subspaces. In each subspace, PDFs for positive samples and negative samples are estimated separately. Figure III.1 also illustrates this idea.

After the tree construction, each node of the tree is a clique constituted by highly correlated features. The relationship revealed by the tree, such as parent-child, siblings etc., describe how related two parts are. Each clique of features is a more compact representation for a certain local structure of the object whose probability can be estimated in a subspace with lower dimensionality. Given the same amount of training samples, probability estimation for each clique is easier and more accurate than that of the original feature space.

According to Eq. III.14, the joint probability for $A \cup C$, $B \cup C$ and C are needed for estimating the probability of $\mathcal{X}$. $A \cup C$ and $B \cup C$ are the subtrees following C, which can be estimated recursively using Eq. III.14 until reaching leaf nodes. This recursion gives the final equations to estimate the probabilities as

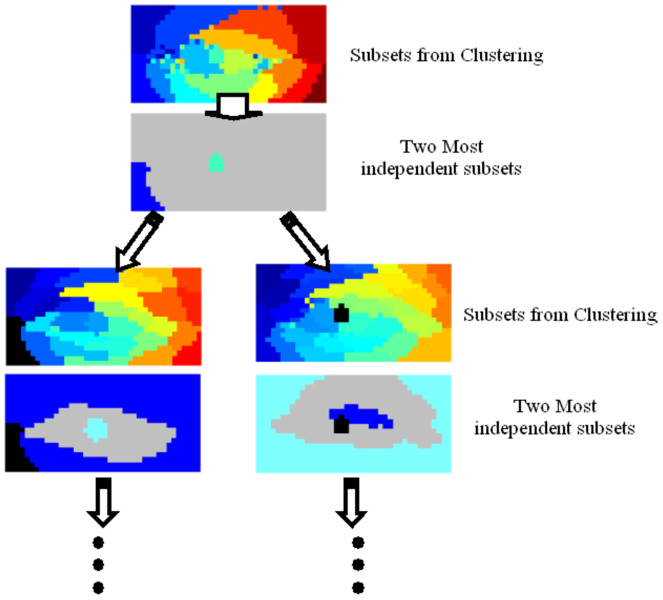


Figure III.3: Illustration of the tree construction (first two levels). In each node, the top image: the clustering results; the bottom image: two most independent cliques highlighted in blue and cyan and other features in gray.

follows (for both object class and background class):

$$\Pr(\mathcal{X}|\text{object}) = \frac{\prod_{\mathcal{X}_i \in \Omega_{leaf}} \Pr(\mathcal{X}_i|\text{object})}{\prod_{\mathcal{X}_i \in \bar{\Omega}_{leaf}} \Pr(\mathcal{X}_i|\text{object})}, \tag{III.15}$$

$$\Pr(\mathcal{X}|\text{background}) = \frac{\prod_{\mathcal{X}_i \in \Omega_{leaf}} \Pr(\mathcal{X}_i|\text{background})}{\prod_{\mathcal{X}_i \in \bar{\Omega}_{leaf}} \Pr(\mathcal{X}_i|\text{background})}, \tag{III.16}$$

where Ω_{leaf} is the set composed of leaf nodes and $\bar{\Omega}_{leaf}$ is the set composed of non-leaf nodes.

Eq. III.15 and III.16 indicate that only two kinds of joint probabilities need to be estimated:

1. Features' joint probability for the feature clique represented by leaf node;

2. Features' joint probability for the feature clique represented by non-leaf node.

In both types of feature cliques, the dimensionality of the feature space is significantly lower than the original un-partitioned space, such that the probability estimation is easier and more accurate.

III.D Probability Estimation in Subspaces

The tree representation partitions the original feature space into subspaces with lower dimensionality. In each subspace, the component features are highly correlated. Although we can use a non-parametric way to learn the distribution, the computation cost is proportional to the size of the training sample set. This is computationally expensive. Another widely used method for probability estimation is to use the Gaussian mixture as a parametric model. However, due to the high correlation between features, the covariance matrix is ill-conditioned, hence the direct use of the Gaussian mixture modeling is not applicable. The high correlation between features implies that there is redundancy between features. We propose to first use ICA for reducing the redundancy. By using ICA projection, we take the assumption that highly dependent features are the mixtures of a small number of underlying independent sources. According to [89], ICA actually finds the independent sources that models the higher-order (higher than the second order) statistics of the observation. Mutual information between the outputs is minimized so as to reduce the redundancy. After identifying the independent sources, the distribution of each source can be modeled individually using Gaussian mixtures. The *informax ICA* proposed in [89] is used for computing the ICA projection.

Let samples from the current feature subset be $\mathcal{X} = \{\mathbf{X}_i\}$, where $\mathbf{X}_i = [X_1^i, X_2^i, \cdots, X_N^i]^{\mathrm{T}}$. Suppose the N-dimensional samples are generated from s independent sources $\mathbf{y}_i = [y_1^i, \cdots, y_s^i]^{\mathrm{T}}$, where $s < N$, we have:

$$\mathbf{X}_i = \mathbf{W}\mathbf{y}_i. \tag{III.17}$$

$\mathbf{W}$ is the mixing matrix. The goal of ICA is to find a linear mapping $\mathbf{U}$ such that the un-mixed sources $\hat{\mathbf{y}}_i$ satisfies:

$$\hat{\mathbf{y}}_i = \mathbf{U}\mathbf{X}_i = \mathbf{U}\mathbf{W}\mathbf{y}_i.$$

Infomax is one way to find the un-mixing matrix $\mathbf{U}$. It maximizes the joint entropy of the outputs, whose learning rule is derived by using the maximum likelihood function. The signal mixing process gives:

$$\Pr(\mathbf{X}) = |\det \mathbf{U}| \prod \Pr(\mathbf{y}).$$

With the independence assumption of the sources, the likelihood can be written as:

$$L(\mathbf{y}, \mathbf{U}) = \log |\det \mathbf{U}| + \sum_k \log \Pr(y_k).$$

Infomax gives the learning algorithm as [89]:

$$\Delta \mathbf{U} \propto \frac{\partial L(\mathbf{y}, \mathbf{U})}{\partial \mathbf{U}} \mathbf{U}^{\mathrm{T}} \mathbf{U}. \tag{III.18}$$

After computing the un-mixing matrix, the underlying independent sources can be recovered. Each independent source is modeled individually with a mixture of Gaussians as:

$$y_k \sim \sum_j \omega_j^{(k)} \mathcal{N}(\mu_j^{(k)}, \sigma^{2}{}_j^{(k)}); \tag{III.19}$$

where $\mathcal{N}(\mu_j^{(k)}, \sigma^{2}{}_j^{(k)})$ is a Gaussian with mean $\mu_j^{(k)}$ and variance $\sigma^{2}{}_j^{(k)}$. The overall probability is:

$$\Pr(\mathbf{y}) = \prod_k \Pr(y_k). \tag{III.20}$$

We use Expectation-Maximization (EM) to estimate the mixture of Gaussians from training samples. Both positive samples and negative samples are modeled in the same way.

Considering the class label, object posterior probability and background posterior probability are learned individually using Eq. III.12∼Eq. III.19. For each feature clique $\mathcal{X}_i$, the posterior probability can be estimated by:

$$\Pr(\mathcal{X}_i|\text{object}) = |\det \mathbf{U}_p^{(i)}| \prod \Pr(\mathbf{y} = \mathbf{U}_p^{(i)} \vec{x}|\text{object}), \vec{x} \in \mathcal{X}_i$$

and

$$\Pr(\mathcal{X}_i|\text{background}) = |\det \mathbf{U}_n^{(i)}| \prod \Pr(\mathbf{y} = \mathbf{U}_n^{(i)}\vec{x}|\text{background}), \vec{x} \in \mathcal{X}_i.$$

$\mathbf{U}_p^{(i)}$ and $\mathbf{U}_n^{(i)}$ are the ICA un-mixing matrices in subspace $\mathcal{X}_i$ for positive samples and negative samples respectively. With these posteriors, Bayesian criterion gives:

$$\mathcal{L}(\mathbf{X}) = \begin{cases} \text{Object} & \text{if } \frac{\prod_{\mathcal{X}_i \in \Omega_{leaf}} \Pr(\mathcal{X}_i|\text{object}) \prod_{\mathcal{X}_i \in \overline{\Omega}_{leaf}} \Pr(\mathcal{X}_i|\text{background})}{\prod_{\mathcal{X}_i \in \overline{\Omega}_{leaf}} \Pr(\mathcal{X}_i|\text{object}) \prod_{\mathcal{X}_i \in \Omega_{leaf}} \Pr(\mathcal{X}_i|\text{background})} > \eta; \\ \text{Background} & \text{otherwise.} \end{cases} \tag{III.21}$$

The negative samples are far from sufficient to cover all possible background scenes, especially when the scenes are unconstrained. In other words, the probability estimate for background samples can only include the known background. Due to the lack of sufficient prior knowledge, for unknown backgrounds, the posteriors estimated for both classes could be very low. This implies that simply using the ratio $\frac{\Pr(\mathbf{X}=\vec{x}|\text{object})}{\Pr(\mathbf{X}=\vec{x}|\text{background})}$ as in the Bayesian criterion is not sufficient. Although this can be alleviated by training in a bootstrap way such that more problematic negative samples can be added into the training step by step, we also use the posterior of the positive samples for the classification decision. Now the decision criterion becomes:

$$\mathcal{L}(\mathbf{X}) = \begin{cases} \text{Object} & \text{if } \frac{\Pr(\mathbf{X}=\vec{x}|\text{object})}{\mathbf{Pr}(\mathbf{X}=\vec{x}|\text{background})} > \eta \quad \text{and} \quad \Pr(\mathbf{X} = \vec{x}|\text{object}) > \eta'; \\ \text{Background} & \text{otherwise.} \end{cases} \tag{III.22}$$

III.E Experimental Evaluation

In this section, we present a performance evaluation of the algorithm and discuss some implementation issues and challenges. We specifically apply the proposed algorithm for the task of human eye detection and localization. In our implementation, we use normalized 8 × 8 DCT features although other features may also be viable.

III.E.1 Determining the Threshold Parameters η and $\eta\prime$

The thresholds η and $\eta\prime$ controls the sensitivity of the detector. From the Bayesian criterion, the optimal value for η should be ratio of the priors. However, the priors are usually unknown for real-world data. We propose to determine the optimal η and $\eta\prime$ from experimental ROC curves. We use two sets of samples, one for training and one for testing. Each sample is an image patch. Figure III.4 gives some examples of the training and testing samples used for eye detection. The top rows in Figure III.4 show training examples and the bottom rows in Figure III.4 show testing examples. We have 2011 positive examples and 8019 negative examples for training; and another 450 positive examples and 7307 negative examples for testing. Eye samples in both the training samples and the testing samples are from the left eye. For a fair comparison, training samples and testing samples are from different databases: training samples are from the FERET database [91]; and testing samples are from the Caltech face database. The typical background in these two databases are different, which directly causes the apparent difference in negative samples.

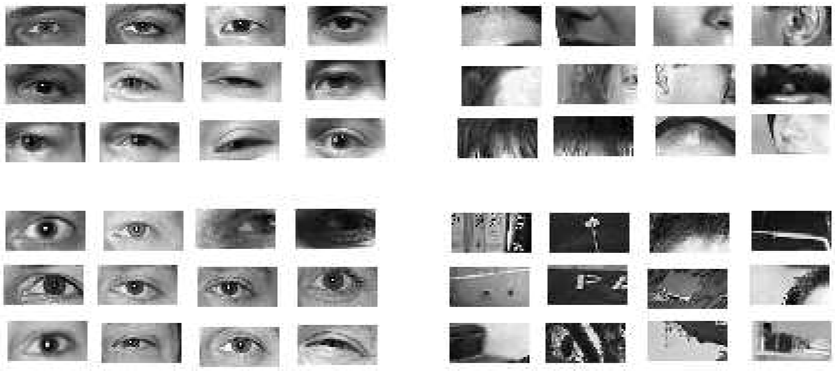


Figure III.4: Training and testing examples. The top rows are training eye samples and non-eye samples respectively. The bottom rows are the corresponding testing examples. The size of the samples is 32 × 64.

Using the test sets, we examine the ROC curves at different thresholds. With fixed η, each $\eta\prime$ gives one ROC curve. A set of ROC curves can be generated in such a way. Figure III.5 gives an example of the set of ROC curves. η is selected to yield a desired ROC characteristic. Here we choose the value that can achieve the highest detection rate with a certain false alarm rate (10% in our implementation). The ROC curve also helps us determine the range of $\eta\prime$. A similar approach is used to refine $\eta\prime$. With $\eta\prime$ fixed within the range determined previously and changing η, we choose the desired ROC curve similarly. Figures Figure III.5-Figure III.6 give the ROC curves for determining η and $\eta\prime$ respectively. According to the ROC curves, we set $\eta = 1$ and $\eta\prime = 3 \times 10^{-6}$.

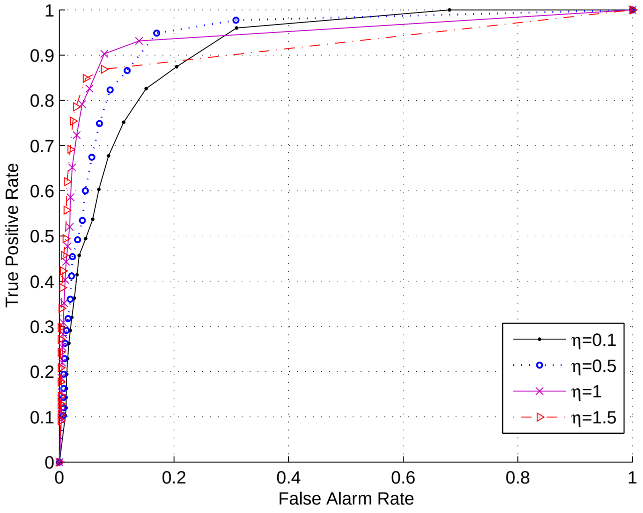


Figure III.5: ROC curves used to determine η and $\eta\prime$. Each ROC curve is obtained with a fixed η.

III.E.2 Comparison with Closely Related Approaches

After the threshold parameters are determined, we compare the proposed algorithm with three existing methods. Two of the three can be regarded as the intermediate steps of using the final tree model. In the first method we use Bayesian criteria directly. A simple Gaussian mixture model is used to learn the entire object's PDF. In the second method, we consider the parts as disjoint and independent, therefore, the object is partitioned into independent and non-overlapped subsets. This model differs from our final tree-structure model in the aspect that

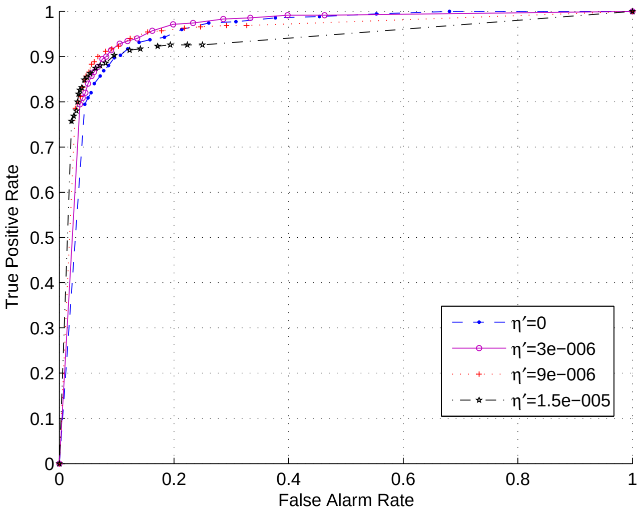


Figure III.6: ROC curves used to determine η and $\eta\prime$. Each ROC curve is obtained with a fixed $\eta\prime$.

all subspaces found in this way are regarded as disjoint and independent. If the partition result is $\mathcal{X}_1, \cdots, \mathcal{X}_s$, the probability is:

$$\Pr(\mathcal{X}_1, \cdots, \mathcal{X}_s|\text{object}) = \prod_{i=1}^{s} \Pr(\mathcal{X}_i|\text{object}); \tag{III.23}$$

$$\Pr(\mathcal{X}_1, \cdots, \mathcal{X}_s|\text{background}) = \prod_{i=1}^{s} \Pr(\mathcal{X}_i|\text{background}); \tag{III.24}$$

In [92], the authors also proposed using clustering for grouping feature vectors. Parent structure features are used, which describe the neighborhood details. Each cluster is constituted of similar feature vectors from the same distributions, hence the object's PDF is obtained by individually computing the feature vector's PDF. We also compare with this approach.

For all these methods, the performance is shown by ROC curves in Figure III.7. The same training and testing sets as described in previous section are used. The comparison shows that partitioning the feature space into independent feature subspaces has a great contribution to the performance. Without the partition step, the probability estimates hardly provide useful information for classification at all. Also, as expected, without considering the relationship between

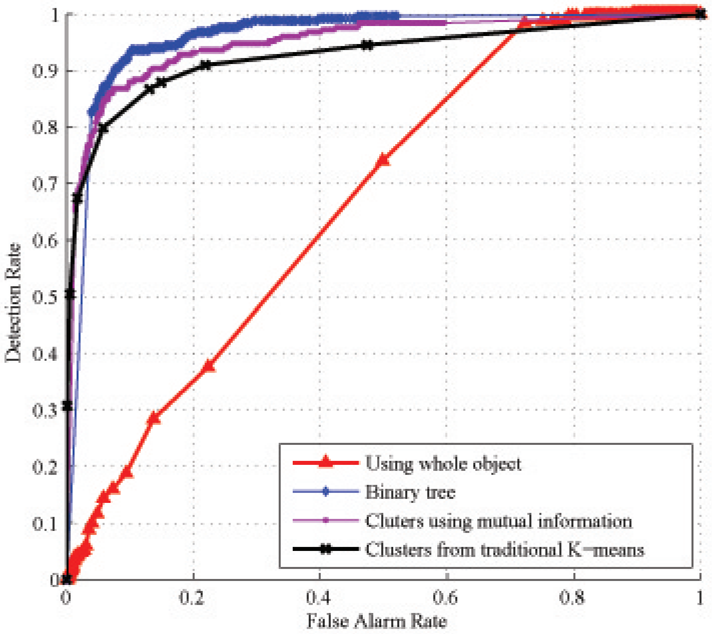


Figure III.7: ROC curves for the algorithm proposed compared with three closely related approaches. “Binary tree” is the proposed approach.

the subsets in varying scales, which means that the subsets (i.e., the parts) are considered as disjoint and independent, the performance is inferior to our proposed approach.

III.E.3 Evaluation Over Face Images

In this section, the performance of eye detection in face images is evaluated. The main challenge is the ability to differentiate the eyes from the other facial parts. We have 311 test face images from the GRAY FERET database [91]. The same training set as used in previous sections is exploited to learn the model and images used in training are not included in the testing data set. Images are normalized to a given size, such that the detection can be done at a fixed scale. At each pixel, its surrounding neighborhood is examined to determine whether it is an eye image. If it is an eye, the location is marked. An exhaustive search over the entire image is used. Non-maxima suppression can be used afterwards for a

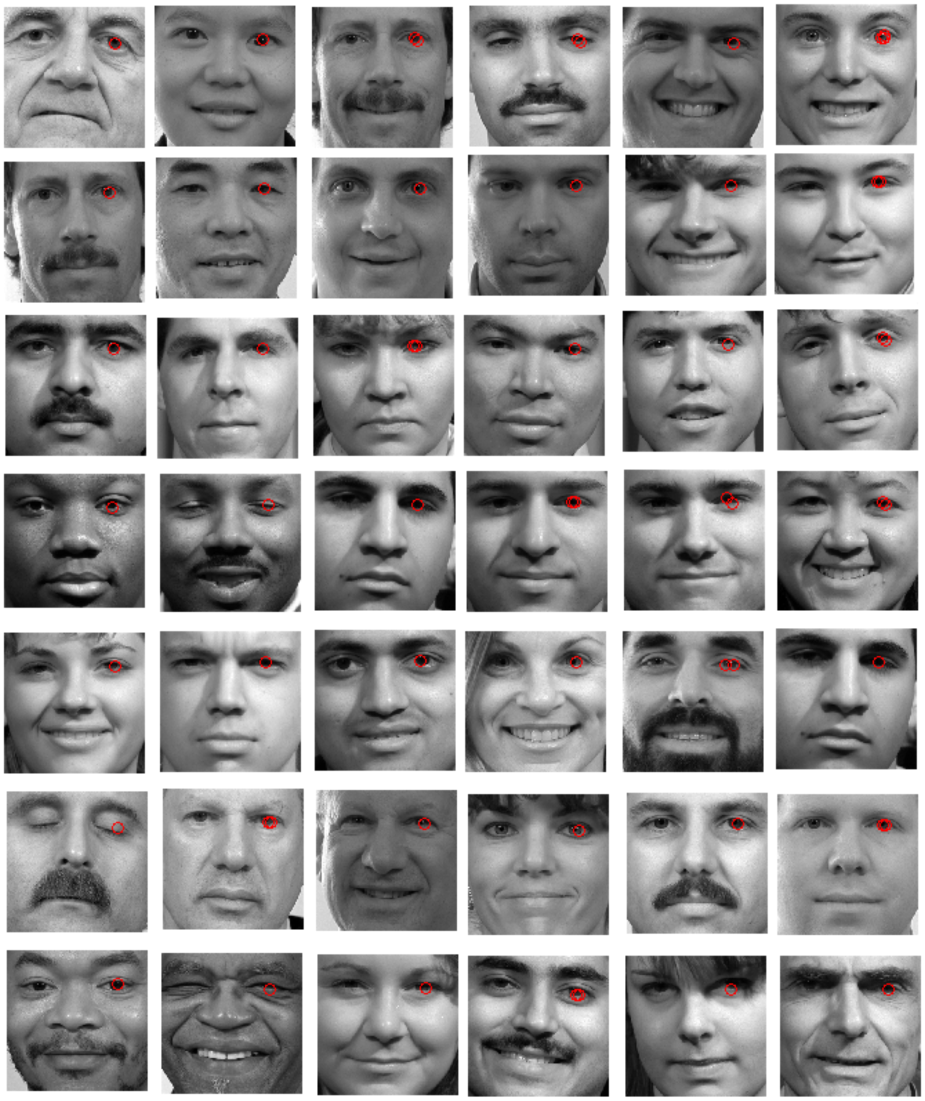


Figure III.8: Successful detection and localization of left eye. The detections are marked by red circles.

clean detection. Since the training sample set only includes the left eyes, we only consider the detection of the left eye as a correct detection. Figure III.8 gives some examples of accurate detection results. Figure III.9 gives some detection examples where the localization error is high. Figure III.10 gives bad detection results, including false alarms and mis-detections.

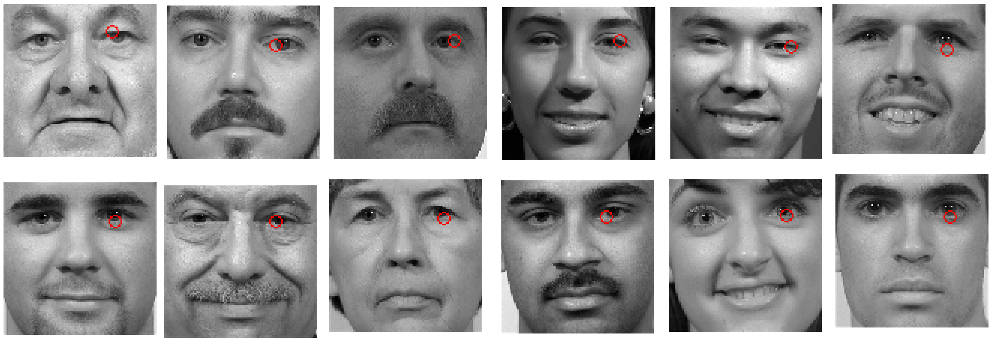


Figure III.9: Correct detections with large localization error. The detections are marked by red circles.

Table III.1: Summary of the eye detection results for the proposed algorithm. The testing images are from the FERET database. The accuracy rate is counted over all the detected results. Close detections are merged to remove multiple detections.

No. of the samples	Left eye detected	False detection	Mis-detection	Right eye detected
311	296	23	9	19
100%	92.43%	7.26%	2.95%	5.99%

We evaluate the performance quantitatively. The detection accuracy is 92.43% with a false alarm rate 7.26%. Detection accuracy is counted as ratio of the correct detections *v.s.* the number of the objects, and the false alarm rate is the ratio of the images containing false detections *v.s* the number of the images. It is worth noting that the proposed algorithm can learn the subtle difference between the left eye and right eye when we only use left eyes samples for training. This suggests that the difference between the left eyes and the right eyes should be considered in training. Our experimental results show that only 5.99% of the images give right eye detections, which is also included in Table Table III.1. Table Table III.1 summarizes the results.

III.E.4 Detection in Complex Background

We also evaluate the algorithm with a much harder set of images with complicated background, using images from the Face Recognition Grand Chal-

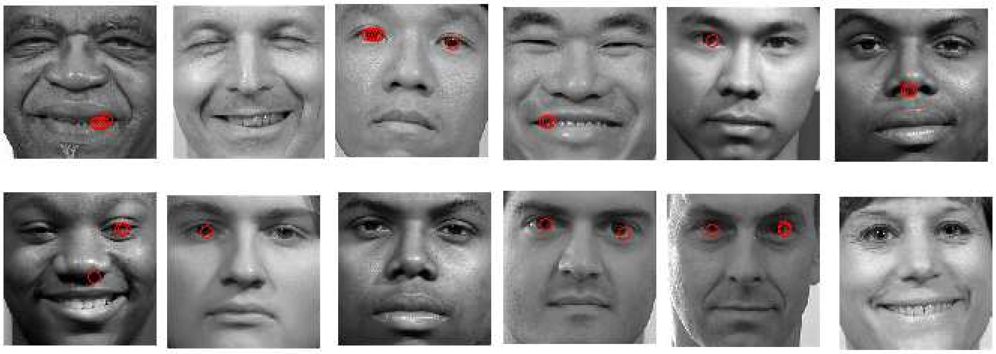


Figure III.10: False detections and mis-detections. The detections are marked by red circles.

lenge (FRGC) database. The database contains indoor and outdoor images with relatively large illumination variations. The database has been evaluated using a commercial system by Viisage, which utilizes a 3D face mesh model to locate the eyes. The Viisage system provides correct detection for most images. However, there is a subset of images where Viisage system generates low confidence. This subset is called *bad image* dataset. This subset was released by FRGC organizer for eye localization studies. Also, the results from the Viisage system are also provided by FRGC organizers for performance evaluation purposes. In this work, we evaluate the performance of the proposed algorithm over the *bad image* subset, and compare it to the provided Viisage results. In each image, there is only a single person with a frontal or near frontal view. Some example images are shown in Figure III.11.

To lower the computational cost, we use a skin-color face detection step to reject the most unlikely areas. In order to avoid mis-detection from this step, we use a very low threshold for rejection. The reason for using skin-color is also because of this adjustable parameter. For every image, each single remaining blob gives a face candidate, and each image may contain several face candidates. These face candidates are investigated individually for eye detection. The size of each face candidate region gives a rough initialization of the scale s, which is used to resize the eye patch to size 32×64. We also examined $0.8s$ and $1.2s$ to find the

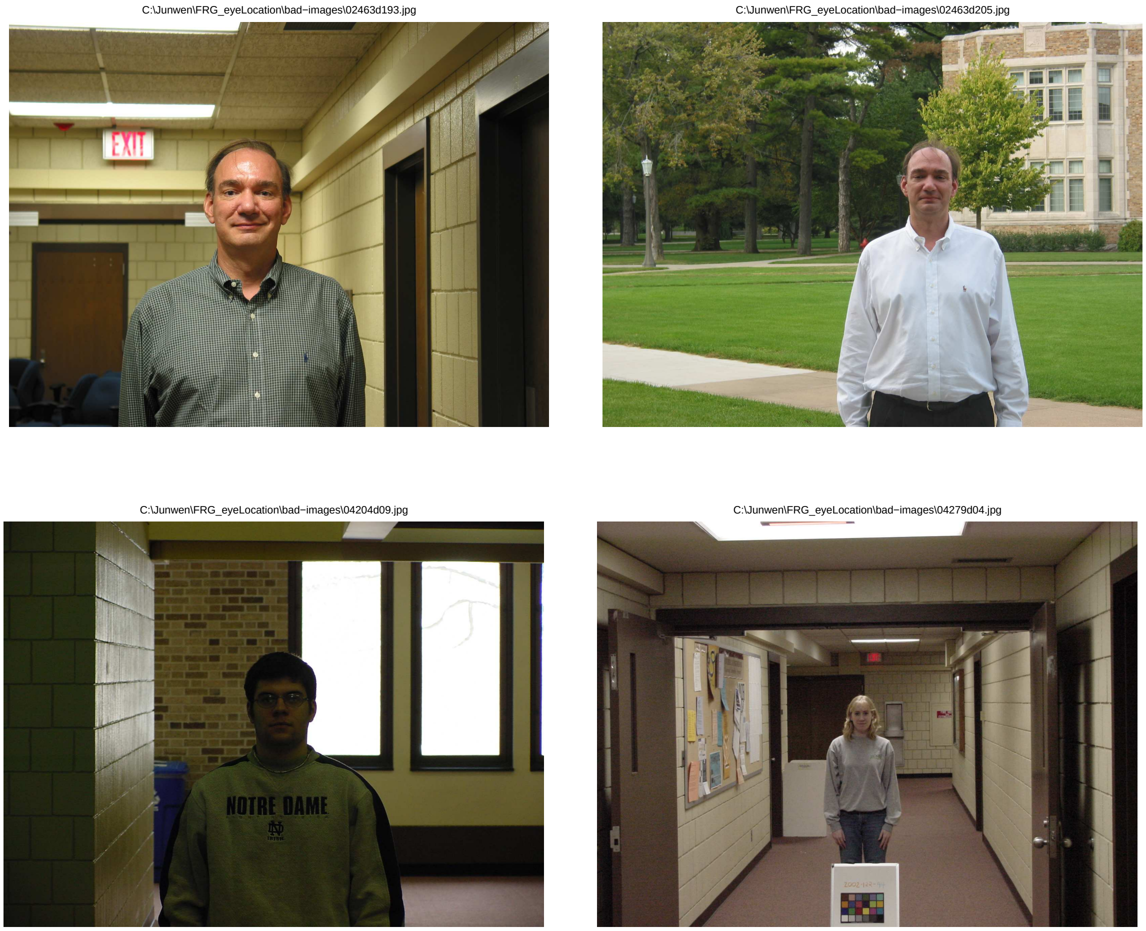

C:\Junwen\FRG_eyeLocation\bad-images\02463d193.jpg
EXIT
C:\Junwen\FRG_eyeLocation\bad-images\02463d205.jpg
C:\Junwen\FRG_eyeLocation\bad-images\04204d09.jpg
NOTRE DAME
C:\Junwen\FRG_eyeLocation\bad-images\04279d04.jpg

Figure III.11: Example images from the FRGC eye localization database. (Images are rescaled to 0.4 × 0.4 of the original size).

best eye candidate. The sub-windows slide every 2 pixels to further decrease the computational cost.

The left eye detector is flipped horizontally to obtain the right eye detector. The neighboring detections are merged by non-maxima suppression of the Bayesian ratio. Only the local maxima of the Bayesian ratio are kept. If more than 2 local maxima are present, we use a geometric constraint to remove false detections. Some examples of the raw detection are shown in Figure III.12. In Table Table III.2 we summarize the obtained results. The detected results are categorized into four classes: good detection for both eyes, one eye detected, false

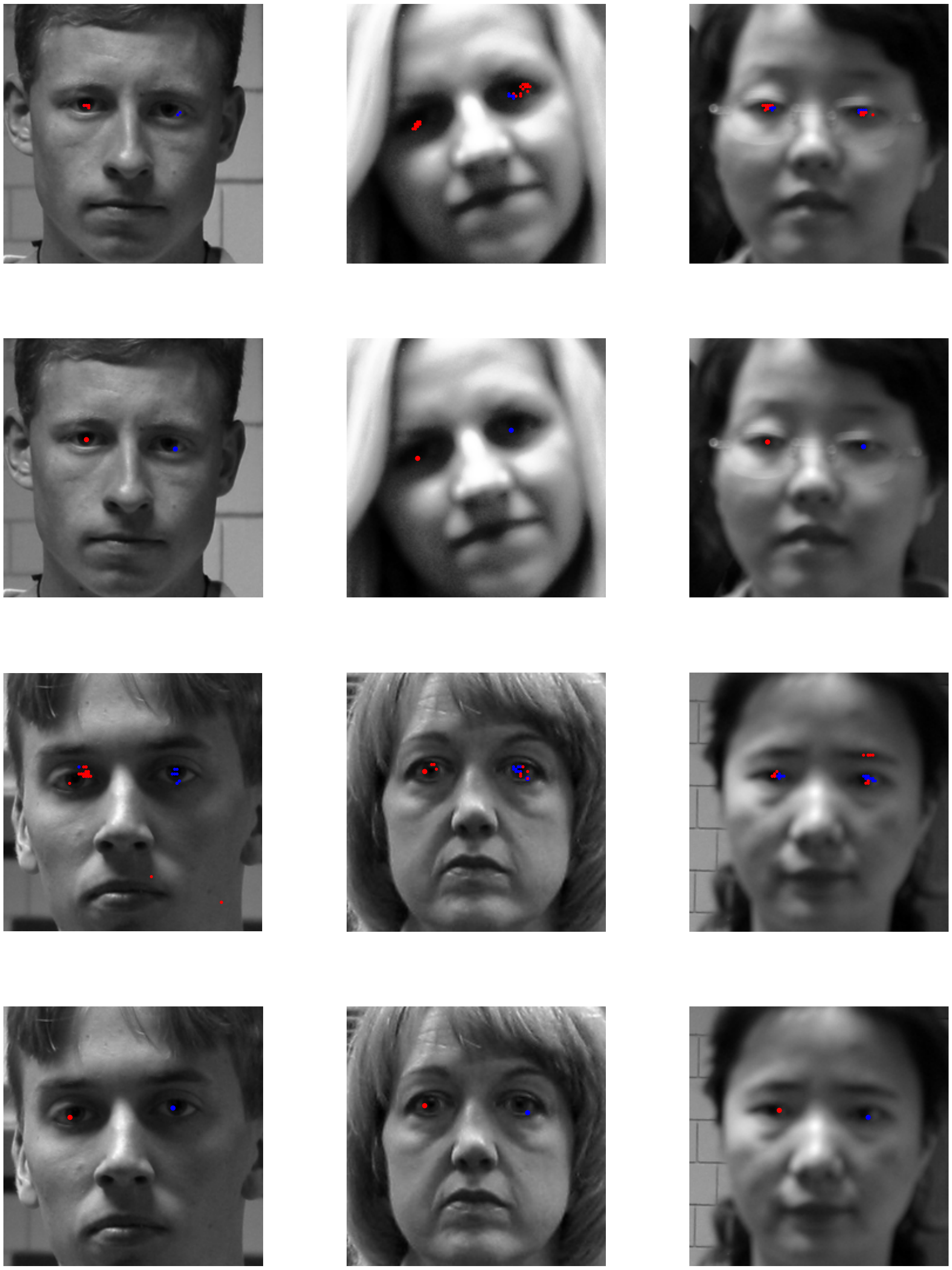


Figure III.12: Examples of detection results from the FRGC ''bad" image database. The first and the third rows: the raw detection; the second and the fourth rows: results after non-maxima suppression and geometric constraint.

detection and mis-detection. Figure III.13 illustrates how we categorize the detection results. The red point in Figure III.13 is the true location of the center of the

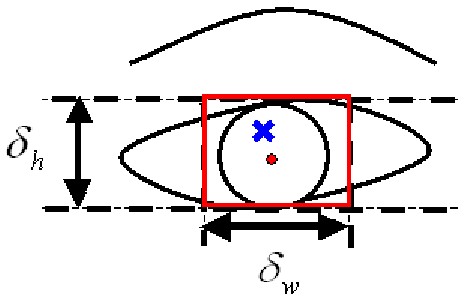


Figure III.13: Illustration of how we categorize the detection results.

Table III.2: Detection results for the "bad-image" data set from FRGC database: comparison between the results from the proposed Binary Tree (BT) algorithm and the results from the Viisage system.

Number	Good	One eye	Mis-detection	False
of the	detection	detected		detection
samples	BT / Viisage	BT	BT / Viisage	BT / Viisage
915	736 / 635	33	43 / 64	103 / 216
100%	80.44% / 69.40%	3.61%	4.70% / 6.99%	11.26 % / 23.61%

eye (marked by a human observer). The blue cross shows the detected result. We use a manually labeled data set as the ground-truth to evaluate the performance. To avoid the bias from the prior knowledge of the algorithm, a human subject with little computer vision background is asked to label the data. We have to keep in mind that the *ground-truth* is subjective, which introduces certain error. We evaluate the distance in the normalized images, which means the distance is evaluated on the eye image patch that has been re-scaled to 32 × 64. If the detection is located within the red rectangle defined by the normalized distance Δ_h and Δ_w, the detection is considered correct. If both eyes are correctly detected, the detection is a good detection. If one eye is correctly detected while the other one is not, this detection is categorized into the "detect-one-eye" category. In Table Table III.2, $\Delta_w = 32$ and $\Delta_h = 24$ pixels (evaluated on the scaled eye image patch) is used to evaluate a correct detection. The results from the Viisage system are also listed for comparison. More results are shown in Figure III.14, Figure III.15 and

Figure III.16; where each figure gives detections for one subject.

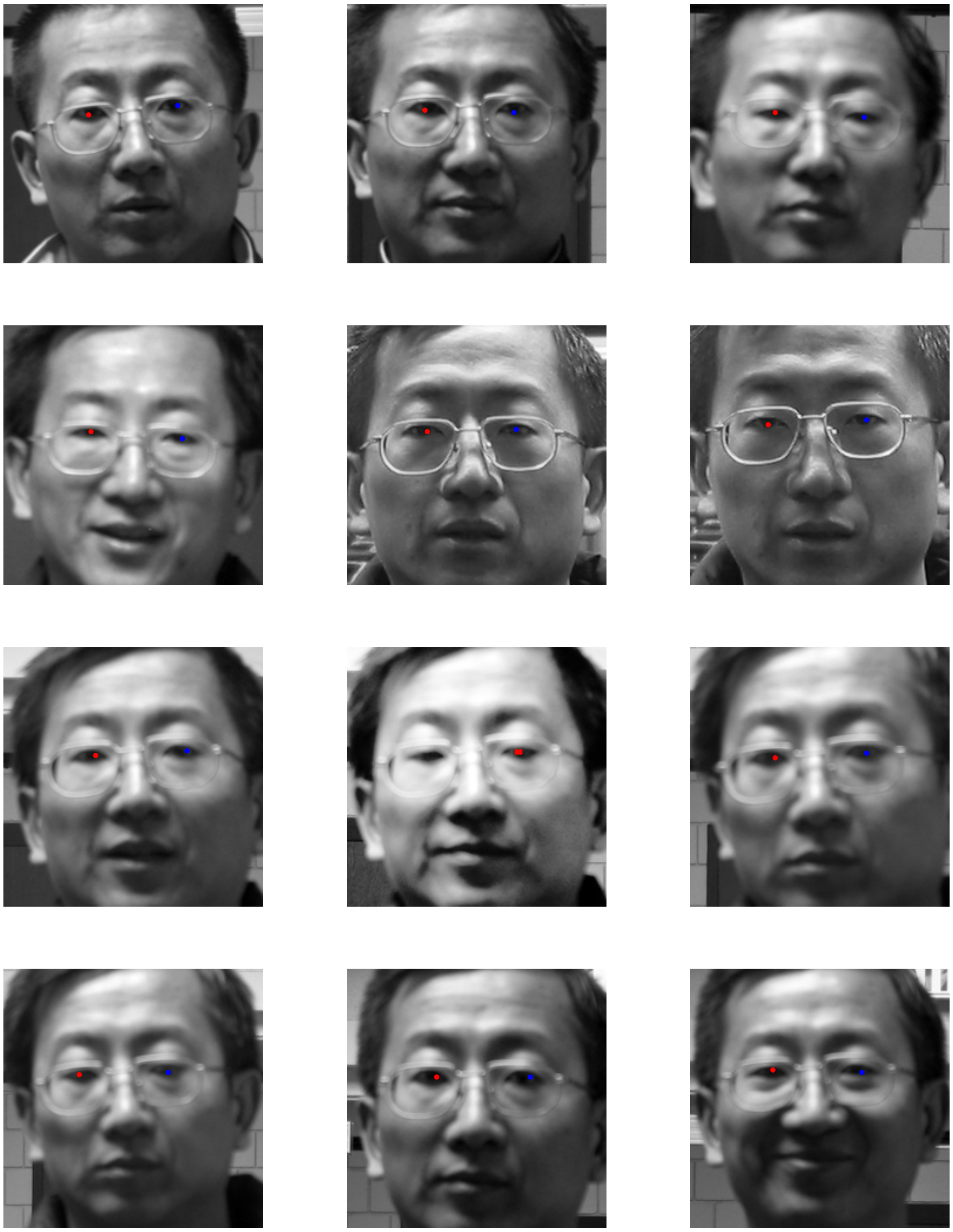


Figure III.14: More detection results for the same subject under varying imaging conditions. In order to show the location accuracy of the detection, only face areas are shown. Different color marks the output from the left and right eye models.

By using normalized DCT features (or other feature such as DC-free

Gabor wavelets), the algorithm exhibits certain ability to handle illumination variations. Figure III.15 and Figure III.16 give some typical examples with the raw detection overlayed on the original images, where each figure contains 6 images from the same subjects under certain illumination changes. It shows that the algorithm has the ability to handle illumination variations in a certain extent. The top-right image in Figure III.15 shows an example of a mis-detection. The artificial indoor illumination shifted the color distributions, such that the detected skin-color blob gives a wrong estimate of the initial scale. This in turn caused the mis-detection of eyes. It is similar for the middle-left image in Figure III.15, which is a false detection. Illumination changes in eye detection problem have been tackled by using IR illuminations and the *red-eye* effect, however, it requires additional hardware setup and handling extreme illumination conditions is still an open challenge.

The histogram of the localization error ϵ can be used to evaluate the localization ability. The localization error ϵ is the Euclidean distance defined in the normalized image. More specifically, if we normalize the eyes to size 32×64, and map both the manually obtained ground-truth $\mathbf{P}_0$ and the detected results $\mathbf{P}_d$ accordingly such that $\mathbf{P}_0 \rightarrow \hat{\mathbf{P}}_0$ and $\mathbf{P}_d \rightarrow \hat{\mathbf{P}}_d$, the localization error ϵ is computed as:

$$\epsilon = \|\hat{\mathbf{P}}_0 - \hat{\mathbf{P}}_d\|.$$

In Figure III.17(a), the histograms of the localization error are shown. Left eye and right eye are evaluated separately, which are shown in left and right respectively. For comparison, the corresponding histograms of the results from the Viisage system are also shown in Figure III.17(b). The numbers of false detection are also marked with a blue line for comparison. The results indicate that although the proposed algorithm has a higher detection rate, the localization ability is not better. However, since the detection is done in a sparser grid (sub-window slides every 2 pixels) for faster computation, this is a possible source of the localization error. Also, some more complex post processing other than non-maxima suppression,

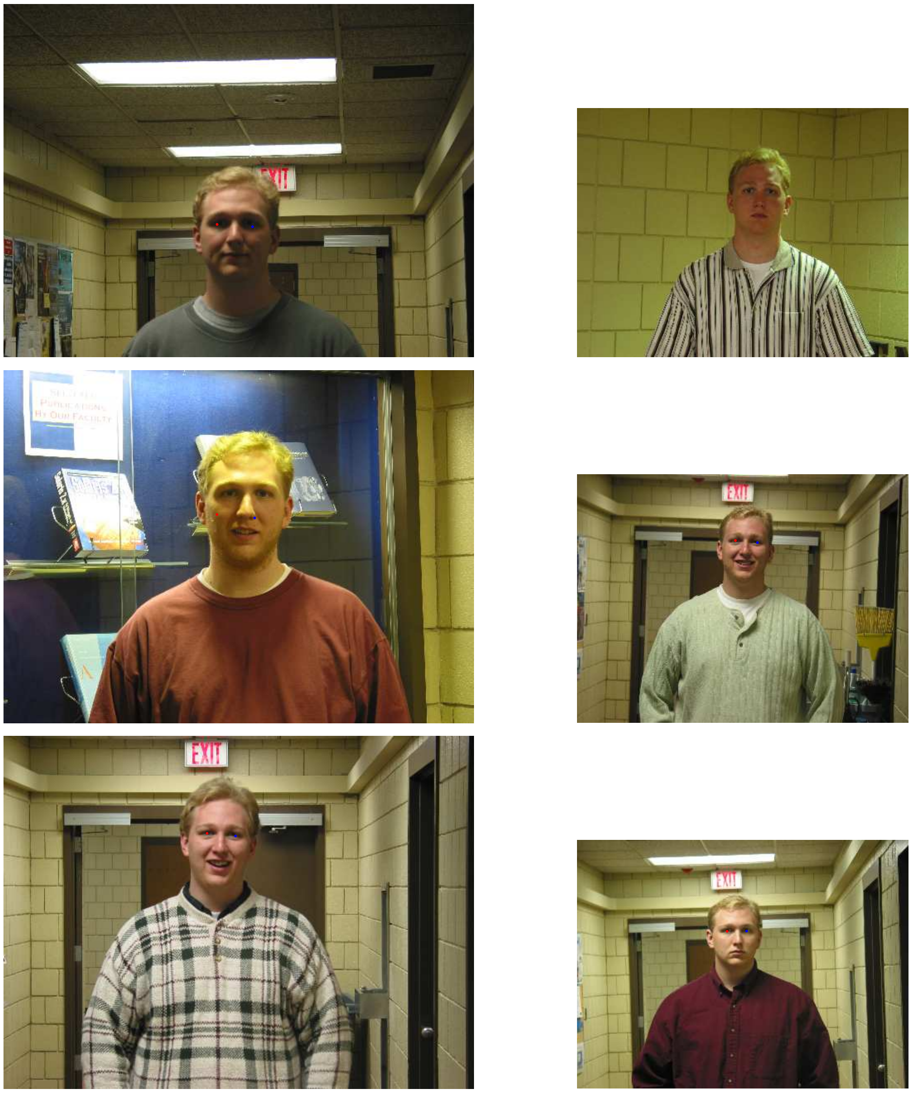


Figure III.15: Detection results for the same subject under certain illumination changes. Different color marks the output from the left and right eye models. Each image is subsamples to 0.125×0.125 of the original image size.

such as using the centroid of the connected detections, could be used to obtain a better performance.

The text of this chapter, in part, is a reprint of the material as it appears in: Junwen Wu and Mohan M. Trivedi, "A Binary Tree for Probability Learning in Eye Detection." in *Proceedings of the IEEE FRGC Workshop, in Conjunction*

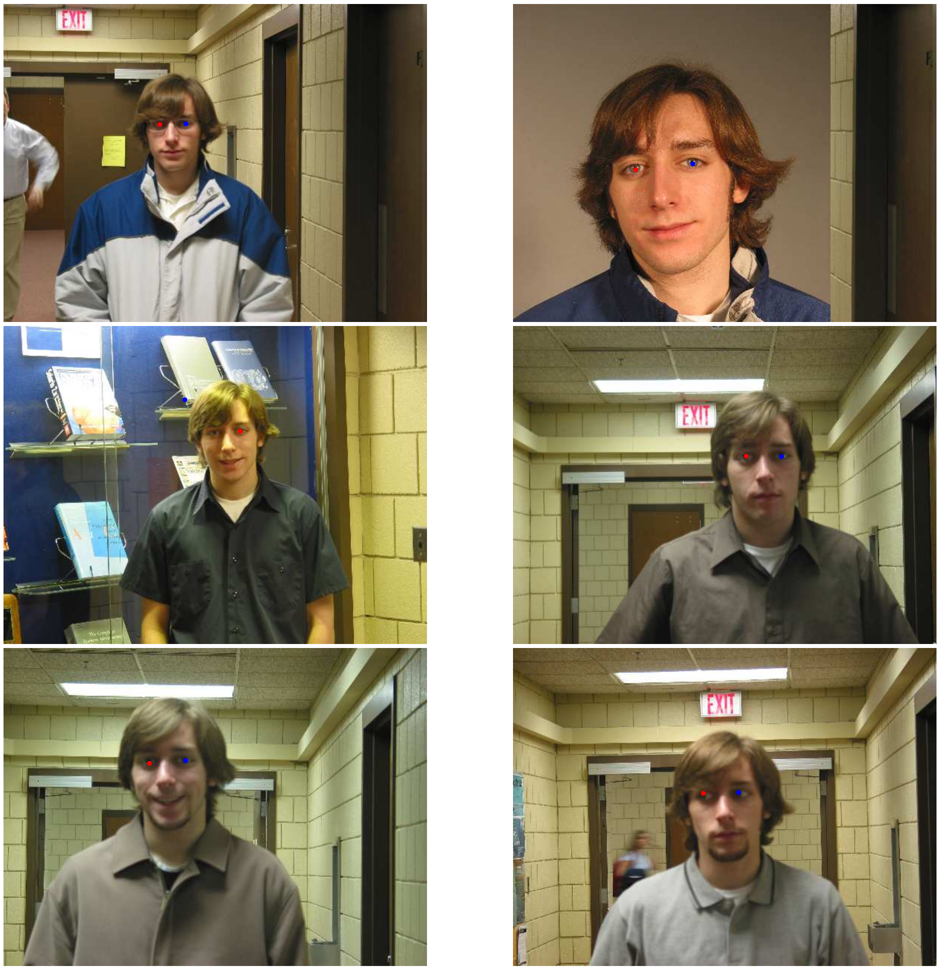
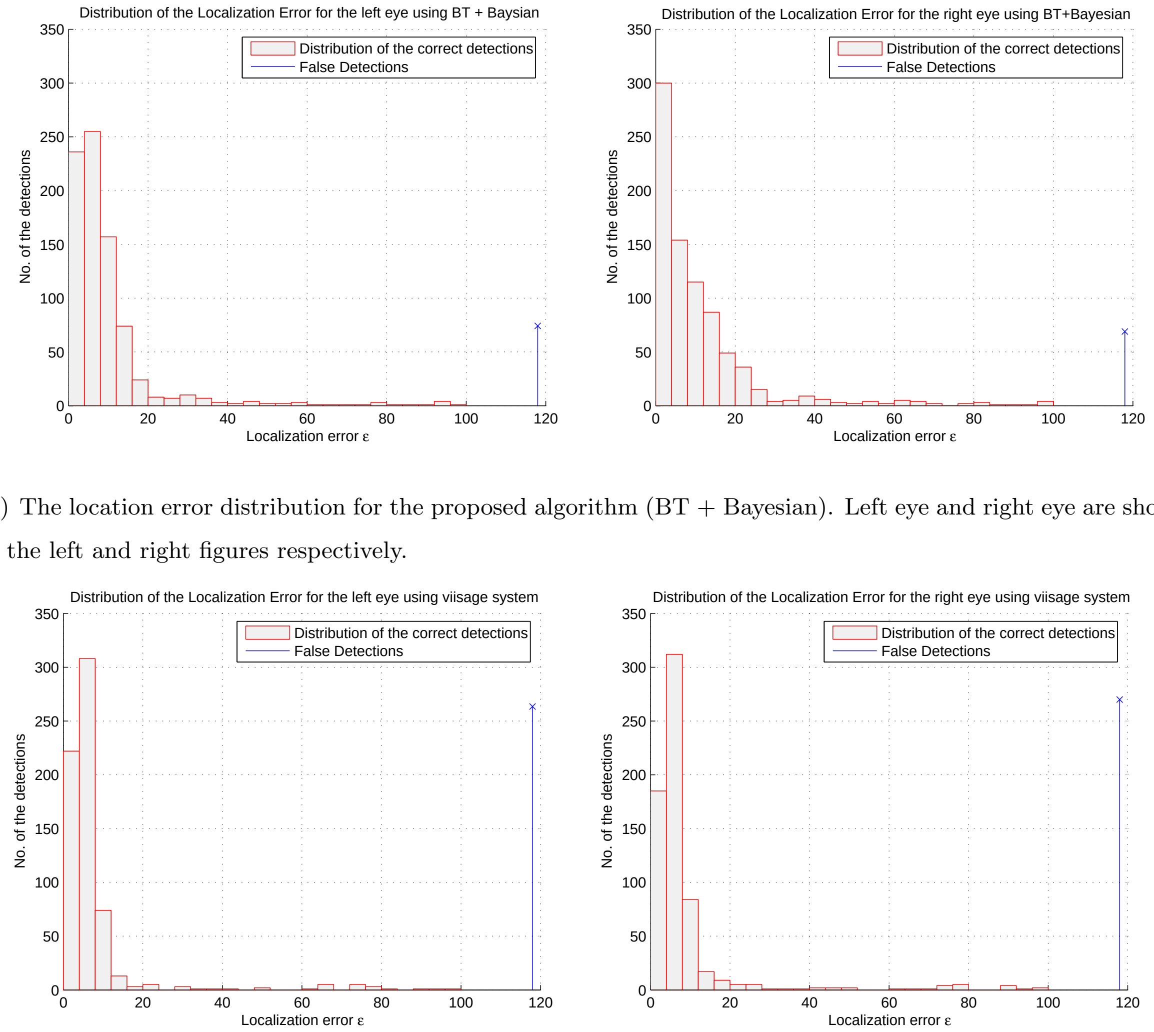


Figure III.16: More detection results for the same subject under certain illumination changes. Different color marks the output from the left and right eye models. Each image is subsamples to 0.125×0.125 of the original image size.

with CVPR 2005, San Diego, CA, 2005 and Junwen Wu and Mohan M. Trivedi, “A Binary Tree Based Probability Learning in Eye Detection: Framework and Evaluations.” Submitted for Publication, *IEEE Transactions on System, Man and Cybernetics.* I was the primary researcher of the cited material and the co-author listed in these publication directed and supervised the research which forms a basis for this chapter.



(a) The location error distribution for the proposed algorithm (BT + Bayesian). Left eye and right eye are shown in the left and right figures respectively.

(b) The location error distribution for the given results from Viisage system. Left eye and right eye are shown in the left and right figures respectively.

Figure III.17: Histograms of the localization error. Top row shows the results from the proposed algorithm, and the bottom row shows the results from the Viisage system. Right column: right eye; left column: left eye.

Chapter IV

Simultaneous Eye Tracking and Blink Detection with Interactive Particle Filters

IV.A Algorithm Overview

In this work we present an algorithm that simultaneously track eyes and detect eye blinks. We use two interactive particle filters, one tracks the open eyes and the other one tracks the closed eyes. The set of particles that gives higher confidence is defined as the primary particle set and the other one is defined as the secondary particle set. Estimates of the eyes' location, as well as the eye class labels (open-eye *v.s.* closed-eye), are determined by the primary particle filter, which is also used to re-initialize the secondary particle filter for the new observation. For each particle filter, the state variables characterize the location and size of the eyes. We use auto-regression (AR) models to describe the state transitions, where the location is modeled by a second order AR and the scale is modeled by a separate first order AR. The observation model is a classification-based model, which tracks eyes according to the knowledge learned from examples instead of the templates adapted from previous frames. Therefore, it can avoid accumulation of the tracking errors. A regression model in the tensor subspace is exploited to measure the posterior probabilities of the observations. Other classification/regression models can be used as well. Figure IV.1 gives the diagram of the system.

IV.B Dynamic Systems and Particle Filters

The dynamics of eyes can be modeled by a dynamic system. Due to the fact that open eyes and closed eyes appear to have significantly different appearances, a straightforward way is to model the dynamics of open-eye and closed-eye individually. We use two interactive particle filters for this purpose. The posterior probabilities learned by the particle filters are used to determine which particle filter gives the correct tracks, and this particle filter is thus labeled as the primary one. Since the particle filters are the key part of this blink detection system, in this section, we present a detailed overview of the dynamic system and its particle filtering solutions, such that the proposed system for simultaneous eye tracking

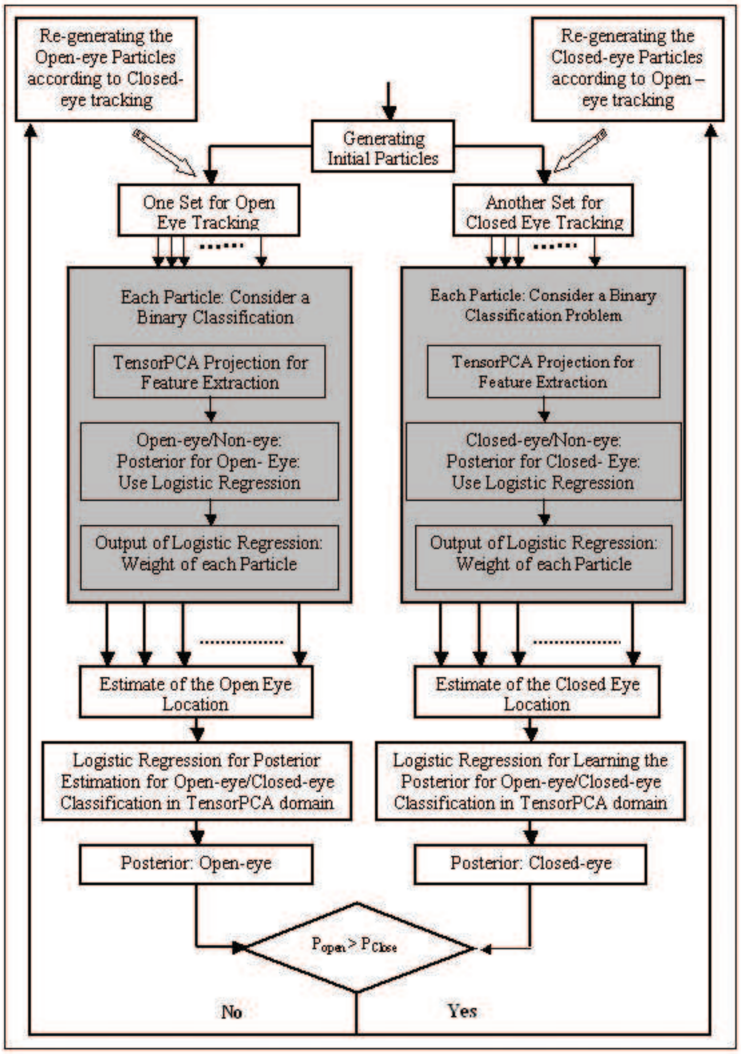

Re-generating the Open-eye Particles according to Closed-eye tracking
Re-generating the Closed-eye Particles according to Open – eye tracking
Generating Initial Particles
One Set for Open Eye Tracking
Another Set for Closed Eye Tracking
Each Particle: Consider a Binary Classification
TensorPCA Projection for Feature Extraction
Open-eye/Non-eye: Posterior for Open- Eye: Use Logistic Regression
Output of Logistic Regression: Weight of each Particle
Each Particle: Consider a Binary Classification Problem
TensorPCA Projection for Feature Extraction
Closed-eye/Non-eye: Posterior for Closed- Eye: Use Logistic Regression
Output of Logistic Regression: Weight of each Particle
Estimate of the Open Eye Location
Estimate of the Closed Eye Location
Logistic Regression for Posterior Estimation for Open-eye/Closed-eye Classification in TensorPCA domain
Logistic Regression for Learning the Posterior for Open-eye/Closed-eye Classification in TensorPCA domain
Posterior: Open-eye
Posterior: Closed-eye
P_open > P_Close
No
Yes

Figure IV.1: Flow-chart for eye blink detection system.

and blink detection can be better understood.

IV.B.1 Dynamic Systems [102, 104, 111, 114, 116]

A dynamic system can be described by two mathematical models. One is the state-transition model, which describes the system evolution rules. The other one is the observation model, which shows the relationship between the observable measurement of the system and the underlying hidden state variables. The state evolution can be represented as a stochastic process $\{\mathbf{S}_t\} \in \mathcal{R}^{n_s \times 1}$ $(t = 0, 1, \cdots)$, where:

$$\mathbf{S}_t = F_t(\mathbf{S}_{t-1}, \mathbf{V}_t). \tag{IV.1}$$

$\mathbf{V}_t \in \mathcal{R}^{n_v \times 1}$ is the state transition noise with known PDF $p(\mathbf{V}_t)$. The dynamic system is observed at discrete times via realization of the stochastic process, which is modeled as follows:

$$\mathbf{Y}_t = H_t(\mathbf{S}_t, \mathbf{W}_t). \tag{IV.2}$$

$\mathbf{Y}_t$ $(t = 0, 1, \cdots)$ is the discrete observation obtained at time t. $\mathbf{W}_t \in \mathcal{R}^{n_w \times 1}$ is the observation noise with known PDF $p(\mathbf{W}_t)$, which is independent from $\mathbf{V}_t$. For simplicity, we use capital letters to refer to the random processes and lowercase letters to denote the specified realizations of the random processes. For example, $\mathbf{Y}_{0:t}$ is used to represent the sequence of the random variables $\{\mathbf{Y}_0, \mathbf{Y}_1, \cdots, \mathbf{Y}_t\}$ and $\mathbf{y}_{0:t}$ is one realization of this sequence.

Both F_t and H_t are general stochastic processes. Given that these two system models are known, the problem is to estimate any function of the state $f(\mathbf{S}_t)$ using the expectation $\mathrm{E}[f(\mathbf{S}_t)|\mathbf{Y}_{0:t}]$. If assuming F_t and H_t are linear, and the two noise PDFs, $p(\mathbf{V}_t)$ and $p(\mathbf{W}_t)$, are Gaussian, the system can be characterized by a Kalman filter [93]. In previous decades, variants of the Kalman filter were widely used for tracking problems, due to its simplicity and closed-form solution. Unfortunately, most dynamic systems cannot be simplified by the linear system and Gaussian noise assumptions, where Kalman filtering techniques only provide the first-order approximations for such dynamic systems. The non-Gaussianity and non-linearity thus need to be incorporated for more accurate studies. The Extended Kalman Filter (EKF) [93] is one way to handle the non-linearity. A more general framework is provided by particle filtering techniques. The particle filter is a Monte-Carlo solution for general form dynamic systems. As an alternative to the EKF, particle filters have the advantage that, with sufficient samples, they approach the Bayesian estimate. In other words, although particle filters also give approximate solutions, when enough particles present, these approximations approach optimal.

IV.B.2 Review of a Basic Particle Filter

Particle filters are sequential analogues of Markov Chain Monte Carlo (MCMC) batch methods. They are also known as Sequential Monte Carlo methods (SMC). Particle filters are widely used in positioning, navigation and tracking for modeling a dynamic system [94, 95, 96, 97, 98, 99, 100]. The basic idea of particle filtering is to use point mass, or particles, to represent the probability densities. The tracking problem can be expressed as a Bayes filtering problem, in which the posterior distribution of the target state is updated recursively when a new observation comes in:

$$p(\mathbf{S}_t|\mathbf{Y}_{0:t}) \propto p(\mathbf{Y}_t|\mathbf{S}_t;\mathbf{Y}_{0:t-1}) \int_{\mathbf{S}_{t-1}} p(\mathbf{S}_t|\mathbf{S}_{t-1};\mathbf{Y}_{0:t-1})p(\mathbf{S}_{t-1}|\mathbf{Y}_{0:t-1})d\mathbf{S}_{t-1}. \quad \text{(IV.3)}$$

The likelihood $p(\mathbf{Y}_t|\mathbf{S}_t;\mathbf{Y}_{0:t-1})$ is the observation model, and $p(\mathbf{S}_t|\mathbf{S}_{t-1};\mathbf{Y}_{0:t-1})$ is the state transition model.

The particle filter is a Monte-Carlo implementation for the Bayes filters. There are several versions of the particle filters, such as Sequential Importance Sampling (SIS) [101, 102, 103, 104, 105]/Sampling-Importance Resampling (SIR) [103, 105, 106], auxiliary particle filters [105, 107] and Rao-Blackwellized particle filters [100, 105, 108, 109] etc. All particle filters are derived based on the following two assumptions. The first assumption is that the state-transition is a first order Markov process. If we have the PDF:

$$\mathbf{S}_t|\mathbf{S}_{t-1} \sim p_{\mathbf{S}_t|\mathbf{S}_{t-1}}(\mathbf{S}|\mathbf{S}_{t-1}).$$

Then the state transition model in Eq. IV.3 can be written as:

$$p(\mathbf{S}_t|\mathbf{S}_{t-1};\mathbf{Y}_{0:t-1}) = p(\mathbf{S}_t|\mathbf{S}_{t-1}). \quad \text{(IV.4)}$$

The second assumption is that the observations $\mathbf{Y}_{1:t}$ are conditionally independent given known states $\mathbf{S}_{1:t}$; which implies that each observation only relies on the current state. With known PDF:

$$\mathbf{Y}_t|\mathbf{S}_t \sim p_{\mathbf{Y}|\mathbf{S}}(\mathbf{Y}|\mathbf{S}_t),$$

we have:

$$p(\mathbf{Y}_t|\mathbf{S}_t;\mathbf{Y}_{0:t-1}) = p(\mathbf{Y}_t|\mathbf{S}_t). \tag{IV.5}$$

These two assumptions simplify the Bayes filter in Eq. IV.3 to:

$$p(\mathbf{S}_t|\mathbf{Y}_{0:t}) \propto p(\mathbf{Y}_t|\mathbf{S}_t)\int_{\mathbf{S}_{t-1}} p(\mathbf{S}_t|\mathbf{S}_{t-1})p(\mathbf{S}_{t-1}|\mathbf{Y}_{0:t-1})d\mathbf{S}_{t-1}. \tag{IV.6}$$

Exploiting this, particle filter uses a number of particles $(\omega^{(i)}, \mathbf{s}_t^{(i)})$ to sequentially compute the expectation of any function of the state, which is $\mathrm{E}[f(\mathbf{S}_t)|\mathbf{y}_{0:t}]$, by:

$$\mathrm{E}[f(\mathbf{S}_t)|\mathbf{y}_{0:t}] = \int f(\mathbf{s}_t)p(\mathbf{s}_t|\mathbf{y}_{0:t})d\mathbf{s}_t = \sum_i \omega_t^{(i)} f(\mathbf{s}_t^{(i)}). \tag{IV.7}$$

This is the theoretical foundation of all particle filters.

IV.B.3 SIS/SIR Particle Filter

In SIS/SIR particle filters [101, 102, 103, 104, 105, 106], the filtering distribution $p(\mathbf{s}_t|\mathbf{y}_{0:t})$ is approximated by a weighted set of particles: $\{(\omega_t^{(i)}, \mathbf{s}_t^{(i)}) : i = 1, \cdots, N\}$. $\omega_t^{(i)}$ are called importance weights, which satisfy:

$$\sum_{i=1}^{N} \omega_t^{(i)} = 1. \tag{IV.8}$$

At each time t, the cloud of the particles empirically measures the posterior distribution of the state given the past observations, or $\pi(\mathbf{s}_t|\mathbf{s}_{0:t-1}, \mathbf{y}_{0:t})$. Eq. IV.6 tells us that the estimation is achieved by a prediction step, $\int_{\mathbf{s}_{t-1}} p(\mathbf{s}_t|\mathbf{s}_{t-1})p(\mathbf{s}_{t-1}|\mathbf{y}_{0:t-1})d\mathbf{s}_{t-1}$, followed by an update step, $p(\mathbf{y}_t|\mathbf{s}_t)$. At the prediction step, the new state $\hat{\mathbf{s}}_t^i$ is sampled from the state evolution process $F_{t-1}(\mathbf{s}_{t-1}^{(i)}, \cdot)$ to generate a new cloud of particle filters. The empirical distribution of this new cloud of particles is an approximation to the conditional probability distribution of $\mathbf{s}_t$ given the observations up to time $t-1$, which is $\pi(\mathbf{s}_t|\mathbf{s}_{0:t-1}, \mathbf{y}_{0:t-1})$. With the predicted state $\hat{\mathbf{s}}_t^i$, an estimate of the observation is obtained and is used in the update step to correct the posterior estimate. Each particle is then re-weighted in proportion to the likelihood of the observation at time t.

Different from the SIS particle filters, SIR particle filters have a "resampling" step. By resampling, samples with low importance ratios are eliminated and samples with high importance ratios are multiplied, so that the degeneracy problem can be alleviated. There are different resampling schemes, such as the sampling by residue [104] and systematic sampling [110]. SIR and SIS can also be combined under the idea of "resampling when necessary". As suggested by [101, 111, 112], resampling is only necessary when the effective number of particles is sufficiently low. In this way, divergence between the empirical density and the true one can be controlled. We use the combination of SIS and SIR in our work. The SIS/SIR algorithm can be summarized as in Algorithm 3.

Algorithm 3 SIS/SIR particle filter

```
For $i = 1, \cdots, N$, draw samples from the importance distributions (prediction step):
    $$\mathbf{s}_t^{(i)} \sim \pi(\mathbf{s}_t|\mathbf{s}_{0:t-1}, \mathbf{y}_{0:t}); \qquad \text{(IV.9)}$$
Evaluate the importance weights for every particle up to a normalized constant (update step):
    $$\hat{\omega}_t^{(i)} = \omega_{t-1}^{(i)} \frac{p(\mathbf{y}_t|\mathbf{s}_t^{(i)})p(\mathbf{s}_t^{(i)}|\mathbf{s}_{t-1}^{(i)})}{\pi(\mathbf{s}_t^{(i)}|\mathbf{s}_{0:t-1}^{(i)}, \mathbf{y}_{0:t})}; \qquad \text{(IV.10)}$$
Normalize the importance weights:
    $$\omega_t^{(i)} = \frac{\hat{\omega}_t^{(i)}}{\sum_{j=1}^{N} \hat{\omega}_t^{(j)}} \prod i = 1, \cdots, N; \qquad \text{(IV.11)}$$
Compute an estimate of the effective number of the particles:
    $$N_{eff} = \frac{1}{\sum_{i=1}^{N} (\omega_k^{(i)})^2}; \qquad \text{(IV.12)}$$
If $N_{eff} < \theta$, where $\theta$ is a given threshold, we perform resampling. $N$ particles are drawn from the current particle set with probabilities proportional to their weights. Replace the current particle set with this new one, and reset each new particle's weight to $\frac{1}{N}$.
```

$\pi(\mathbf{s}_t^{(i)}|\mathbf{s}_{0:t-1}^{(i)}, \mathbf{y}_{0:t}) = \pi(\mathbf{s}_t^{(i)}|\mathbf{s}_{t-1}^{(i)}, \mathbf{y}_{0:t})$ is also called the proposal distribution. A common and simple choice is to use the prior distribution [113] as the proposal distribution, which is also known as a bootstrap filter. For simplicity, we use the bootstrap filter in our work. Therefore, the weight update step Eq. IV.10

now becomes:

$$\hat{\omega}_t^{(i)} = \omega_{t-1}^{(i)} p(\mathbf{y}_t | \mathbf{s}_t^{(i)}). \tag{IV.13}$$

IV.C The Proposed Interactive Particle Filters

The appearance of eyes presents significant changes when blinks occur. To effectively handle such appearance changes, we use two interactive particle filters, one for open eyes and another one for closed eyes. These two particle filters are only different in the observation measurement. In the following sections, we present the three elements of the proposed particle filters: state transition model, observation model and prediction/update scheme.

IV.C.1 State Transition Model

The system dynamics, which are described by the state variables, are defined by the eye's location and the size of the eye image patches. The state vector is: $\mathbf{S}_t = (u_t, v_t; \rho_t)$, where (u_t, v_t) defines the location and ρ_t is the scale. ρ_t is used to define the size of eye image patches and normalize them to a fixed size. In other words, the state vector $(u_t, v_t; \rho_t)$ means that the image patch under study is centered at (u_t, v_t) and its size is $32\rho_t \times 64\rho_t$, where 32×64 is the fixed size of the eye patches we use in our study.

A second-order autoregressive (AR) model is used for estimating the eyes' movement. The AR model has been widely used in particle filter tracking literature for modeling the motion. It can be written as:

$$\begin{cases} \mathbf{u}_t = \bar{\mathbf{u}} + \mathbf{A}(\mathbf{u}_{t-1} - \bar{\mathbf{u}}) + \mathbf{B}\boldsymbol{\mu}_t \\ \mathbf{v}_t = \bar{\mathbf{v}} + \mathbf{A}(\mathbf{v}_{t-1} - \bar{\mathbf{v}}) + \mathbf{B}\boldsymbol{\mu}_t \end{cases}; \tag{IV.14}$$

where:

$$\mathbf{u}_t = \begin{pmatrix} u_t \\ u_{t-1} \end{pmatrix} \quad \text{and} \quad \mathbf{v}_t = \begin{pmatrix} v_t \\ v_{t-1} \end{pmatrix}. \tag{IV.15}$$

$\bar{\mathbf{u}}$ and $\bar{\mathbf{v}}$ are the corresponding mean values for u and v respectively. As pointed out by [114], this dynamic model is actually a temporal Markov chain. It is capable of capturing complicated object motion. $\mathbf{A}$ and $\mathbf{B}$ are matrices representing the deterministic and the stochastic components respectively. $\mathbf{A}$ and $\mathbf{B}$ can be either obtained by a maximum-likelihood estimation or set manually from prior knowledge. $\boldsymbol{\mu}_t$ is the i.i.d. Gaussian noise.

We use a first-order AR to model the scale transition, which is:

$$\rho_t - \bar{\rho} = C(\rho_{t-1} - \bar{\rho}) + D\eta_t. \tag{IV.16}$$

Similar to the motion model, C is the parameter describing the system deterministic component, and D is the parameter describing the system stochastic component. $\bar{\rho}$ is the mean value of the scales, and η_t is the i.i.d. measurement noise. We assume η_t is uniformly distributed. The scale is crucial for many image appearance based classifiers. An incorrect scale causes a significant difference in the image appearance. Therefore, the scale transition model is one of the most important prerequisite for obtaining an effective particle filter for measuring the observation. Experimental evaluation shows that the AR model with uniform i.i.d noise is appropriate for tracking the scale changes.

IV.C.2 Classification-based Observation Model

In literature, many efforts have been done to address the problem of selecting the proposal distribution [95, 115, 116, 117, 118]. A carefully selected proposal distribution can alleviate the sample depletion problem, which refers to the problem that the particle-based posterior approximation collapses over time to a few particles. However, the introduction of complicated proposal distributions greatly increases the computational complexity. Also, the selection of a good proposal distribution is not a trivial issue.

In this work, we focus on a better observation model $p(\mathbf{y}_t|\mathbf{s}_t)$. The rationale is based on the observation that combined with the resampling step, a more accurate likelihood learning from a better observation model can move the particles to areas of high likelihood. This will in turn mitigate the sample depletion problem, leading to a significant increase in performance. In literature, many existing approaches use simple on-line template matching [96, 98, 99, 119] to get the observation model, where the templates are constructed from low-level features, such as color, edges, contour etc, from previous observations. The likelihood is usually estimated based on a Gaussian distribution assumption [108, 117]. However, such approaches in a large extent rely on a reasonably stable feature detection algorithm. Also, usually a large number of the single low-level feature points are needed. For example, the contour-based method requires that the state vector be able to describe the evolution of all contour points. This results in a high dimensional state space. Correspondingly, the computational cost is expensive. One solution is to use abstracted statistics of these single feature points, such as using color histogram instead of direct color measurement. However, this causes a loss in the spatial layout information, which implies a sacrifice in the localization accuracy. Instead we use a subspace-based classification model for measuring the observation to get a more accurate probability evaluation. Statistics learned from a set of training samples are used for classification instead of simple template matching and online updating, such that the problem of error accumulation can be alleviated. The likelihood estimation problem, $p(\mathbf{y}_t^{(i)}|\mathbf{s}_t^{(i)})$, becomes a problem of estimating the distribution of a bernoulli variable, which is $p(y_t^{(i)} = 1|\mathbf{s}_t^{(i)})$. $y_t^{(i)} = 1$ means that the current state generates a positive example. In our eye tracking and blink detection problem, it represents that an eye patch is located. Logistic regression is a straight-forward solution for this purpose. Obviously, other existing classification/regression techniques can be used as well.

Such classification-based particle filtering framework makes simultaneous tracking and recognition feasible and straightforward. There are two different ways to embed the recognition problem. The first approach is to use a single particle filter, whose observation model is a multi-class classifier. The second approach is to use multiple particle filters, where for each particle filter its observation model uses a binary classifier designed for a specific object class. The particle filter who gets the highest posterior is used to determine the class label as well as the object location, and at the next frame $t + 1$, the other particle filters are re-initialized accordingly. We use the second approach. Individual observation models are built for open-eye and closed-eye separately, such that two interactive sets of particles can be obtained. The observation models contain two parts: tensor subspace analysis for feature extraction, and logistic regression for class posterior learning. The two parts are individually discussed below. Posterior probabilities measured by particles from these two particle filters are individually denoted as $p_o = p(y_t = 1_{oe}|\mathbf{s}_t)$ and $p_c = p(y_t = 1_{ce}|\mathbf{s}_t)$ respectively; where $y_t = 1_{oe}$ refers to the presence of an open-eye and $y_t = 1_{ce}$ refers to the presence of a closed-eye.

Subspace Analysis for Feature Extraction

Most existing applications of using particle filters for visual tracking involve high-dimensional observations. With the increase of the dimensionality in observations, the number of particles required increases exponentially. Therefore, lower dimensional feature extraction is necessary. Sparse low-level features, such as the abstracted statistics of the low-level features, have been proposed for this purpose. Examples of the most commonly used features are color histogram [118, 120], edge density [95, 121], salient points [122] and contour points [98, 99] etc. The use of such features makes the system capable of easily accommodating the scale changes and handling occlusions, however, the performance relies on the robustness of the feature detection algorithms. Instead of these variants of low-level features, we use eigen-subspace for feature extraction and dimensionality reduction. Eigenspace projection provides a holistic feature representation that preserves spatial and textural information. It has been widely exploited in computer vision applications. For example, eigen-face has been an effective face recognition technique for decades. Eigen-face focuses on finding the most representative lower-dimensional space in which the pattern of the input can be best described. It tries to find a set of “standardized face ingredients” learned from a set of given face samples. Any face image can be decomposed as the combination of these standard faces. However, this principal component analysis (PCA) based technique treats each image input as a vector, which causes the ambiguity in image local structure. Also, with the increase of the the input image size, the computational cost increases.

Instead of PCA, in [83, 84, 123], a natural alternative for PCA in image domain is proposed, which is the multi-linear analysis. Multi-linear analysis offers a potent mathematical framework for analyzing the multi-factor structure of the image ensemble. For example, a face image ensemble can be analyzed from the following perspectives: identities, head poses, illumination variations and facial expressions. Multi-linear analysis uses tensor algebra to tackle the problem of disentangling these constituent factors. By this way, the sample structures can be better explored and a more informative data representation can be achieved. Under different optimization criterion, variants of the multi-linear analysis technique have been proposed. One solution is the direct expansion of the PCA algorithm, TensorPCA from [84], which is obtained under the criteria of the least reconstruction error. Both PCA and tensorPCA are unsupervised techniques, where the class labels are not incorporated in such representations. Here we use a supervised version of the tensor analysis algorithm, which is called tensor subspace analysis (TSA) [123]. Extended from locality preservation projections (LPP) [124], TSA detects the intrinsic geometric structure of the tensor space by learning a lower dimensional tensor subspace. We compare both observation models of using tensorPCA and TSA. TSA preserves the local structure in the tensor space manifold, hence a better performance should be obtained. Experimental evaluation validates this conjecture. In the following paragraphs, a brief review of the theoretical fundamentals of PCA, tensorPCA and TSA are presented.

I. PCA for subspace projection

PCA is a widely used method for dimensionality reduction. PCA offers a well-defined model, which aims to find the subspace that describes the direction of the most variance and at the same time suppress known noise as well as possible. The subspace from PCA projection is spanned by the principal eigenvectors of the covariance matrix, which is shown in Eq. IV.17. Suppose the samples are represented by vectors $\{\mathbf{x}_i\}, i = 1, \cdots, N$, where $\mathbf{x}_i \in \mathcal{R}^{n\times 1}$, the covariance matrix is:

$$\mathbf{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^{\mathrm{T}}. \tag{IV.17}$$

$\boldsymbol{\mu}$ is the sample mean, defined by:

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i. \tag{IV.18}$$

The principal components are computed by solving the eigenvalue problem of Eq. IV.19 [88]:

$$\mathbf{\Sigma V} = \mathbf{V\Lambda}; \tag{IV.19}$$

where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$ is the diagonal eigenvalue matrix whose diagonal entries $\lambda_1 > \lambda_2 > \cdots > \lambda_n$ are the corresponding eigenvalues of $\mathbf{\Sigma}$. $\mathbf{V}$ is the matrix whose columns are the corresponding eigenvectors. The eigenvalues of the covariance matrix indicate the amount of information preserved on the corresponding principal directions. By picking the eigenvectors with the largest eigenvalues, the information lost is minimized in the mean-square sense. Different levels of data abstraction can be controlled by the number of the chosen eigenvectors, which also preserves a varying amount of energy of the original data. The reduced PCA subspace is formed by the first P eigenvectors, and the transformed vectors can be represented as:

$$\mathbf{z}_i = \mathbf{V}_P^{\mathrm{T}}\mathbf{x}_i; \tag{IV.20}$$

where $\mathbf{V}_P$ is the $n \times P$ matrix formed by the first P eigenvectors and $\mathbf{z}_i$ is the obtained projection vector.

For frame t, each particle i determines an observation $\mathbf{x}_t^{(i)}$ given its state $(u_t^{(i)}, v_t^{(i)}; \rho_t^{(i)})$. PCA is used to extract the corresponding features $\mathbf{z}_t^{(i)}$. Now the observation model becomes the problem of computing the posterior $p(y_t^{(i)} = 1|\mathbf{z}_t^{(i)})$.

PCA provides a powerful linear technique for data reduction. Images are naturally represented as second-order tensors (matrices). However, in order to use PCA to get subspace, the matrices are rewritten as vectors in lexicographical order, which introduces ambiguities in the local spatial structure. Because of this, tensor space analysis is used instead for efficient computation as well as avoiding such ambiguities.

II. TensorPCA and TSA subspaces

Tensor space analysis handles images using its natural 2D matrix representation, which better preserves the local spatial structure of images. TensorPCA subspace analysis projects a high-dimensional rank-2 tensor onto a low-dimensional rank-2 tensor space, where the tensor subspace projection minimizes the reconstruction error. Different from the traditional PCA, tensor space analysis provides techniques for decomposing the ensemble in order to disentangle the constituent factors or modes. Since the spatial location is determined by two modes: horizontal position and vertical position, tensor space analysis has the ability to preserve the spatial location, while the dimension of the parameter space is much smaller.

As a generalization of the traditional PCA, tensorPCA is defined over the second-order tensor space. Similarly as the traditional PCA, the projection finds a set of orthogonal bases that information is best preserved. Also, tensorPCA subspace projection decreases the correlation between pixels while the projected coefficient indicates the information preserved on the corresponding tensor basis. However, for tensorPCA, the set of bases are composed by second-order tensors instead of vectors. If we use matrix $\mathbf{X}_i \in \mathcal{R}^{M_1 \otimes M_2}$ to denote the original image samples, and use matrix $\mathbf{Z}_i \in \mathcal{R}^{P_1 \otimes P_2}$ as the tensorPCA projection result, tensorPCA can be simply computed by [84]:

$$\mathbf{Z}_i = \check{\mathbf{U}}^{\mathrm{T}} \mathbf{X}_i \check{\mathbf{V}}. \tag{IV.21}$$

The column vectors of the left projection matrix $\check{\mathbf{U}}$ are the eigenvectors of matrix

$$\mathbf{S}_U = \sum_{i=1}^{N} ((\mathbf{X}_i - \overline{\mathbf{X}}_m)(\mathbf{X}_i - \overline{\mathbf{X}}_m)^{\mathrm{T}});$$

and the column vectors of the right projection matrix $\check{\mathbf{V}}$ are the eigenvectors of

$$\mathbf{S}_V = \sum_{i=1}^{N}((\mathbf{X}_i - \overline{\mathbf{X}}_m)^{\mathrm{T}}(\mathbf{X}_i - \overline{\mathbf{X}}_m));$$

while $\overline{\mathbf{X}}_m = \frac{1}{N}\sum_{i=1}^{N}\mathbf{X}_i$. The dimensionality of $\mathbf{Z}_i$ reflects the information preserved, which can be controlled by a parameter α. For example, assume the left projection matrix is computed from $\mathbf{S}_U = \check{\mathbf{U}}\mathbf{C}\check{\mathbf{U}}^{\mathrm{T}}$, then the rank of the left projection matrix $\check{\mathbf{U}}$ is determined by:

$$P_1 = \arg\min_q\{\frac{\sum_{i=1}^{q} C_i}{\sum_{i=1}^{M_1} C_i} > \alpha\}; \tag{IV.22}$$

where C_i is the i-th diagonal element of the diagonal eigenvalue matrix $\mathbf{C}$ ($C_i > C_j$ if $i > j$). The rank of the right projection matrix $\check{\mathbf{V}}$, P_2, can be decided similarly.

Both PCA and TensorPCA are unsupervised techniques. It is not clear whether the information preserved is optimal for classification. Also, only the Euclidean structure is explored instead of the possible underlying non-linear local structure of the manifold. The Laplacian based dimensionality reduction technique is an alternate way which focuses on discovering the non-linear structure of the manifold [125]. It considers preserving the manifold nature while extracting the subspaces. By introducing this idea into tensor space analysis, the following objective function can be obtained [123]:

$$\min_{\mathbf{U},\mathbf{V}}\sum_{i,j}\|\mathbf{U}^{\mathrm{T}}\mathbf{X}_i\mathbf{V} - \mathbf{U}^{\mathrm{T}}\mathbf{X}_j\mathbf{V}\|\mathcal{D}_{i,j}; \tag{IV.23}$$

where $\mathcal{D}_{i,j}$ is the weight matrix of a nearest neighbor graph similar to the one used in LPP [124]:

$$\mathcal{D}_{i,j} = \begin{cases} \exp\{-\frac{(\frac{\mathbf{X}_i}{\|\mathbf{X}_i\|} - \frac{\mathbf{X}_j}{\|\mathbf{X}_j\|})^2}{2}\}, & \text{if } \mathbf{X}_i \text{ and } \mathbf{X}_j \text{ are from the same class;} \\ 0, & \text{if } \mathbf{X}_i \text{ and } \mathbf{X}_j \text{ are from different classes.} \end{cases} \tag{IV.24}$$

We use the iterative approach provided in [123] to compute the left and right projection matrices $\check{\mathbf{U}}$ and $\check{\mathbf{V}}$. The same as tensorPCA, for a given example $\mathbf{X}$, TSA gives:

$$\mathbf{Z}_i = \check{\mathbf{U}}^{\mathrm{T}}\mathbf{X}_i\check{\mathbf{V}}. \tag{IV.25}$$

Similarly as the traditional PCA, at each frame t, the i-th particle determines an observation $\mathbf{X}_t^{(i)}$ from its state $(u_t^{(i)}, v_t^{(i)}; \rho_t^{(i)})$. Tensor analysis extracts the corresponding features $\mathbf{Z}_t^{(i)}$. Now the observation model becomes computing the posterior $p(y_t^{(i)} = 1|\mathbf{Z}_t^{(i)})$. For simplicity, in the following section, we omit the time index t and denote the problem as $p(y^{(i)} = 1|\mathbf{Z}^{(i)})$. Logistic regression is a natural solution for this purpose, which is a generalized linear model for describing the probability of a bernoulli distributed variable.

Logistic Regression for Modeling Probability

Regression is the problem of modeling the conditional expected value of one random variable based on the observations of some other random variables, which are usually referred to as dependent variables. The variable to model is called the response variable. In the proposed algorithm, the dependent variables are the coefficients from the tensor subspace projection: $\mathbf{Z}^{(i)} = (z_1^{(i)}, \cdots, z_k^{(i)}, \cdots)$, and the response variable to model is the class label $y^{(i)}$, which is a bernoulli variable that defines the presence of an eye subject. For closed-eye particle filter, this bernoulli variable defines the presence of a closed eye; while for open-eye particle filter, this variable defines the presence of an open eye.

Generalized linear models are largely used in regression, whose relationship between $y^{(i)}$ and its dependent variables $(z_1^{(i)}, \cdots, z_k^{(i)}, \cdots)$ can be written as:

$$y^{(i)} = g(\beta_0 + \sum_k \beta_k z_k^{(i)}) + e; \tag{IV.26}$$

where e is the error and $g^{-1}(\bullet)$ is called the link function. The variable $y^{(i)}$ can be estimated by:

$$E(y^{(i)}) = g(\beta_0 + \sum_k \beta_k z_k^{(i)}). \tag{IV.27}$$

Logistic regression is one type of the generalized linear models, which uses the *logit* as the link function. Logit is a function as follows:

$$\text{logit}(p) = \log(\frac{p}{1-p}). \tag{IV.28}$$

It models a number p between 0 and 1. Considering the response variable in our application, it is an indication of the presence of an eye subject, which clearly has a bernoulli distribution, logistic regression can be used to estimate the probability of the response variable as follows:

$$p(y^{(i)} = 1|\mathbf{Z}^{(i)}) = \frac{e^{\beta_0 + \sum_k \beta_k z_k^{(i)}}}{1 + e^{\beta_0 + \sum_k \beta_k z_k^{(i)}}}. \qquad \text{(IV.29)}$$

This provides a way to estimate of the class posterior probability.

IV.C.3 State Update

The observation models for open-eye and closed-eye are individually trained. We have one TSA subspace learned from open-eye/non-eye training samples, and another TSA subspace learned from closed-eye/non-eye training samples. Each TSA projection determines a set of transformed features, which are denoted as: $\{\mathbf{Z}_{oe}^{(i)}\}$ and $\{\mathbf{Z}_{ce}^{(i)}\}$. $\mathbf{Z}_{oe}^{(i)}$ is the transformed TSA coefficients for the open eyes and $\mathbf{Z}_{ce}^{(i)}$ is the transformed TSA coefficients for the closed eyes. Correspondingly, for open-eye and closed-eye, individual logistic regression models are used separately for modeling p_c and p_o as follows:

$$p_o^{(i)} = p_{oe}(y^{(i)} = 1|\mathbf{Z}_{oe}^{(i)}); \quad p_c^{(i)} = p_{ce}(y^{(i)} = 1|\mathbf{Z}_{ce}^{(i)}). \qquad \text{(IV.30)}$$

The posteriors are used to update the weights of the corresponding particles, as indicated in Eq.IV.13. The updated weights are $\omega_c^{(i)}$ and $\omega_o^{(i)}$.

If we have

$$\max_i p_o^{(i)} > \max_i p_c^{(i)}, \qquad \text{(IV.31)}$$

it indicates the presence of open eyes, and the particle filter for tracking the open-eye is the primary particle filter. Otherwise the eyes of the human subject in the current frame are closed, which indicates the presence of a blink, and the particle filter for the closed-eye is determined as the primary particle filter. At frame t, assume the particles for the primary particle filter are $\{(u_t^{(i)}, v_t^{(i)}; \rho_t^{(i)}; \omega_t^{(i)})\}$, then the location (u_t, v_t) of the detected eye is determined by:

$$u_t = \sum_i \omega_t^{(i)} u_{t-1}^{(i)}; \quad v_t = \sum_i \omega_t^{(i)} v_{t-1}^{(i)}; \tag{IV.32}$$

and the scale ρ_t of the eye image patch is:

$$\rho_t = \sum_i \omega_t^{(i)} \rho_{t-1}^{(i)}. \tag{IV.33}$$

We compute the effective number of particles N_{eff}. If $N_{eff} < \theta$, we perform re-sampling for the primary particle filter. The particles with high posteriors are multiplied in proposition to their posteriors. The secondary particle filter is re-initialized by setting the particles' previous states to (u_t, v_t, ρ_t) and the importance weights $\omega_t^{(i)}$ to uniform.

IV.D Experimental Evaluation

The performance is evaluated from two aspects: the blink detection accuracy and the tracking accuracy. There are two factors that explain the blink detection rate: first, how many blinks are correctly detected; second, the detection accuracy of the blink duration. Videos collected under different scenarios are studied, including indoor videos and in-car videos. A quantitative comparison is listed. To evaluate the tracking accuracy, a benchmark data is required to provide the ground-truth. We use a marker-based motion capturing system to collect the ground-truth data. The experimental setup for obtaining the benchmark data is explained, and the tracking accuracy is presented.

IV.D.1 Blink Detection Accuracy

We use videos collected under different scenarios for evaluating the blink detection accuracy. The videos are collected for different subjects at varying times for both indoor and in-car cameras. Eight representative sequences are used for comparison, and the comparison results are summarized in Table Table IV.1. The true number of blinks, the detected number of blinks and the number of false positives are shown. Also, since eye blinking is a dynamic process, the characteristics of the camera, such as the data acquisition speed and digital *v.s.* analog, as well as the video scenarios, are also described for references. Images in Figure IV.2 $\sim$ Figure IV.5 give some examples of the detection results, which also show the typical video frames we used for studying. These four figures correspond to the first four sequences in Table Table IV.1 respectively. Red boxes show the tracked eye location, while blue dots show the center of the tracking results. If there is a red bar on the top right corner, it means that the eyes are closed in the current frame. Examples of the typical false detections or mis-detections are also shown.

Blink duration time plays an important role in HCI systems. Involuntary blinks are usually fast while voluntary blinks usually last longer [126]. Therefore, it is also necessary to compare the detected blink duration with the manually labeled true blink duration (in terms of the frame numbers). Examples are shown in Figure IV.6, where the results from the four sequences that have the most blinks are shown. The first chart is for the first sequence in Table Table IV.1, the second chart is for the third sequence in Table Table IV.1, the third chart is for the fifth sequence in Table Table IV.1 and the last chart is for the seventh sequence in Table Table IV.1. The detected blink durations and the manually labeled blink durations are displayed side by side for comparison, where the horizontal axis is the blink index, and the vertical axis shows the duration time in terms of the frame numbers. Experimental evaluation shows that the proposed algorithm is capable of capturing short blinks as well as the long voluntary blinks accurately.

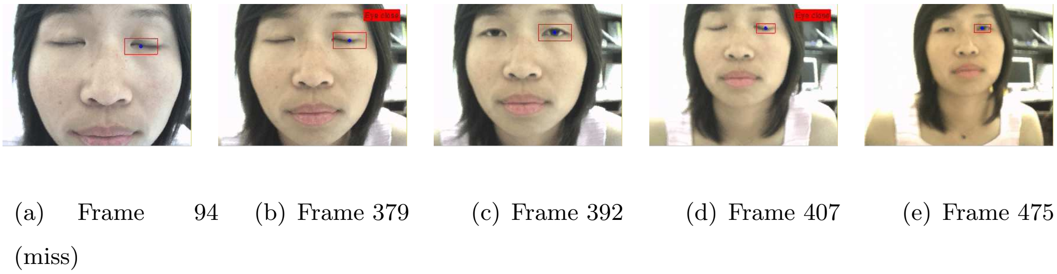
As indicated in Eq. IV.31, the ratio of the posterior maxima, which is $\frac{\max_i p_o^{(i)}}{\max_i p_c^{(i)}}$, determines the presence of an open-eye or close-eye. Figure IV.7 shows an example of the obtained ratios for one sequence. Log-scale is used. Let $p_o = \max_i p_o^{(i)}$ and $p_c = \max_i p_c^{(i)}$, the presence of the closed-eye frame is determined when $p_o < p_c$, which corresponds to $\log(\frac{p_o}{p_c}) < 0$ in the log-scale. Examples of the corresponding frames are also shown in Figure IV.8(a)-Figure IV.8(c) for illustration.

Table IV.1: Blink detection accuracy.

	Number of the Blinks	Number of the correct Detection	Number of the false positives	Video Specifics	Subject Motion
Seq 1	16	14	0	Indoor Digital Camera	Subject moves back and forth
Seq 2	7	7	2	In Car Analog Camera	Natural driving
Seq 3	19	15	1	Indoor Digital Camera	Extensive pupil motion
Seq 4	11	10	3	In Car Analog Camera	Natural driving
Seq 5	29	27	1	Indoor Digital Camera	Voluntary and involuntary blinks
Seq 6	1	1	0	Indoor, low lighting Analog Camera	Non-frontal subject view
Seq 7	13	12	0	Indoor Analog Camera	Translational motion
Seq 8	12	6	1	Indoor, low lighting Digital Camera	Non-frontal subject view, move back and forth

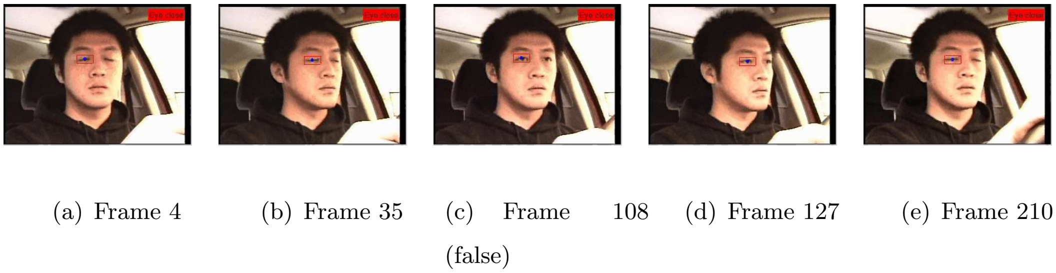
IV.D.2 Comparison of TensorPCA Subspace *v.s.* TSA Subspace

As stated above, by introducing multi-linear analysis, the images can better preserve the local spatial structure. However, variants of the tensor subspace basis can be obtained based on different objective functions. TensorPCA is a straightforward extension of the 1D PCA analysis. Both are un-supervised approaches. TSA extends LPP that preserves the non-linear locality in the manifold, which also incorporates the class information. It is believed that by introducing the local manifold structure and the class information, TSA can obtain a better performance. Experimental evaluations verified this claim. Particle filters that individually use tensorPCA subspace and TSA subspace for observation models are compared for eye tracking and blink detection purpose. Examples of the com-



(a) Frame 94 (miss) (b) Frame 379 (c) Frame 392 (d) Frame 407 (e) Frame 475

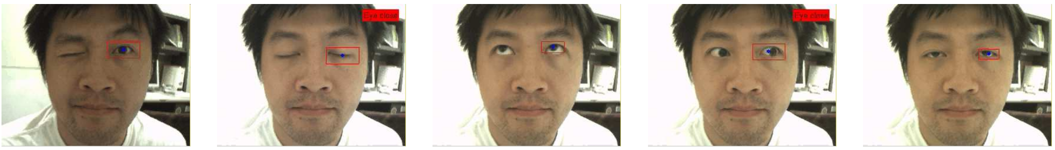
Figure IV.2: Examples of the blink detection results for the 1st sequence as described in Table Table IV.1. Red boxes are tracked eyes, and the blue dots are the center of the eye locations. The red bar on the top-left indicates the presence of closed-eyes.



(a) Frame 4 (b) Frame 35 (c) Frame 108 (false) (d) Frame 127 (e) Frame 210

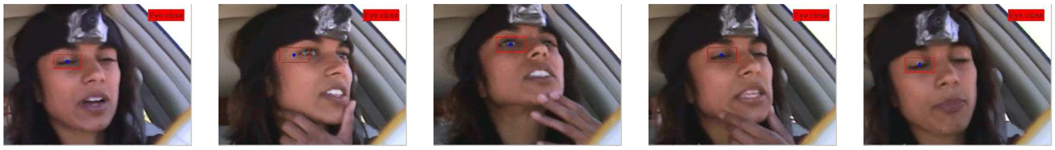
Figure IV.3: Examples of the blink detection results for the 2nd sequence as described in Table Table IV.1. Red boxes are tracked eyes, and the blue dots are the center of the eye locations. The red bar on the top-left indicates the presence of closed-eyes.

parison are shown in Figure IV.9. As suggested, TSA presents a more accurate tracking result. In Figure IV.9, examples of the tracking results from both the tensorPCA observation model and the TSA observation model are shown. In each sub-figure, the left image shows result from TSA subspace, and the right image shows result from tensorPCA subspace. Just as above, red bounding boxes show the tracked eyes, the blue dots show the center of the detection, and the red bar at the top-right corner indicates the presence of a detected closed-eye frame. For subspace based analysis, image alignment is critical for classification accuracy. An inaccurate observation model causes errors in the posterior probability computation, which in turn results in inaccurate tracking and poor blink detection.



(a) Frame 2 (b) Frame 18 (c) Frame 38 (d) Frame 45 (false) (e) Frame 135

Figure IV.4: Examples of the blink detection results for the 3rd sequence as described in Table Table IV.1. Red boxes are tracked eyes, and the blue dots are the center of the eye locations. The red bar on the top-left indicates the presence of closed-eyes.



(a) Frame 42 (b) Frame 302 (false) (c) Frame 349 (d) Frame 489 (e) Frame 769

Figure IV.5: Examples of the blink detection results for the 4th sequence as described in Table Table IV.1. Red boxes are tracked eyes, and the blue dots are the center of the eye locations. The red bar on the top-left indicates the presence of closed-eyes.

IV.D.3 Comparison of Different Scale Transition Models

It is worth noting that for subspace based observation model, the scale for normalizing the size of the images is crucial. A bad scale transition model can severely deteriorate the performance. Two popular models have been used to model the scale transition, and the performance is compared. The first one is the AR model as in Eq. IV.16, and the other one is a Gaussian transition model in which the transition is controlled by a Gaussian distributed random noise, as follows:

$$\rho_t \sim \mathcal{N}(\rho_{t-1}, \sigma^2), \tag{IV.34}$$

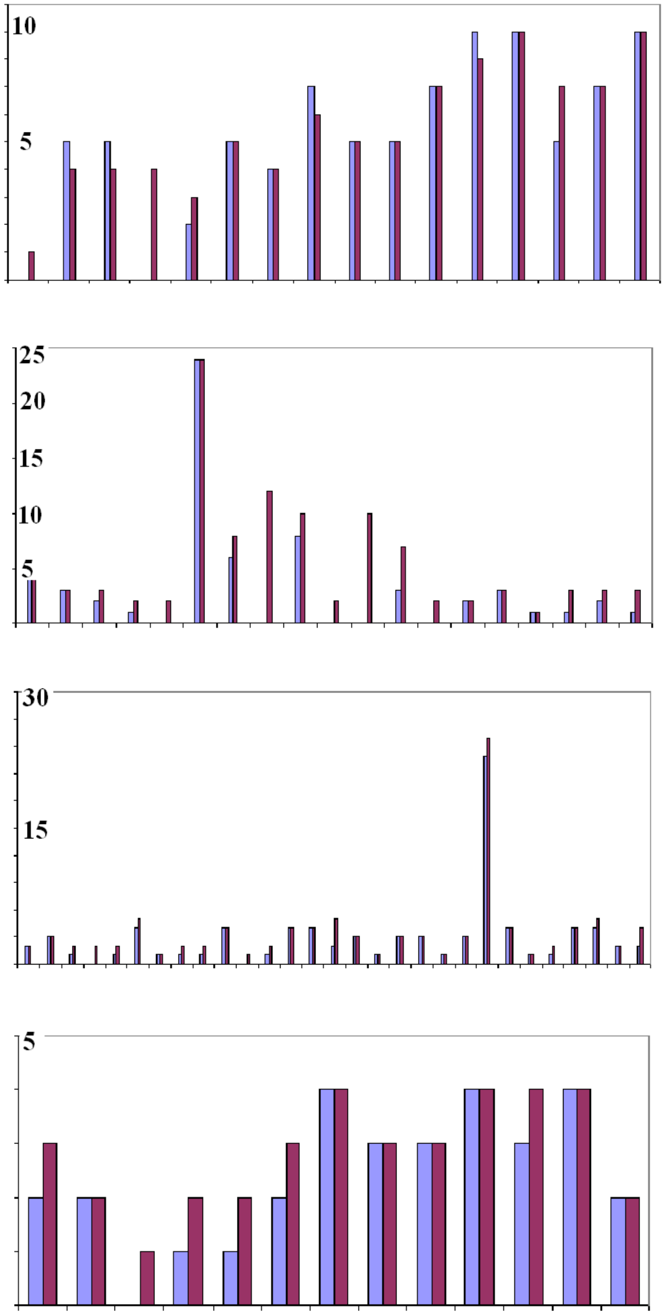


Figure IV.6: Accuracy of the blink duration time: true blink duration *v.s.* the detected blink duration. The heights of the bars: blink duration in frame numbers. The blue bars: detected blinks; the magenta bars: true blinks.

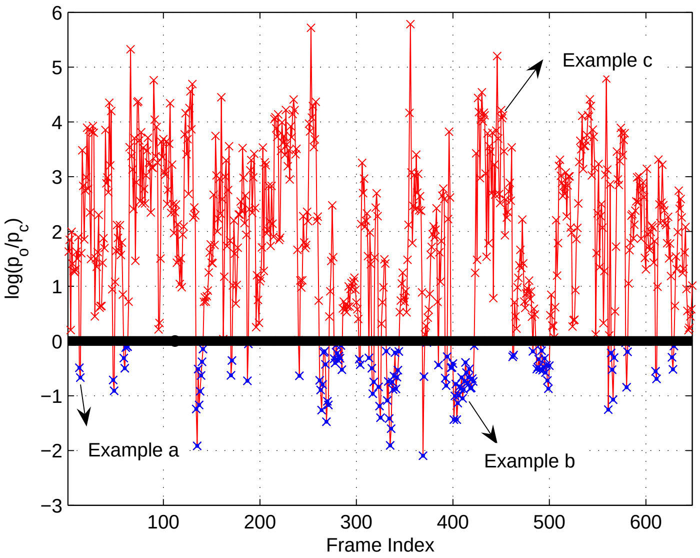


Figure IV.7: The log ratio of posteriors $\log \frac{p_o}{p_c}$ for each frame in Seq. 5. p_o: posteriors of being open-eye; p_c: posteriors of being closed-eye. Red crosses indicate the open-eye frames, and the blue crosses indicate the detected closed-eye frames.

where $\mathcal{N}(\rho, \sigma^2)$ is a Gaussian distribution with ρ as the mean and σ^2 as the variance. Examples are shown in Figure IV.10. The parameters of the Gaussian transition model is obtained by the MAP criteria according to a manually labeled training sequence. In each sub-figure, the left image shows the result from using the AR model for scale transition, and the right one shows the result from using the Gaussian transition model. Experimental results show that AR model performs better. It is because AR model has certain "memory" of the past system dynamics, while Gaussian transition model can only remember the history of its immediate past. Therefore, the "short-memory" of Gaussian transition model uses less information to predict the scale transition trajectory, which is not effective and in turn causes the failure of the tracking.

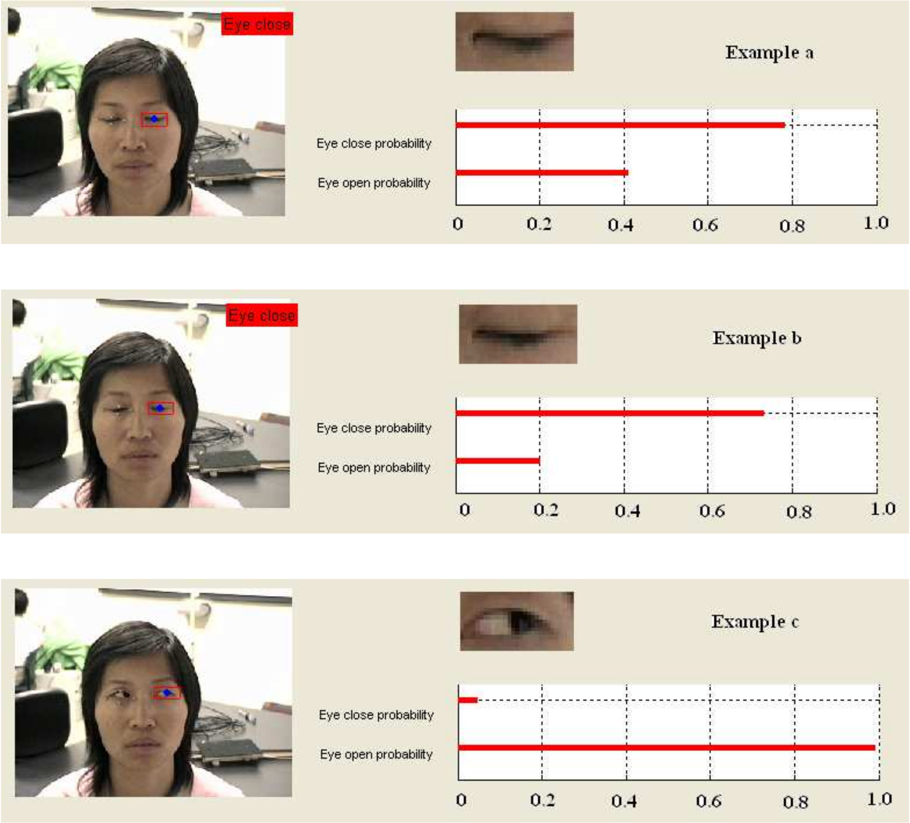
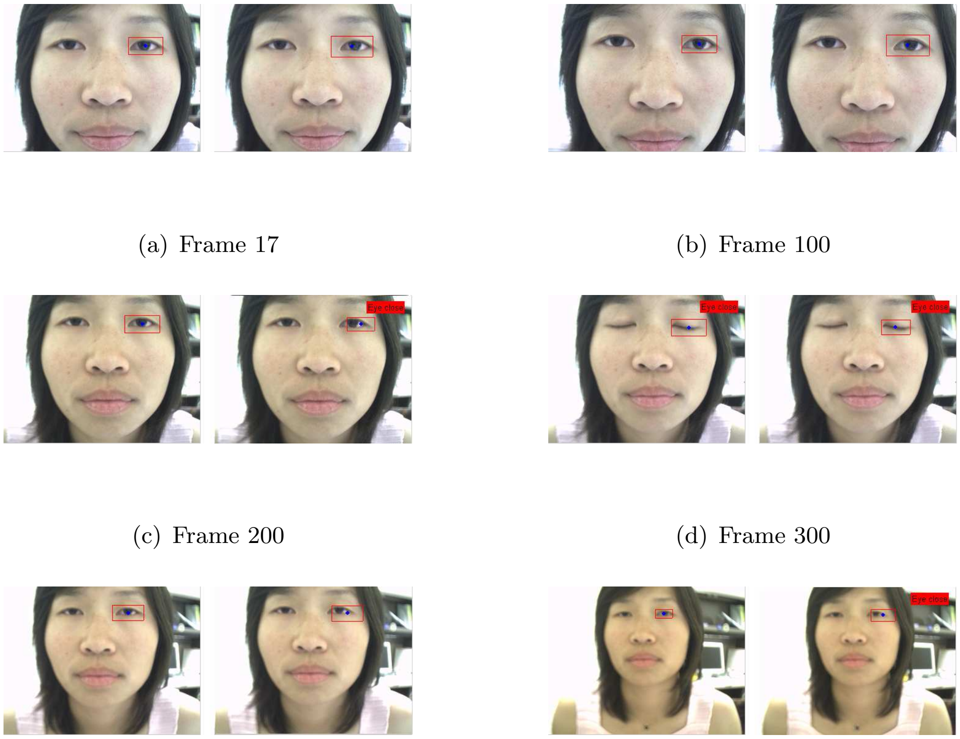


Figure IV.8: The frames corresponding to example a, b and c in Figure IV.7. The tracked eyes and the posteriors p_c and p_o are also shown. The top red lines: closed-eye posteriors; the bottom red lines: the open-eye posteriors.

IV.D.4 Eye Tracking Accuracy

Benchmark data is required for evaluating the tracking accuracy. We use the marker-based Vicon motion capture and analysis system for providing the groundtruth. Vicon system has both hardware and software components. The hardware includes a set of infrared cameras (usually at least 4), controlling hardware modules and a host computer to run the software. The software includes Vicon IQ that manages, sets up, captures and processes the motion data, the database manager for keeping records of the data files, their calibration files and the models. We use four Vicon MCAM cameras to track four reflective markers. The setup is shown as in Figure IV.11. Vicon system tracks the markers' position



(a) Frame 17

(b) Frame 100

(c) Frame 200

(d) Frame 300

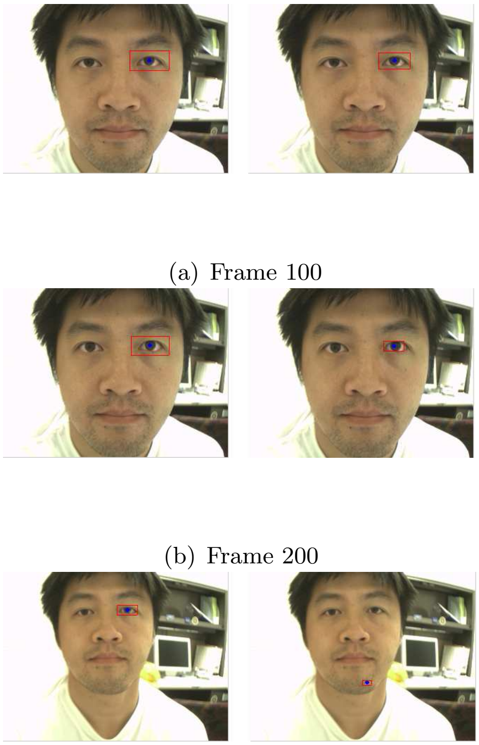
(e) Frame 400

(f) Frame 417

Figure IV.9: Comparison of using TSA subspace vs using tensorPCA subspace in observation models. In each sub-figure, the left image shows the result from using TSA subspace, and the right one shows the result from using tensorPCA subspace.

in Vicon's reference coordinate system, and the video camera collects the video we need for evaluating the proposed algorithm.

Before collecting data, Vicon system requires preprocesses including camera calibration, data acquisition and model building. With the included calibration tool for the motion capture system, a reflectance marker's 3D position can be obtained in either the Vicon camera coordinate system or an assigned world coordinate system. Since the Vicon camera coordinate system is different from the video camera coordinate system, a calibration between these two camera system is also required. We use a checker-board pattern with reflectance markers on specified location for this purpose, as shown in Figure IV.12. Intrinsic parameters **KK** and



(a) Frame 100

(b) Frame 200

(c) Frame 380

Figure IV.10: Comparison of using AR vs using Gaussian transition model in the scale model. In each sub-figure, the left image shows the result from AR scale transition model, and the right one shows the result from the Gaussian scale transition model.

extrinsic parameters $\mathbf{R}_e$ and $\mathbf{T}_e$ are computed. Intrinsic parameters give the transform from the 3D coordinates in the camera reference frame to the 2D coordinates in the image domain; while extrinsic parameters define the transform between the grid reference frame (as shown in Figure IV.13) and the camera reference frame. From intrinsic parameters, the 3D coordinates in the camera coordinate system $(X_c, Y_c, Z_c)^{\mathrm{T}}$ can be related with the 2D coordinates in the image plane $(x_p, y_p)^{\mathrm{T}}$ by:

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \mathbf{KK}\phi(\begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \end{bmatrix}). \tag{IV.35}$$

where $\phi(\bullet)$ is a nonlinear function describing the lens distortion. Extrinsic pa-

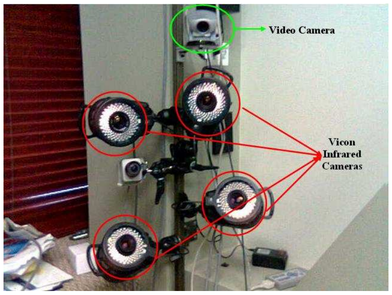


Figure IV.11: Setup for collecting groundtruth data with Vicon system. Cameras in red circles are Vicon infrared cameras, and the camera in green circle is the video camera for collecting testing sequences.

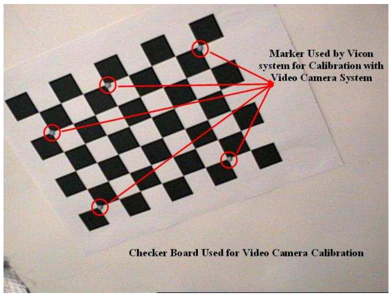

Marker Used by Vicon system for Calibration with Video Camera System
Checker Board Used for Video Camera Calibration

Figure IV.12: Checker board pattern for calibration between video camera coordinate system and Vicon camera coordinate system. Reflectance markers are put at specific locations.

rameters describe the relation between the 3D coordinate in the camera system $\mathbf{M}_c = (X_c, Y_c, Z_c)^{\mathrm{T}}$ and the 3D coordinate in a given grid reference frame $\mathbf{M}_e = (X_e, Y_e, Z_e)^{\mathrm{T}}$, as follows:

$$\mathbf{M}_c = \mathbf{R}_e \times \mathbf{M}_e + \mathbf{T}_e. \tag{IV.36}$$

Figure IV.13 gives an example of the grid reference frame. Each pose of the checker-board defines one grid reference frame, hence an individual set of extrinsic parameters can be determined. The reflectance markers are assumed to be infinitely thin, such that their depth can be neglected. Therefore, the reflectance markers's coordinates in current grid reference frame are known, denoted as $\mathbf{M}_e^i$.

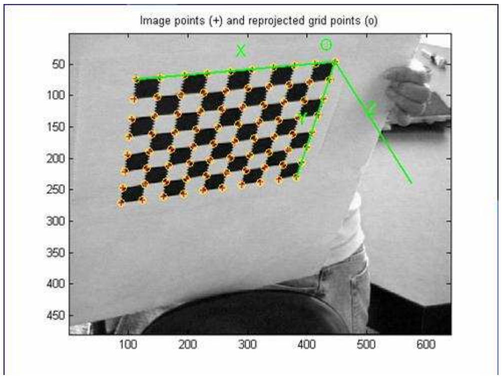


Figure IV.13: Example of the grid reference frame.

$\mathbf{M}_e^i$ can be transformed back to the video camera reference frame, which gives the 3D coordinates in the video camera reference frame $\mathbf{M}_c^i$, using the corresponding extrinsic parameters $\mathbf{R}_e^i$ and $\mathbf{T}_e^i$. These markers are also visible by the Vicon system, as shown in Figure IV.14. Calibrated Vicon system gives the 3D positions of the markers, which are denoted as $\mathbf{M}_v^i$, in the Vicon camera system reference frame. Hence, $\mathbf{M}_c^i$ and $\mathbf{M}_v^i$ can be related by an affine transform:

$$\mathbf{M}_c^i = \mathbf{R}_{vc} \times \mathbf{M}_v^i + \mathbf{T}_{vc}. \tag{IV.37}$$

This relation keeps unchanged when the pose of the check-board changes. A set of $\{(\mathbf{M}_c^i, \mathbf{M}_e^i)\}$ $(i = 1, \cdots, q)$ can be used to determine this transform. We use

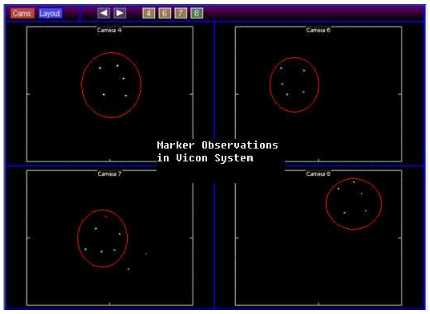


Figure IV.14: Reflectance markers observed by Vicon IQ system.

the approach proposed by Goryn and Hein in [127] to estimate $\mathbf{R}_{vc}$ and $\mathbf{T}_{vc}$. The rotation matrix $\mathbf{R}_{vc}$ can be determined by least-square approach as follows:

$$\mathbf{R}_{vc} = \mathbf{W}\mathbf{Q}^{\mathrm{T}}; \tag{IV.38}$$

where $\mathbf{W}$ and $\mathbf{Q}$ are unitary matrices obtained from SVD decomposition of the matrix:

$$\mathbf{c} = \frac{1}{N}\sum_{i=1}^{q}(\mathbf{M}_c^i - \bar{\mathbf{M}}_c)(\mathbf{M}_v^i - \bar{\mathbf{M}}_v)^{\mathrm{T}};$$

$$\bar{\mathbf{M}}_c = \frac{1}{q}\sum_{i=1}^{q}\mathbf{M}_c^i \quad \bar{\mathbf{M}}_v = \frac{1}{q}\sum_{i=1}^{q}\mathbf{M}_v^i.$$

The translation vector $\mathbf{T}_{vc}$ can be obtained accordingly by:

$$\mathbf{T}_{vc} = \bar{\mathbf{M}}_c - \mathbf{R}_{vc} \times \bar{\mathbf{M}}_v. \tag{IV.39}$$

Eq. IV.39 together with Eq. IV.35 determines the mapping from the markers' 3D position given by Vicon system to the 2D pixel position in the image plane. Therefore, with the ViconIQ system providing the markers' 3D positions in Vicon camera systems, we can get our ground-truth data. For reliable tracking, four markers are used, as shown in Figure IV.15. We use the Vicon system to track the right eye location as well as providing the scale of the image; and apply the proposed algorithm on tracking and blink detection of left eye. After normalization with the scale, the distance between the right eye and left eye is constant, so that the benchmark data can be used for evaluating the tracking accuracy. The fixed size for computing the subspace is 40 × 60. We use the center of the markers as the groundtruth for eyes' location.

Figure IV.16 gives an example of the tracking accuracy. The horizontal axis shows the frame number, and the vertical axis shows the error in pixels after normalization with the scales. The error is the distance between the center of detection to the groundtruth. Experimental results show that in certain frames, the tracking error is bigger. This is because the proposed algorithm tries to center at the pupil, instead of the center of the eyes.

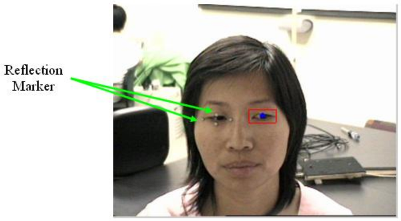


Figure IV.15: Marker deployment for tracking accuracy benchmark data collection.

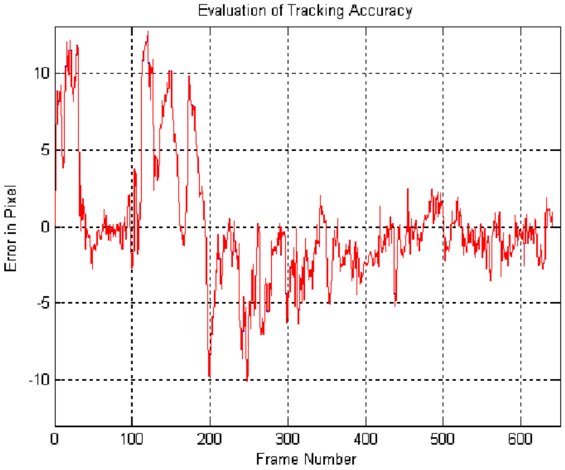


Figure IV.16: Tracking error after normalization using the scales. The horizontal axis is the frame index, and the vertical axis is the tracking error in pixels after normalization with the scales.

The text of this, in part, is a reprint of the material as it appears in: Junwen Wu and Mohan M. Trivedi, "Simultaneous Eye Tracking and Blink Detection with Interactive Particle Filters." Submitted for Publication, *IEEE Transactions on System, Man and Cybernetics.* I was the primary researcher of the cited material and the co-author listed in this publication directed and supervised the research which forms a basis for this chapter.

Chapter V

Two Stage Head Pose Estimation: Framework and Evaluation

V.A Head Pose Estimation: An Overview

The proposed solution for head pose estimation problem is a two-stage scheme in a coarse-to-fine fashion. The two-stage approach is based on the rationale that visual cues characterizing head pose has unique multi-resolution spatial frequency characterization and structural signature. In the first stage, we use subspace analysis in a Gabor wavelet transform space. Our study indicates that statistical subspace analysis is insufficient to deal with data misalignment and background noise, however, the noise does not drive the estimate far away from its true head pose. Therefore, it is reasonable to assume that the true pose is located in a subset of $p \times p$ neighboring poses around the estimate with a high accuracy. We use this subset of poses as the output from the first stage instead of the single pose estimate. The first level outputs a range in which the true pose is located. This defines a smaller classification problem. To get a comprehensive view of the underlying data structure, we examined four different subspaces to find the best subspace descriptor: Principal Component Analysis, or PCA [88]; Kernel Principal Component Analysis, or Kernel PCA [128]; Multi-class Fisher Discriminant Analysis, or FDA [88] and kernel discriminant analysis, or KDA [129, 130]. We show that analysis in the kernel space can provide a better performance. Also, discriminant analysis is slightly better than PCA.

Now we only need to determine the true pose from the small range of head poses constrained by the first level output. Since geometric structure of local facial features contains the necessary details for a finer pose assessment, in the second stage, we use a structural landmark analysis in the transformed domain to refine the estimate. More specifically, we use a revised version of the face bunch graph [62]. Face bunch graph includes two sets of elements, one is the node set and the other is the edge set. Nodes are Gabor jets of facial landmarks, which capture the appearance feature. Edges are the Euclidean distance between the nodes, which describe the geometric configuration. Since the geometric configuration modeled by the bunch graph is not subject-independent, a single bunch graph from averaging the geometric configuration of different training samples is also not subject-independent. Simple averaging is not sufficient to describe all subjects. Therefore, We use multiple bunch graphs per pose with each bunch graph built from an available geometric configuration, thereby allowing as many types of geometric configuration as possible to be accommodated. The diagram in Figure V.1 outline this algorithm. Although the structure landmarks based analysis is very time-consuming, the introduce of the two-stage strategy allow us to only examine the poses constrained by the first stage. Different from the face recognition task solved in [62], we only need to recover the identity-independent head pose. In [62], an exhaustive elastic bunch graph searching is used so as to find the fiducial points that contains subjects' identity. However, the distortion in the geometric structure caused by the exhaustive elastic search would introduce ambiguity for close poses. Furthermore, for pose estimation, we do not require the exact match of the fiducial points since the nodes from Gabor jets are actually able to describe the neighborhood property. That is the reason we use the "semi-rigid" bunch graph, in which the nodes can only be individually adjusted locally in legitimate geometric configurations. We use multiple bunch graph per pose to incorporate all available geometric structure. Since the first stage estimate restricts the possible candidate in a small subset, the computational cost is still reasonable. Moreover, by using the two-stage framework, the first stage limits the node search space such that the possibility of introducing ambiguous matching is lowered. Therefore the proposed two-stage algorithm not only help lower the computational cost, nut also it is important for performance gain.

The data set we use to evaluate the algorithm span pan angle from -90^o-$+90^o$ and the tilt angle changing from -45^o (head pointing up) $+60^o$ (head pointing down). 86 poses are included. Each pose is labeled as shown in Figure V.2 for reference.

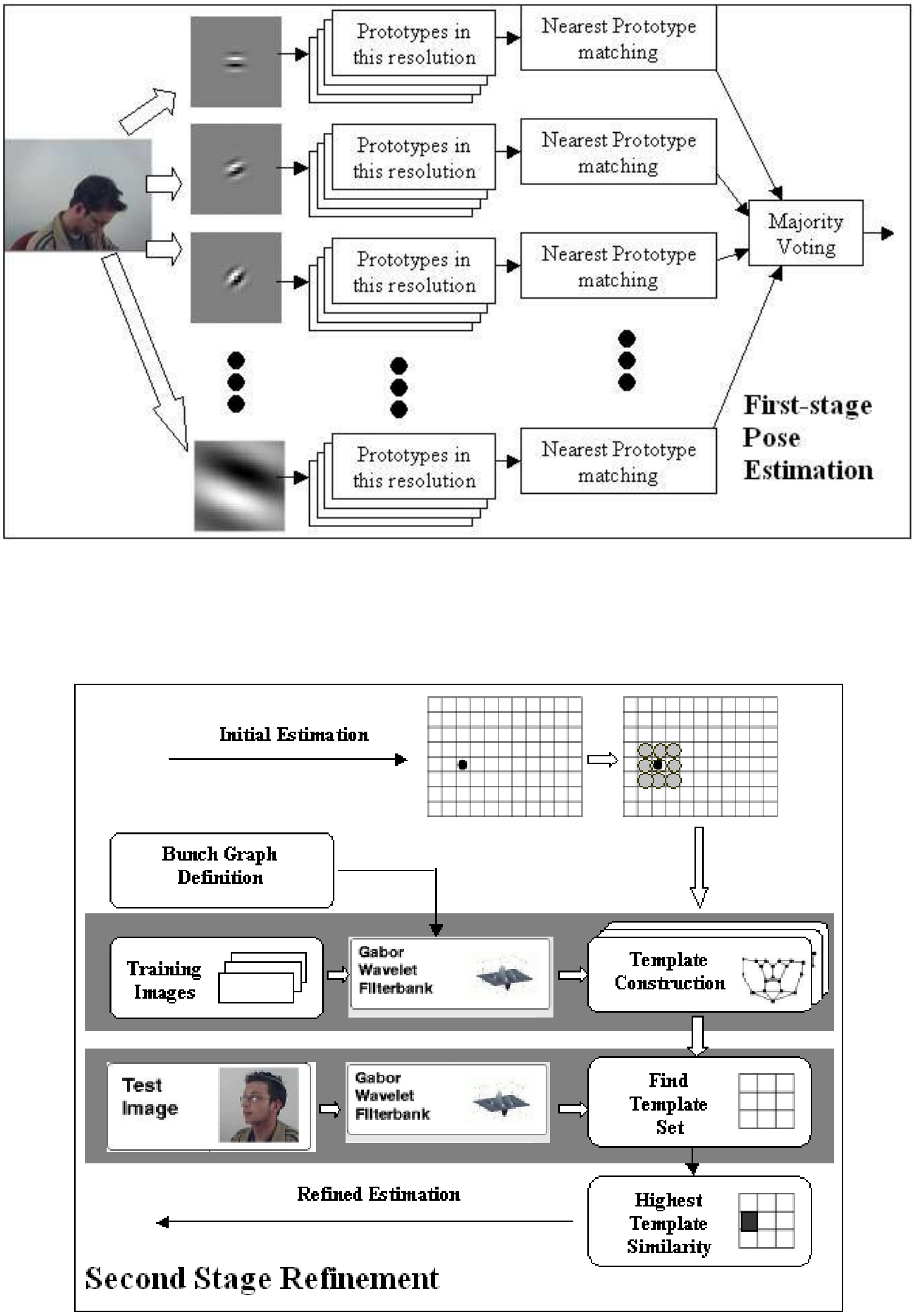

Prototypes in this resolution
Nearest Prototype matching
Prototypes in this resolution
Nearest Prototype matching
Prototypes in this resolution
Nearest Prototype matching
Majority Voting
Prototypes in this resolution
Nearest Prototype matching
First-stage Pose Estimation
Initial Estimation
Bunch Graph Definition
Training Images
Gabor Wavelet Filterbank
Template Construction
Test Image
Gabor Wavelet Filterbank
Find Template Set
Refined Estimation
Highest Template Similarity
Second Stage Refinement

Figure V.1: Flow chart of the two-stage pose estimation framework. The top diagram is for the first-stage estimation and the bottom diagram is for the second-stage refinement. The output of the first stage is the input of the second stage.

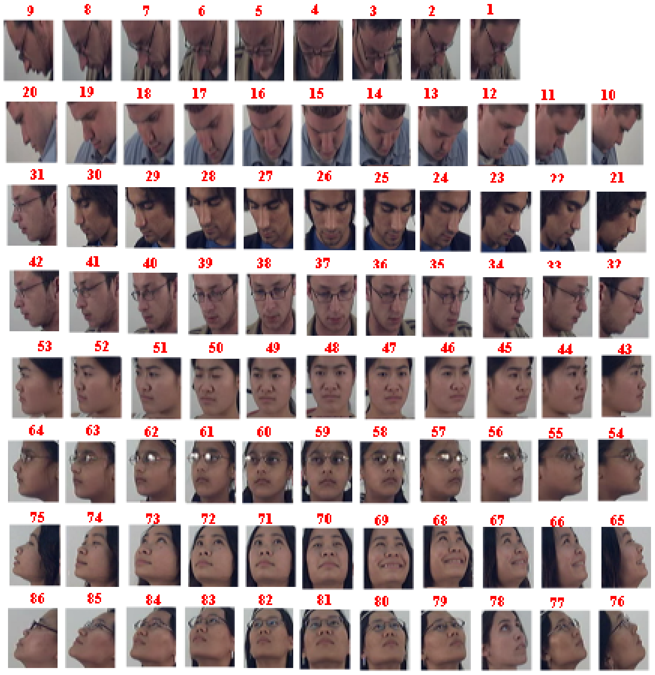


Figure V.2: Illustration for the pose labels. The top right two poses are eliminated because of a lack of samples.

V.B Stage 1: Multi-resolution Subspace Analysis

V.B.1 Feature Extraction

Frequency domain analysis techniques have a nice property in extracting the structural features as well as suppressing the undesired variations, such as the image brightness changes caused by the change of the illumination. However, frequency domain representation cannot preserve the localization information. Naturally, people will seek a joint spatial frequency representation. Gabor wavelet transform is one of such solutions. Gabor wavelets are recognized as being good feature detectors since optimal wavelets can ideally extract the positions and orientations of both global and local features [131] as well as preserving structural frequency information. Gabor wavelet transform is a convolution of the image with

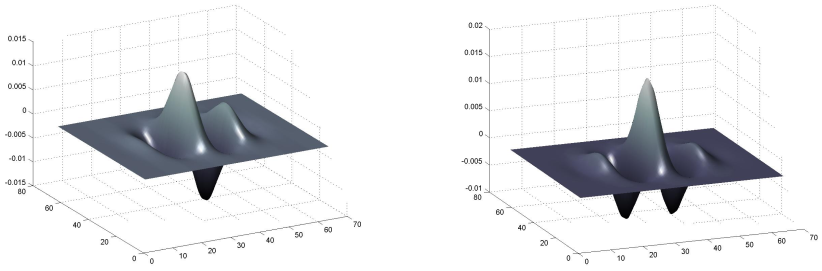


Figure V.3: The real and imaginary component of the mother wavelet.

a family of Gabor kernels. All Gabor kernels are generated by a mother wavelet through dilations and rotations. The mother wavelet is a plane wave generated from a complex exponential and restricted by a Gaussian envelop. In equation V.1-V.3, a DC-free mother wavelet is given [62, 131]:

$$\Psi_{\tilde{\mathbf{k}}}(\tilde{\mathbf{x}}) := B(k, \tilde{\mathbf{x}})[\exp(i\tilde{\mathbf{k}} \cdot \tilde{\mathbf{x}}) - \exp(-\frac{\sigma^2}{2})]; \tag{V.1}$$

$$B(k, \tilde{\mathbf{x}}) = \frac{k^2}{\sigma^2} \exp\{-\frac{k^2}{2\sigma^2}\|\tilde{\mathbf{x}}\|^2\}; \tag{V.2}$$

$$\|\Psi_{\tilde{\mathbf{k}}}(\tilde{\mathbf{x}})\|^2 \sim k^2; \tag{V.3}$$

where $B(k, \tilde{\mathbf{x}})$ is the Gaussian envelop function restricting the plane wave; $\exp(i\tilde{\mathbf{k}} \cdot \tilde{\mathbf{x}})$ is the complex-valued plane wave and $\exp(-\frac{\sigma^2}{2})$ is the DC-component. The set of Gabor kernels can be given as:

$$\Psi_{\tilde{\mathbf{k}}}(\tilde{\mathbf{x}}) = k^2 \cdot \Psi_{[1;0]}(k\mathbb{R}(\phi) \cdot \tilde{\mathbf{x}}); \tag{V.4}$$

where $\tilde{\mathbf{k}} = (k, \phi)$ is the spatial frequency in polar coordinates and

$$\mathbb{R}(\phi) = \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix}.$$

DC-free versions of Gabor kernels are invariant to image brightness [62]. We use the magnitude of the filter response as our feature representation, since the phase response is highly sensitive to mis-alignment. In our implementation, we use a family of Gabor kernels with 48 spatial frequencies (6 scales and 8 rotations). Figure V.3 gives the real part as well as the imaginary part of the mother wavelet. Examples of the magnitude of the transformed data are shown in Figure V.4.

V.B.2 Subspace Projection in Transformed Domain

The wavelet transformation serves to find the image structure in the spatial-frequency domain. However, the transformed features suffer from high dimensionality. Also, the discriminant information between classes are not extracted. Subspace projection is used to reduce the dimension as well as to extract the most essential information for classification/representation. Different subspace projection serves to find the most representative information based on different criterion. To better study the underlying structure of the data, four popular subspaces are used for data representation, and their performances are compared: PCA subspace, FDA subspace and their corresponding non-linear forms, KPCA and KDA. For the clarity of presentation, in the following sections, the data set is denoted as $\{\mathbf{x}_{c,j}\}_{j=1,\cdots,N}$, where $N = \sum_{c=1}^{C} N_c$ and $\{\mathbf{x}_i\}_{i=1,\cdots,N} = \bigcup_{c=1}^{C}\{\mathbf{x}_{c,j}\}_{j=1,\cdots,N_c}$. C is the number of classes and N_c is the number of samples in the c-th class.

PCA for Subspace Projection

PCA is a widely used method in subspace feature extraction. In Chapter IV.C.2, a review of the PCA computation is given. PCA provides a powerful linear technique for data reduction. When the data's distribution is Gaussian, PCA gives an accurate density model. That means, for the given Gaussian data, PCA identifies the axes of its Gaussian model. However, the linearity and Gaussian assumptions usually do not hold for real world data. Most interesting data in real world are non-Gaussian and assume certain non-linearities. Since PCA is a linear transformation derived from second order statistics, it is clearly beyond PCA's capabilities to extract the nonlinear structure or the higher order statistics of the feature space. This introduces Kernel PCA, which explores the non-linearity of the

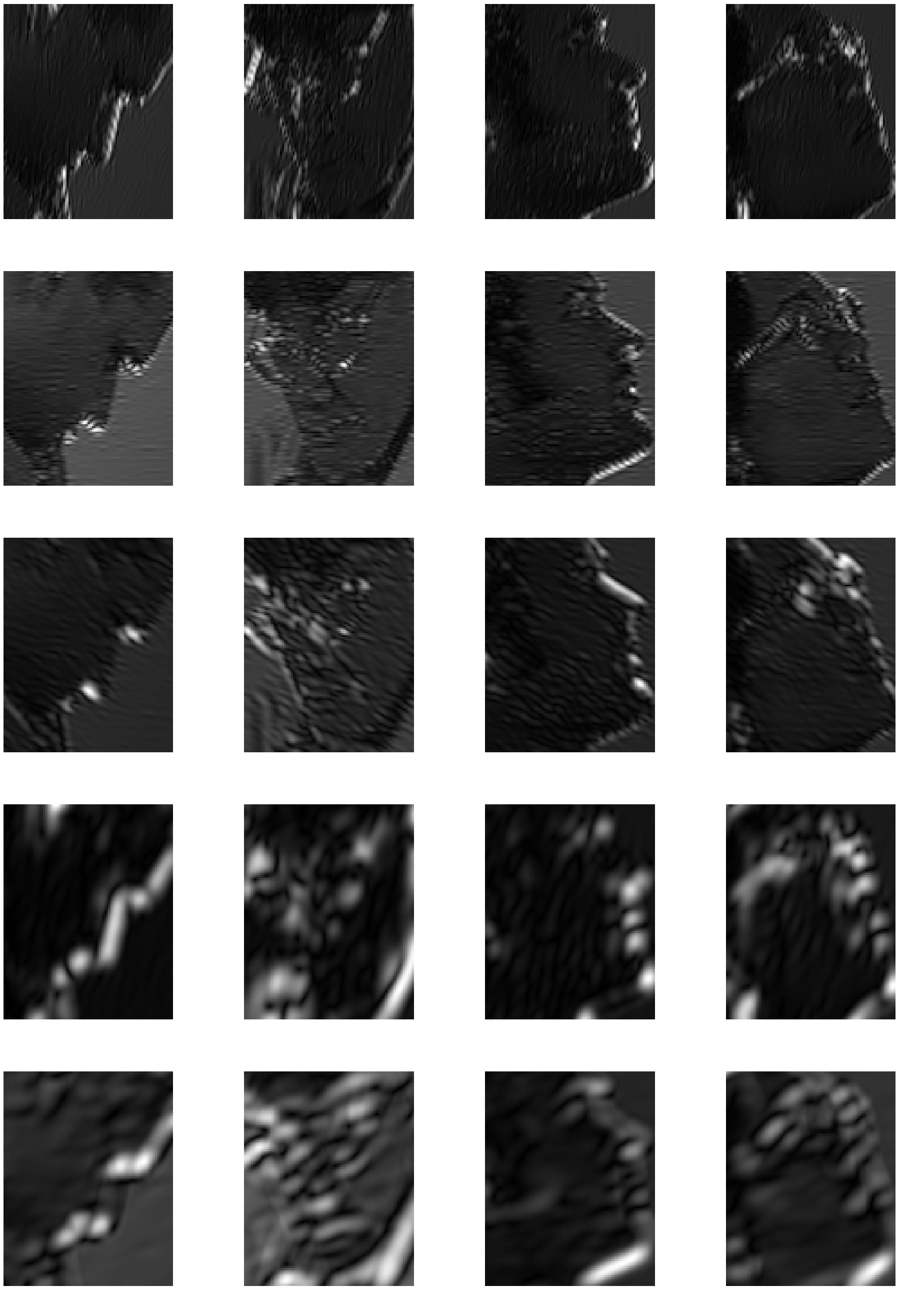


Figure V.4: Examples of the Gabor filter responses. The first row: the 1st scale angle $\frac{3\pi}{4}$; the second row: the 2nd scale, angle 0. The third row: the 3rd scale, angle $\frac{7\pi}{4}$; the fourth row: the 5th scale, angle $\frac{3\pi}{4}$. The fifth row: the 6th scale, angle $\frac{\pi}{4}$.

feature space. In [54] the authors also use Kernel PCA in modeling the multi-view faces.

KPCA for Subspace Subtraction [128]

Assuming the data has nonlinear distribution, we can map it onto a new higher dimensional feature space $\{\mathbf{\Phi}(\mathbf{x}) \in \mathcal{F}\}$ by the nonlinear mapping $\mathbf{\Phi} : \mathbf{x} \mapsto \mathbf{\Phi}(\mathbf{x})$. The desire is to get linear data in the new feature space. The nonlinear PCA representation is obtained by a linear PCA representation in the transformed feature space $\mathcal{F}$. We then perform the regular PCA in $\mathcal{F}$, which gives:

$$\mathbf{\Sigma} = \frac{1}{\sum_{c=1}^{C} N_c} \sum_{c=1}^{C} \sum_{i=1}^{N_c} (\mathbf{\Phi}(\mathbf{x}_{c,i}) - \mathbf{\Phi}(\boldsymbol{\mu}))(\mathbf{\Phi}(\mathbf{x}_{c,i}) - \mathbf{\Phi}(\boldsymbol{\mu}))^{\mathrm{T}}; \tag{V.5}$$

where:

$$\mathbf{\Phi}(\boldsymbol{\mu}) = \frac{1}{\sum_{c=1}^{C} N_c} \sum_{c=1}^{C} \sum_{i=1}^{N_c} \mathbf{\Phi}(\mathbf{x}_{c,i}). \tag{V.6}$$

Substitute Eq. V.6 into Eq. V.5, we can see that only dot product $\mathbf{\Phi}(\mathbf{x}_i) \bullet \mathbf{\Phi}(\mathbf{x}_j)$ is involved for calculating the covariance matrix:

$$\mathbf{\Sigma} = \sum_{c=1}^{C} \sum_{i=1}^{N_c} \frac{[\mathbf{\Phi}(\mathbf{x}_{c,i}) - \frac{\sum_{k=1}^{C} \sum_{j=1}^{N_k} \mathbf{\Phi}(\mathbf{x}_{k,j})}{\sum_{k=1}^{C} N_k}][\mathbf{\Phi}(\mathbf{x}_{c,i}) - \frac{\sum_{k=1}^{C} \sum_{j=1}^{N_k} \mathbf{\Phi}(\mathbf{x}_{k,j})}{\sum_{k=1}^{C} N_k}]^{\mathrm{T}}}{\sum_{c=1}^{C} N_c}; \tag{V.7}$$

Therefore, we do not need to have an explicit function for the nonlinear transform. Introduce the kernel trick, which defines the kernel as:

$$\mathcal{K}(\mathbf{x}_{c,i}; \mathbf{x}_{k,j}) \equiv \mathbf{\Phi}(\mathbf{x}_{c,i}) \bullet \mathbf{\Phi}(\mathbf{x}_{k,j}). \tag{V.8}$$

And the Gram matrix is defined as:

$$\mathbf{K} = \begin{pmatrix} \mathcal{K}(\mathbf{x}_{1,1}; \mathbf{x}_{1,1}) & \mathcal{K}(\mathbf{x}_{1,1}; \mathbf{x}_{1,2}) & \cdots & \mathcal{K}(\mathbf{x}_{1,1}; \mathbf{x}_{C,N_C}) \\ \mathcal{K}(\mathbf{x}_{2,1}; \mathbf{x}_{1,1}) & \mathcal{K}(\mathbf{x}_{2,1}; \mathbf{x}_{1,2}) & \cdots & \mathcal{K}(\mathbf{x}_{2,1}; \mathbf{x}_{C,N_C}) \\ & & \vdots & \\ \mathcal{K}(\mathbf{x}_{C,N_C}; \mathbf{x}_{1,1}) & \mathcal{K}(\mathbf{x}_{C,N_C}; \mathbf{x}_{1,2}) & \cdots & \mathcal{K}(\mathbf{x}_{C,N_C}; \mathbf{x}_{C,N_C}) \end{pmatrix} \tag{V.9}$$

Linear PCA problem in space $\mathcal{F}$ then gives:

$$\mathbf{\Sigma}\mathbf{v} = \lambda\mathbf{v}. \tag{V.10}$$

The Hilbert space assumption constrains $\mathbf{v}$'s solution space within the span of $\mathbf{\Phi}(\mathbf{x}_1), \cdots, \mathbf{\Phi}(\mathbf{x}_N)$, which means:

$$\mathbf{v} = \sum_i \alpha_i \mathbf{\Phi}(\mathbf{x}_i). \tag{V.11}$$

Let $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_N]$. The PCA problem can be reduced to the following one:

$$\mathbf{K}'\boldsymbol{\alpha} = N\lambda\boldsymbol{\alpha}. \tag{V.12}$$

$\mathbf{K}'$ is the slightly different version from $\mathbf{K}$ by including the non-zero mean of the features in the nonlinear space. It can be written as [128, 132]:

$$\mathbf{K}' = (\mathbf{I} - \mathbf{e}\mathbf{e}^{\mathrm{T}})\mathbf{K}(\mathbf{I} - \mathbf{e}\mathbf{e}^{\mathrm{T}}); \tag{V.13}$$

where $\mathbf{e} = \frac{1}{\sqrt{N}}[1, 1, \cdots, 1]^{\mathrm{T}}$.

Kernel PCA provides a way to warp the training samples from the input space to a feature space by the Gram matrix. The implicit mapping reveals certain manifold structure of the data. Hence, a better generalization ability can be achieved. In [132], the authors point out that by choosing different kernels, Kernel PCA can actually learn the manifold well through exploring the local data structure. The eigen-decomposition of the Gram matrix provides an embedding that captures the low-dimensional structure of the manifold. In our implementation, we use the traditional Gaussian kernel.

However, it is not clear if the statistical structure captured by PCA/KPCA is also good for classification, even in the kernel space. It is because the first principals will probably not reveal the most discriminating structure of the underlying class information. This inspires us to pursue FDA as an alternative, which is the multiple-class version of fisher discriminant analysis [88].

FDA for Subspace Projection

While PCA seeks a projection subspace that can achieve a more compact representation of the data, discriminant analysis seeks a projection subspace that is efficient for discrimation. The basic idea is to find a projection that can make the data from one class as compact as possible while separate the data from different class as much as possible. The same as binary classification problem, for multiple-class problem, the distance of samples within class is described by the within-class scatter matrix:

$$\mathbf{S}_W = \sum_{c=1}^{C}\sum_{i=1}^{N_c}(\mathbf{x}_{c,i} - \boldsymbol{\mu}_c)(\mathbf{x}_{c,i} - \boldsymbol{\mu}_c)^{\mathrm{T}}; \tag{V.14}$$

where

$$\boldsymbol{\mu}_c = \frac{1}{N_c}\sum_{i=1}^{N_c}\mathbf{x}_{c,i}.$$

The distance of samples between classes is described by the between-class scatter matrix:

$$\mathbf{S}_B = \sum_{c=1}^{C} N_c(\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^{\mathrm{T}}; \tag{V.15}$$

where

$$\boldsymbol{\mu} = \frac{1}{\sum_{c=1}^{C} N_c}\sum_{c=1}^{C}\sum_{i=1}^{N_c}\mathbf{x}_{c,i}.$$

.

Projection **W** is found by fisher's criterion, which maximize the Raleigh coefficient:

$$\mathcal{J}(\mathbf{W}) = \frac{\mathbf{W}^{\mathrm{T}}\mathbf{S}_B\mathbf{W}}{\mathbf{W}^{\mathrm{T}}\mathbf{S}_W\mathbf{W}}. \tag{V.16}$$

This turns out to be an eigen-decomposition problem. The solution can be found by:

$$\mathbf{S}_B\mathbf{w}_i = \lambda_i\mathbf{S}_W\mathbf{w}_i. \tag{V.17}$$

It can also be written as:

$$\mathbf{S}_W^{-1}\mathbf{S}_B\mathbf{w}_i = \lambda_i\mathbf{w}_i. \tag{V.18}$$

Similar as PCA, FDA is a linear transformation and cannot discover the nonlinear structure. We use the extended nonlinear version of discriminant analysis, KDA [129], to better explore the discriminant information.

KDA for Subspace Projection

The same as Kernel PCA, KDA processes data in the non-linearly transformed space. A nonlinear function $\mathbf{\Phi}$ maps the data $\mathbf{x}$ from the original space $\Re^n$ into the feature space $\mathcal{F}$, where the data present to be linearly distributed. The projection subspace is constrained in the span of the transformed samples by the Hilbert space assumption. The k-th projection direction is:

$$\mathbf{w}_k = \sum_{i=1}^{N} \alpha_i^{(k)} \mathbf{\Phi}(\mathbf{x}_i). \tag{V.19}$$

The sample mean becomes:

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{\Phi}(\mathbf{x}_i). \tag{V.20}$$

Then we have:

$$\mathbf{w}_k^{\mathrm{T}} \boldsymbol{\mu} = \frac{1}{N} \sum_{j=1}^{N} \sum_{i=1}^{N} \alpha_j^{(k)} \mathbf{\Phi}(\mathbf{x}_j)^{\mathrm{T}} \mathbf{\Phi}(\mathbf{x}_i). \tag{V.21}$$

Similarly the class mean vector $\boldsymbol{\mu}_c$ now becomes:

$$\boldsymbol{\mu}_c = \frac{1}{N_c} \sum_{i=1}^{N_c} \mathbf{\Phi}(\mathbf{x}_{c,i}). \tag{V.22}$$

and:

$$\mathbf{w}_k^{\mathrm{T}} \boldsymbol{\mu}_c = \frac{1}{N_c} \sum_{j=1}^{N} \sum_{i=1}^{N_c} \alpha_j^{(k)} \mathbf{\Phi}(\mathbf{x}_j)^{\mathrm{T}} \mathbf{\Phi}(\mathbf{x}_{c,i}). \tag{V.23}$$

Introduce the kernel trick again, which defines the kernel as the dot product of the data in the feature space:

$$\mathcal{K}(\mathbf{x}_i; \mathbf{x}_j) = \mathbf{\Phi}(\mathbf{x}_i) \bullet \mathbf{\Phi}(\mathbf{x}_j).$$

Define the identity vectors $\mathbf{1} \in \Re^N$ as $\mathbf{1} = [1, 1, \cdots, 1]^{\mathrm{T}}$ and $\mathbf{1}_c \in \Re^{N_c}$ as $\mathbf{1}_c = [1, 1, \cdots, 1]^{\mathrm{T}}$. Eq. V.21 and Eq. V.23 can be written as:

$$\mathbf{w}_k^{\mathrm{T}} \boldsymbol{\mu} = \boldsymbol{\alpha}_k^{\mathrm{T}} \mathbf{K} \mathbf{1},$$

and

$$\mathbf{w}_k^{\mathrm{T}} \boldsymbol{\mu}_c = \boldsymbol{\alpha}_k^{\mathrm{T}} \mathbf{K}_c \mathbf{1}_c,$$

where $\boldsymbol{\alpha}_k = [\alpha_1^{(k)}, \cdots, \alpha_N^{(k)}]^{\mathrm{T}}$.

The $N \times N$ matrix $\mathbf{K}$ is the Gram matrix as defined in Eq. V.9; and the $N \times N_c$ matrix $\mathbf{K}_c$ is defined as below:

$$\mathbf{K}_c = \begin{pmatrix} \mathcal{K}(\mathbf{x}_1; \mathbf{x}_{c,1}) & \mathcal{K}(\mathbf{x}_1; \mathbf{x}_{c,2}) & \cdots & \mathcal{K}(\mathbf{x}_1; \mathbf{x}_{c,N_c}) \\ \mathcal{K}(\mathbf{x}_2; \mathbf{x}_{c,1}) & \mathcal{K}(\mathbf{x}_2; \mathbf{x}_{c,2}) & \cdots & \mathcal{K}(\mathbf{x}_2; \mathbf{x}_{c,N_c}) \\ & & \vdots & \\ \mathcal{K}(\mathbf{x}_N; \mathbf{x}_{c,1}) & \mathcal{K}(\mathbf{x}_N; \mathbf{x}_{c,2}) & \cdots & \mathcal{K}(\mathbf{x}_N; \mathbf{x}_{c,N_c}) \end{pmatrix}.$$

After derivation, the between class scatter matrix can be represented by:

$$\mathbf{W}^{\mathrm{T}}\mathbf{S}_B\mathbf{W} = \mathbf{W}^{\mathrm{T}} \sum_{c=1}^{C} N_c(\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^{\mathrm{T}}\mathbf{W} = \mathbf{U}^{\mathrm{T}}(\sum_{c=1}^{C} \frac{\mathbf{K}_c\mathbb{I}_c\mathbf{K}_c^{\mathrm{T}}}{N_c} - \frac{\mathbf{K}\mathbb{I}\mathbf{K}}{N})\mathbf{U}; \tag{V.24}$$

where $\mathbb{I} = \mathbf{1}\mathbf{1}^{\mathrm{T}}$ is an $N \times N$ matrix with all 1 entries and $\mathbb{I}_c = \mathbf{1}_c\mathbf{1}_c^{\mathrm{T}}$ is an $N_c \times N_c$ matrix with all 1 entries. The within class scatter matrix $\mathbf{S}_W$ can be represented by:

$$\mathbf{W}^{\mathrm{T}}\mathbf{S}_W\mathbf{W} = \sum_{c=1}^{C}\sum_{i=1}^{N_c}(\mathbf{x}_{c,i} - \boldsymbol{\mu}_c)(\mathbf{x}_{c,i} - \boldsymbol{\mu}_c)^{\mathrm{T}} = \mathbf{U}^{\mathrm{T}}(\sum_{c=1}^{C}\mathbf{K}_c^{\mathrm{T}}\mathbf{K}_c - \sum_{c=1}^{C}\frac{\mathbf{K}_c\mathbb{I}_c\mathbf{K}_c^{\mathrm{T}}}{N_c})\mathbf{U}. \tag{V.25}$$

The Raleigh's coefficient now becomes:

$$\mathcal{J}(\mathbf{U}) = \frac{\mathbf{U}^{\mathrm{T}}(\sum_{c=1}^{C}\frac{1}{N_c}\mathbf{K}_c\mathbb{I}_c\mathbf{K}_c^{\mathrm{T}} - \frac{1}{N}\mathbf{K}\mathbb{I}\mathbf{K})\mathbf{U}}{\mathbf{U}^{\mathrm{T}}(\sum_{c=1}^{C}\mathbf{K}_c^{\mathrm{T}}\mathbf{K}_c - \sum_{c=1}^{C}\frac{1}{N_c}\mathbf{K}_c\mathbb{I}_c\mathbf{K}_c^{\mathrm{T}})\mathbf{U}}. \tag{V.26}$$

The new projection is pursued by finding $\mathbf{U} = [\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_m]$ that maximizes the Raleigh's coefficient. Similar to the linear fisher discriminant analysis, the solution can be found by the eigen-decomposition:

$$(\sum_{c=1}^{C}\mathbf{K}_c^{\mathrm{T}}\mathbf{K}_c - \sum_{c=1}^{C}\frac{1}{N_c}\mathbf{K}_c\mathbb{I}_c\mathbf{K}_c^{\mathrm{T}})^{-1}(\sum_{c=1}^{C}\frac{1}{N_c}\mathbf{K}_c\mathbb{I}_c\mathbf{K}_c^{\mathrm{T}} - \frac{1}{N}\mathbf{K}\mathbb{I}\mathbf{K})\boldsymbol{\alpha}_i = \lambda_i\boldsymbol{\alpha}_i. \tag{V.27}$$

In Figure V.5 and Figure V.6 the property of different subspace projections are illustrated by 2D toy examples. In Figure V.5, the original 2D data are projected

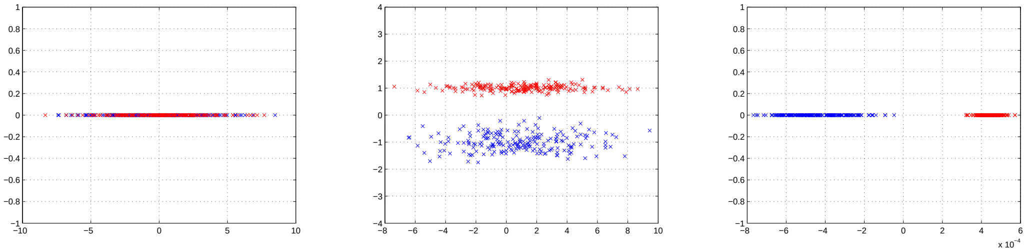


Figure V.5: Example of PCA and FDA subspace representation. The leftmost image: the original 2D data. The middle image: the projected data from PCA subspace. The rightmost image: the projected data from FDA subspace.

onto the 1D PCA subspace as well as the 1D FDA subspace. PCA subspace is chosen as the direction along the eigenvector that has the largest eigenvalue. The projected results are shown in the figure. In the FDA subspace, the projected data from different classes are well-separated, while in the PCA subspace the projected data are still mixed together. This illustrates that although PCA is efficient in compact data representation, it is not as powerful in classification. In Figure V.6, we compare the separation abilities for these four subspaces on nonlinear data set. PCA, KPCA, FDA and KDA subspace projections are shown for a binary 2D toy data set. As can be seen, PCA and FDA are not able to produce a more discriminating representations due to the non-linearity of the data, whereas the KPCA and KDA transform the data into two well-separated clusters. For both the above toy example and the real pose data, we use the following Gaussian function as the kernel:

$$\hat{\mathbf{x}}_i = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|}, i = 1, 2;$$

$$\mathcal{K}(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2) = \exp\{-\frac{\|\hat{\mathbf{x}}_1 - \hat{\mathbf{x}}_2\|^2}{2}\}. \tag{V.28}$$

V.B.3 Classification by Nearest Prototype Matching

We use the nearest prototype matching for the first stage classification. Each pose is represented by a set of subspaces, where each subspaces is computed from filter response in one resolution. In each subspace the class mean is used as a

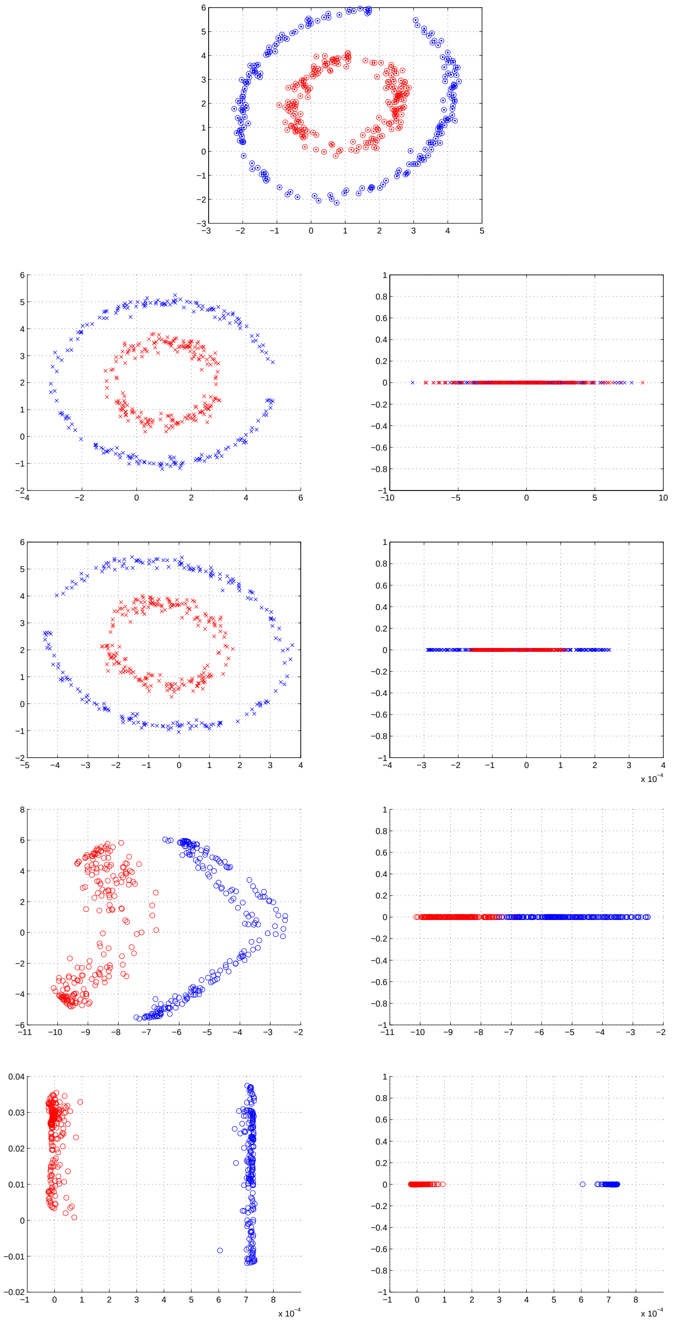


Figure V.6: Examples of the subspace representation (nonlinear data). The first row: original data. Left images in row 2-5: the projection (left to right: PCA, FDA, KPCA, KDA). Right images in row 2-5: the projected low-dimensional data.

prototype. Every pose is modeled by a number of prototypes. We use the Euclidean distances to measure the similarity to the prototypes. The pose estimate is given by the prevailing class label from all resolutions as illustration in Figure V.1. Given the k-th projection subspace as $\mathbf{U}_k$, for each pose, the prototype in this resolution is given by:

$$\mathbf{m}_c^{(k)} = \frac{1}{N_c}\sum_{i=1}^{N_c}\mathbf{U}_k\mathbf{x}_{c,i}; k = 1, 2, \cdots, 48, c = 1, \cdots, C. \tag{V.29}$$

Therefore, for a testing sample $\mathbf{x}$, its class label in the k-th resolution is given by:

$$\begin{aligned} d_k &= \|\mathbf{y}^{(k)} - \mathbf{m}_c^{(k)}\|; \\ \mathrm{q}(c) &= \sum_{k=1}^{48}\delta(d_k = c); \\ \mathcal{L} &= \max_c \mathrm{q}(c); \end{aligned} \tag{V.30}$$

where $\mathbf{y}^{(k)}$ is the testing sample's projection in the k-th subspace and $\delta(d_k = c) = 1$ if $d_k = c$; otherwise $\delta(d_k = c) = 0$.

For non-perfect data, such as data with misalignment, the holistic features discussed in the first stage are not sufficient to get a high-accuracy estimate. Therefore, we only trust the pose estimated from this first stage up to a $p \times p$ ($p = 3$ in our application) neighborhood, which means if a pose is determined to have i-th pan angle and j-th tilt angle, we have a high confidence that the true pose will be located in the region that pan angle $\in [i-1, i+1]$ and tilt angle $\in [j-1, j+1]$, as explained by the example in Figure V.7. Instead of giving a final solution, the first-stage estimation only provides a range of possible solutions. This step introduces certain tolerance to the data noise, such as the noise from mis-alignment. A second-stage is applied thereafter to solve the sub-problem, where only poses in the subset of the solution space are tackled.

Although reported good performance has been achieved using Euclidean distance as the distance measure [133, 134], it is also worth noting that some other distance measure can also be used to benefit the algorithm. As indicated

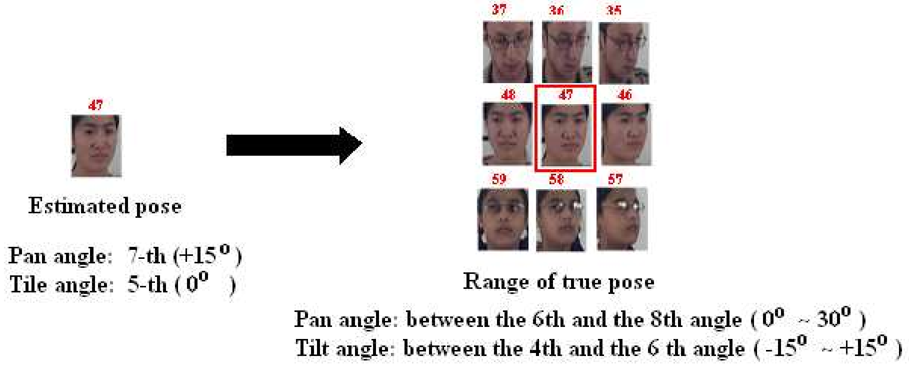

37
36
35
47
48
47
46
59
58
57
Estimated pose
Pan angle: 7-th (+15°)
Tile angle: 5-th (0°)
Range of true pose
Pan angle: between the 6th and the 8th angle (0° ~ 30°)
Tilt angle: between the 4th and the 6 th angle (-15° ~ +15°)

Figure V.7: Example of interpreting the first-stage estimation.

by [133, 135], other distance measures, such as cosine distance measure, are also used to get a better performance.

V.C Stage 2: Structural Landmark Analysis

The second stage serves to refine the coarse pose estimation. In this section, we use a revised version of the face bunch graph introduced in [62] for this purpose. The face graph is a labeled graph which connects the local image feature together with the image's geometric configuration. It exploits the local salient features on a human face, e.g. pupils, nose tip, corners of mouth, and etc. together with their locations. The motivation behind the use of geometric configuration of salient points on a face lies in an observation that with different degrees of change in poses, the relative locations between salient points correspondingly change.

V.C.1 Bunch Graph Construction

Each face image constructs a model graph. The model graph is a labeled graph with its nodes corresponding to the Gabor jets at the predefined salient facial features, and its edges labeled by the distance vectors between the nodes. A Gabor jet is the concatenation of the Gabor wavelet responses at a single image

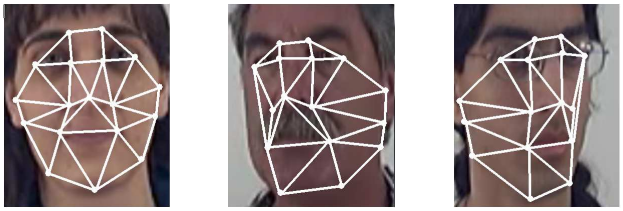


Figure V.8: Examples of the model face graph. The leftmost one is for the frontal view. The middle one is for the pose with 0^o tilt angle and -15^o pan angle. The rightmost one is for the pose with tilt angle 0^o and pan angle $+15^o$.

point. Only the magnitude of the filter response is used. Some examples of the model graphs are show in Figure V.8. Occlusion of the salient features in the current view determines how many nodes are used. More nodes assert more geometric constraints, which is useful for pose discriminating; however, more identity information could be preserved as well, which is not desired.

Each pose is represented by one set of bunch graphs from the model graphs of the same pose. The nodes of the face bunch graph are the bundles of the corresponding nodes in the model graphs. Subjects from different races, age groups and different genders possess different geometric configurations, hence the geometric structure is not subject-independent. A single bunch graph does not have a good generalization ability in terms of the topographic property. Therefore, a single bunch graph is not sufficient to model all subjects. Although a simple average of all the available geometric configurations followed by an exhaustive search and match can still be used to find the identity-related fiducial points, this would also add ambiguities to the global structure between close poses. For the purpose of retrieving the pose information while suppressing the subject identity, we keep the geometric configurations for all training samples under the same pose and use a *semi-rigid* search for matching, which means only local adjustment is allowed to refine the estimated face graph. Therefore, for each pose we actually have the same number of bunch graphs as the model graphs. Each bunch graph inherits the edge information from an individual model graph, while the bunch graphs for the same pose only differ in the edge labels. This is illustrated in Figure V.9. This strategy is a trade-off between identity and identity-independent poses. It enables us to avoid large distortion in geometric structure that causes ambiguities between neighboring poses. The set of bunch graphs for each pose are constructed offline, and they are used as the templates for matching.

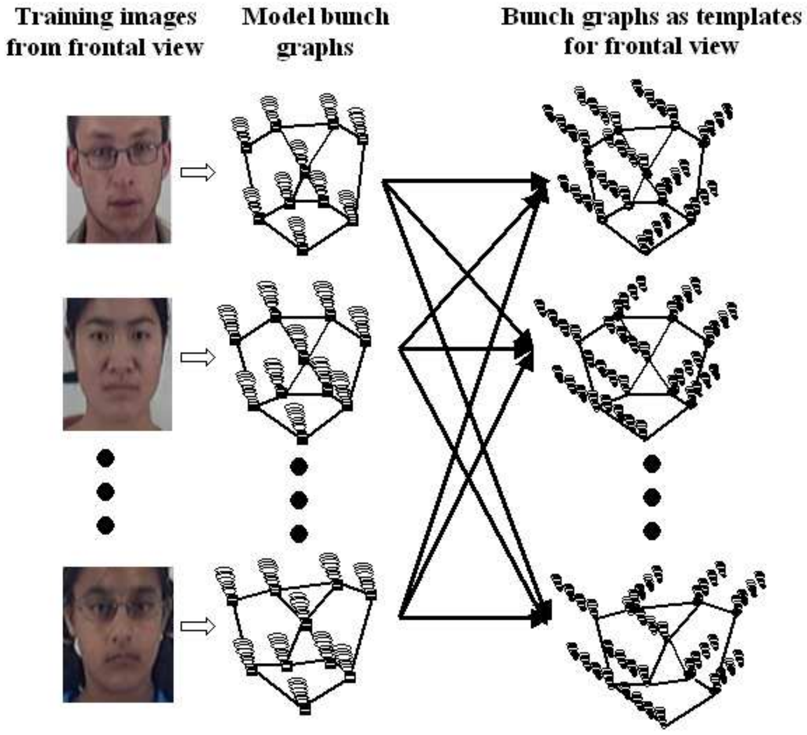

Training images from frontal view
Model bunch graphs
Bunch graphs as templates for frontal view

Figure V.9: Construction of the bunch graphs as the template for a single pose. Here we use the pose with frontal view, whose pan angle is 0° and tilt angle is 0°. This figure is for illustration only and more nodes are used for actual graph construction.

V.C.2 Similarity Measurement and Graph Matching

Denote the subset of poses confined by the first stage estimation as $\mathcal{P}_s$. Given a test image, every pose candidate in $\mathcal{P}_s$ gives an estimated face graph by searching the sets of nodes that maximize the graph similarity. Graph similarity is determined both by the similarity of the nodes and the distance in edge labels. We use the normalized cross correlation as the nodes similarity metric [62]. Let $\mathbf{J}(i) = (f_1(i), \cdots, f_F(i))$ be the Gabor jet for i-th nodes; where F is the number of the Gabor filters. Nodes similarity D is given by the normalized cross correlation, which is:

$$D(\mathbf{J}(i); \mathbf{J}(k)) = \frac{\sum_{m=1}^{F} f_m(i) f_m(k)}{\sqrt{\sum_{m=1}^{F} f_m^2(i) \sum_{m=1}^{F} f_m^2(k)}}. \tag{V.31}$$

The graph similarity S between the estimated face graph $\mathcal{G} = (\mathbf{J}_m, \delta_e)$ and some bunch graph $\mathcal{B} = (\{\mathbf{J}_m^{B_i}\}_i, \delta_e^B)$ is defined as:

$$\mathcal{S}(\mathcal{G}, \mathcal{B}) = \frac{1}{M} \sum_{m=1}^{M} \max_i (D(\mathbf{J}_m; \mathbf{J}_m^{B_i})) - \frac{\lambda}{E} \sum_{e=1}^{E} \frac{(\delta_e - \delta_e^B)^2}{(\delta_e^B)^2}; \tag{V.32}$$

where λ is the relaxation factor and it determines the relative importance of the topography term.

For a certain head pose in $\mathcal{P}_s$, its corresponding set of bunch graph templates can determine a best matched face graph. Since we have multiple bunch graphs, each of them can generate a possible face graph. The best matched one needs to be found as the representative face graph for this pose, which is given in Algorithm 4.

It is worth noting that by constraining the bunch graph to be semi-rigid, a trade-off is made between the ability to automatically locate facial nodes and the ability to preserve the geometric constraint that confines the head poses. Jet similarity plays a crucial role in identifying the initial guess of the geometric constraint. However, although Gabor jets perform well in identifying similar facial features, it is not sufficient for distinguishing ambiguous features. By using the two-stage framework, the first stage limits the node search space such that the possibility of introducing ambiguous matching is lowered. Therefore the proposed two-stage algorithm not only helps reduce the computational cost, also it is important for performance gain.

Algorithm 4 Face graph matching for refining the estimated head pose.

Step 1: Scan the testing image. For every bunch graph template, fix the geometric constraint, which means $\lambda = \infty$. Use the graph similarity to find a set of points that has a best match with the nodes from the bunch graph. Since for each pose, the bunch graph templates only differ in the edge information, we are actually searching for the set of matching points under different geometric constraints. Each bunch graph template gives a set of matching points. The set of matching points together with their relative locations, which are the same as those from the rigid geometric constraint from the corresponding bunch graph, determines one face graph; whose nodes are Gabor Jets of the set of matching points and the edges are the relative distance between nodes. Therefore, for each bunch graph template, a face graph is obtained. Their matching score is computed and the face graph that gives the highest matching score is the final face graph estimate, which is:

$$t^{\star} = \arg\max_{t} \mathcal{S}(\mathcal{G}_t, \mathcal{B}_t), \text{with} \lambda = \infty. \tag{V.33}$$

Step 2: Relax the geometric constraint now by locally adjust the node positions of the estimated face graph $\mathcal{G}_{t^{\star}}$. Compare the matching score after each adjustment until a best match is obtained. The new set of points and their relative locations are recorded.

Step 3: The refined points and their relative positions from the second step determine the updated face graph; whose nodes are Gabor Jets of the set of updated points and the edges are the relative distance between the new nodes. Recompute the matching score for this updated face graph using the graph similarity definition, this gives the similarity of the pose in the test image to the current pose.

Step 4: Compare the scores from estimated face graphs for different poses in the subset determined by the first stage, the pose that has the highest similarity gives the final pose estimate.

V.D Experimental Evaluation and Analysis

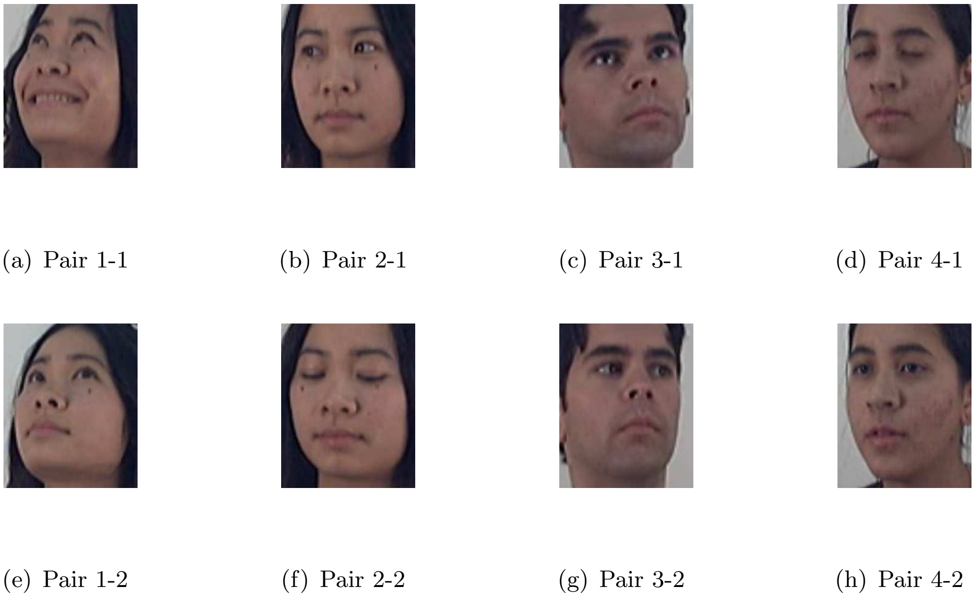
We use a data set that contains 28 subjects to evaluate this approach, where for the same subject the subtle variations in poses and the changes in facial expressions are also considered. The subjects poses are quantized into 86 classes. The pan angle spans from -90^o to $+90^o$; with 15^o intervals from -60^o to 60^o, and then the poses with 90^o pan angles are also considered. The tilt angle has a consistent interval of 15^o from -45^o to 60^o. A magnetic sensor is used to provide the ground-truth information. The magnetic sensor can be calibrated to have the same reference frame as the video camera.

The face images are prepared by the following way. We use the output from Viola and Jones face detectors [44] to crop the face region for head pose estimation. Individual face detectors were trained for different head poses; altogether 9 face detectors were used. However, since the face detector is tuned to a specific view, for each sample, we need to manually determine which face detector to use for cropping the face region. 3894 images of size 67×55 and their mirror images are used, so altogether there are 7788 images. Each pose has $80 \sim 100$ samples, randomly split into two parts at the subject level, one for training and one for testing. . That is to say, for each pose, the training set and the testing set contains disjoint subjects. Each subject may contribute several samples. The reason is that although for pose classification problem, the main challenge is the generalization ability for handling different subjects, the algorithm should also have the ability to tolerate the variations from the same subject, such as the variations caused by subtle pose changes and the variations from the facial expressions. In Figure V.10, examples are shown to illustrate this. Every column gives a pair of images, which are from the same subject in the same pose class. Both images are used in the data set due to the sufficient variation in poses or facial expression changes.

V.D.1 Stage 1: “Coarse” Pose Estimation

Output of the first stage is a $p \times p$ subset of poses. The accuracy is evaluated accordingly: if the true pose does not belong to this subset, it is counted as a false estimate. In our implement $p = 3$ is used if not specially stated. Greater p gives better accuracy, however, more computational cost will result for the second stage refinement. In Figure V.11-Figure V.12, the first stage accuracy from the different four subspace methods are presented. The total accuracy is summarized in table Table V.1.

As expected, the kernel based subspaces can provide a higher accuracy;

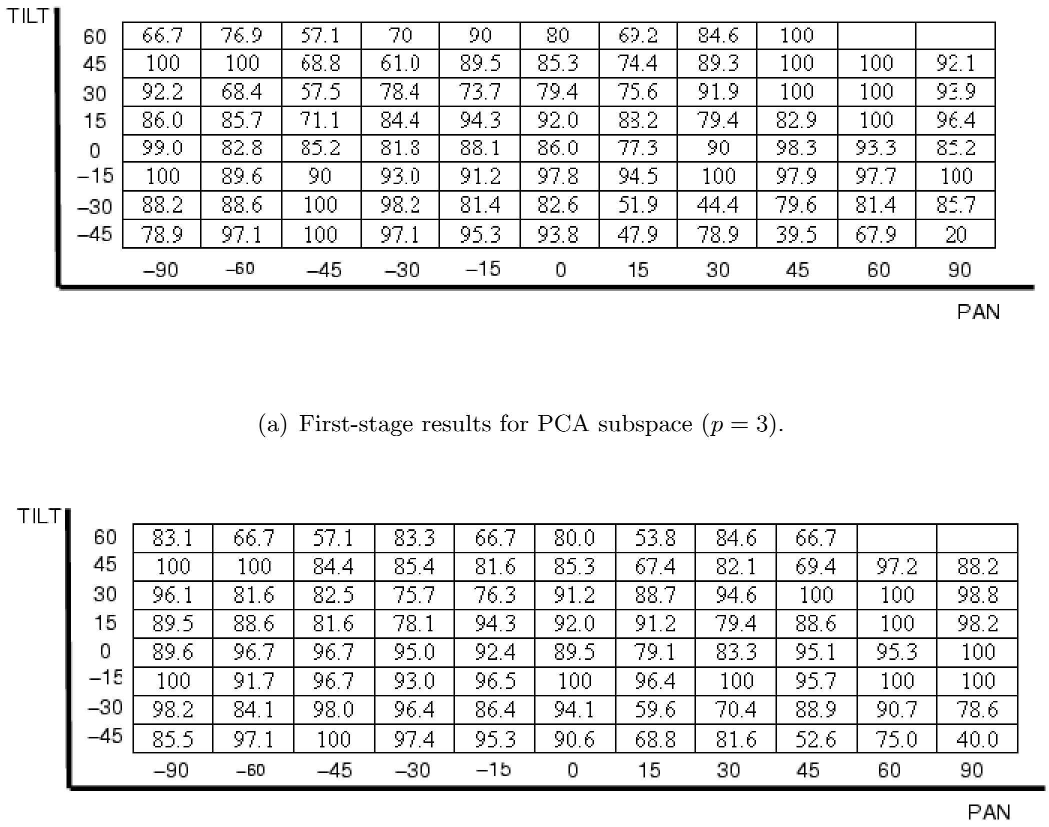


(a) Pair 1-1
(b) Pair 2-1
(c) Pair 3-1
(d) Pair 4-1
(e) Pair 1-2
(f) Pair 2-2
(g) Pair 3-2
(h) Pair 4-2

Figure V.10: Examples of variations for the same subject in same pose class. Each column shows images from the same pose, with subtle variation in poses and facial expressions.

also the discriminant analysis performs slightly better than the principal component analysis. If p = 1, which means only subspace analysis is used alone, the experimental results show that all four subspace did not give a satisfactory results comparable with those reported. This is not a surprise, since the subspace analysis is very sensitive to the data noise, such as background and data alignment. In our data set, the face position is not well-aligned. Also, hair and shoulder becomes background noises since the mis-alignment causes it appear in some images while not in the other, even for the same pose. In such case, the subspace analysis alone is not capable to obtain as good performance.

To better present the error distribution over different poses, in Figure V.13 we use a grayscale coded error distribution diagram to show the accuracy for each pose for KDA subspace (evaluated under p = 3). Darker color shows more error. The color coded error distribution diagram shows that the error rate is bigger when a person looks down. It is consistent with the intuition, since when a person



TILT											
60	66.7	76.9	57.1	70	90	80	69.2	84.6	100		
45	100	100	68.8	61.0	89.5	85.3	74.4	89.3	100	100	92.1
30	92.2	68.4	57.5	78.4	73.7	79.4	75.6	91.9	100	100	93.9
15	86.0	85.7	71.1	84.4	94.3	92.0	83.2	79.4	82.9	100	96.4
0	99.0	82.8	85.2	81.8	88.1	86.0	77.3	90	98.3	93.3	85.2
−15	100	89.6	90	93.0	91.2	97.8	94.5	100	97.9	97.7	100
−30	88.2	88.6	100	98.2	81.4	82.6	51.9	44.4	79.6	81.4	85.7
−45	78.9	97.1	100	97.1	95.3	93.8	47.9	78.9	39.5	67.9	20
	−90	−60	−45	−30	−15	0	15	30	45	60	90

PAN

(a) First-stage results for PCA subspace ($p = 3$).

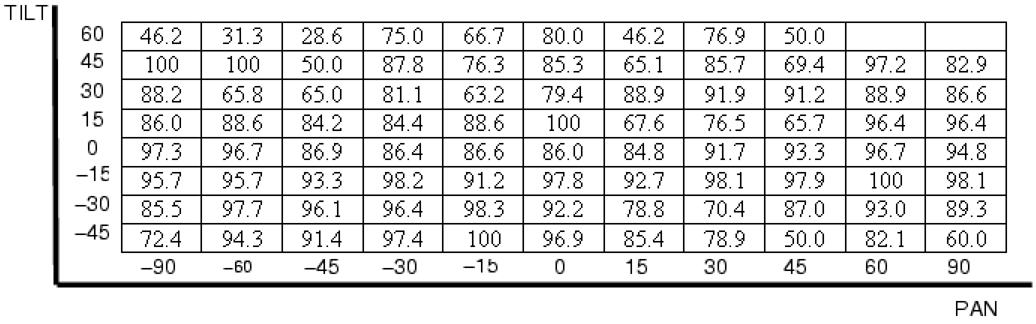
TILT											
60	83.1	66.7	57.1	83.3	66.7	80.0	53.8	84.6	66.7		
45	100	100	84.4	85.4	81.6	85.3	67.4	82.1	69.4	97.2	88.2
30	96.1	81.6	82.5	75.7	76.3	91.2	88.7	94.6	100	100	98.8
15	89.5	88.6	81.6	78.1	94.3	92.0	91.2	79.4	88.6	100	98.2
0	89.6	96.7	96.7	95.0	92.4	89.5	79.1	83.3	95.1	95.3	100
−15	100	91.7	96.7	93.0	96.5	100	96.4	100	95.7	100	100
−30	98.2	84.1	98.0	96.4	86.4	94.1	59.6	70.4	88.9	90.7	78.6
−45	85.5	97.1	100	97.4	95.3	90.6	68.8	81.6	52.6	75.0	40.0
	−90	−60	−45	−30	−15	0	15	30	45	60	90

PAN

(b) First-stage results for KPCA subspace with Gaussian kernel($p = 3$).

Figure V.11: First-stage classification accuracy for PCA/KPCA subspaces ($p = 3$)

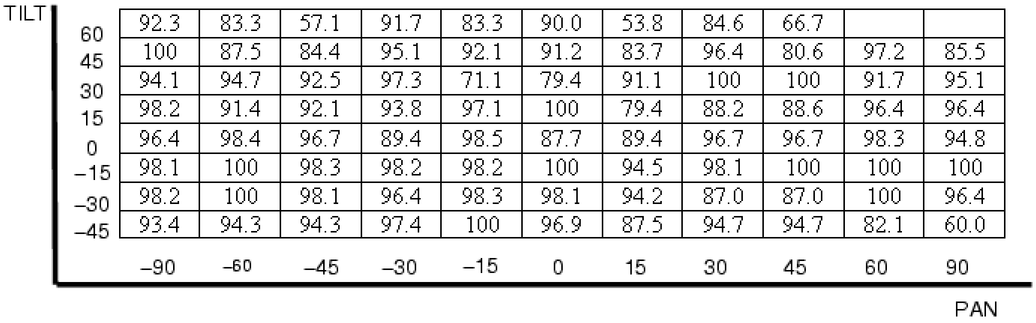
looks down, the hairline would increase the noise level, also the facial features are less visible. The use of the two-stage framework is more robust than the subspace alone, since the use of geometric configuration can get rid of the ambiguity from data noise. More experiments validate the advantage of the two-stage framework. We purposely translate the cropping window for the testing face images by $\pm 2; \pm 4; \pm 6; \pm 8; \pm 16$ pixels in both directions, which aggravates the misalignment. Use the same KDA subspace obtained in previous step to test the performance. The accuracy is evaluated for both $p = 1$ and $p = 3$, as show in Figure V.14. Experimental results indicate that when using $p = 3$ to evaluate the accuracy, the accuracy is actually quite stable with the aggravating misalignment. However,



TILT	−90	−60	−45	−30	−15	0	15	30	45	60	90
60	46.2	31.3	28.6	75.0	66.7	80.0	46.2	76.9	50.0		
45	100	100	50.0	87.8	76.3	85.3	65.1	85.7	69.4	97.2	82.9
30	88.2	65.8	65.0	81.1	63.2	79.4	88.9	91.9	91.2	88.9	86.6
15	86.0	88.6	84.2	84.4	88.6	100	67.6	76.5	65.7	96.4	96.4
0	97.3	96.7	86.9	86.4	86.6	86.0	84.8	91.7	93.3	96.7	94.8
−15	95.7	95.7	93.3	98.2	91.2	97.8	92.7	98.1	97.9	100	98.1
−30	85.5	97.7	96.1	96.4	98.3	92.2	78.8	70.4	87.0	93.0	89.3
−45	72.4	94.3	91.4	97.4	100	96.9	85.4	78.9	50.0	82.1	60.0

PAN

(a) First-stage results for FDA subspace ($p = 3$).



TILT	−90	−60	−45	−30	−15	0	15	30	45	60	90
60	92.3	83.3	57.1	91.7	83.3	90.0	53.8	84.6	66.7		
45	100	87.5	84.4	95.1	92.1	91.2	83.7	96.4	80.6	97.2	85.5
30	94.1	94.7	92.5	97.3	71.1	79.4	91.1	100	100	91.7	95.1
15	98.2	91.4	92.1	93.8	97.1	100	79.4	88.2	88.6	96.4	96.4
0	96.4	98.4	96.7	89.4	98.5	87.7	89.4	96.7	96.7	98.3	94.8
−15	98.1	100	98.3	98.2	98.2	100	94.5	98.1	100	100	100
−30	98.2	100	98.1	96.4	98.3	98.1	94.2	87.0	87.0	100	96.4
−45	93.4	94.3	94.3	97.4	100	96.9	87.5	94.7	94.7	82.1	60.0

PAN

(b) First-stage results for KDA subspace with Gaussian kernel($p = 3$).

Figure V.12: First-stage classification accuracy for FDA/KDA subspaces ($p = 3$)

when $p = 1$, the accuracy keeps stable for small misalignment ($<$ 4 pixels), and drops fast with increasing misalignment. Since the second-stage is not affected by the misalignment, a stable output for the first-stage with increasing misalignment will increase the robustness to misalignment in the overall system. This shows the advantage of the 2-stage framework.

A kernel determines the induced bias of a learning algorithm on a specific data set; thus a proper way to select optimal kernel is crucial for such learning algorithms as KDA. However, kernel selecting is not a trivial work [136, 137, 138]. It needs to fit the prior knowledge without excessive learning, which causes the overfitting problem. Many researchers have spent a lot of efforts in determining

Table V.1: First-stage multi-resolution subspace analysis results evaluated under different p.

	p=1	$p = 3$	p=5
PCA	36.4	**86.6**	96.9
FDA	40.1	**88.0**	97.3
KPCA	42.0	**90.2**	99.2
KDA	50.3	**94.0**	97.9

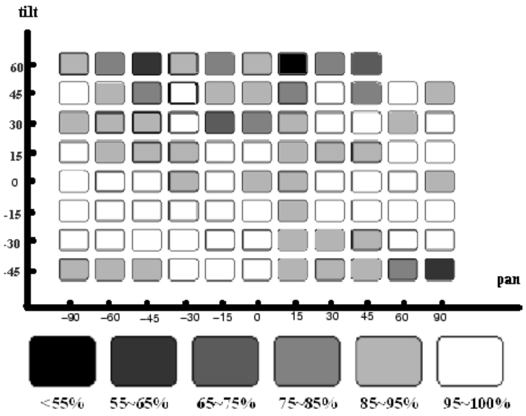


Figure V.13: Grayscale coded diagram for the error distribution of the KDA subspace for the first-stage, evaluated on $p = 3$.

the optimal kernel, including the function form as well as the kernel parameters. Here in this work we will not discuss this problem in detail. Only two types of most commonly used kernels are compared: linear kernel and polynomial kernel, with both parameters fixed. For linear kernel, we use:

$$\hat{\mathbf{x}}_i = \frac{\hat{\mathbf{x}}_i}{\|\hat{\mathbf{x}}_i\|}, i = 1, 2;$$

$$\mathcal{K}(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2) = 1 + \hat{\mathbf{x}}_1 \hat{\mathbf{x}}_2^{\mathrm{T}}. \tag{V.34}$$

The overall accuracy is 92.3%, which is only slightly worse then KDA with Gaussian kernel. Although as reported in [139] a second-order polynomial kernel can achieve

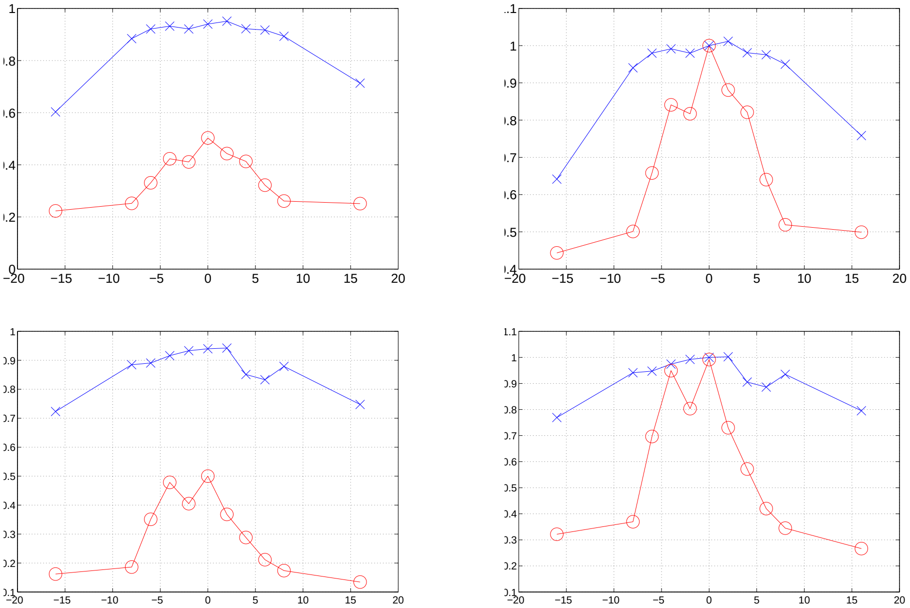
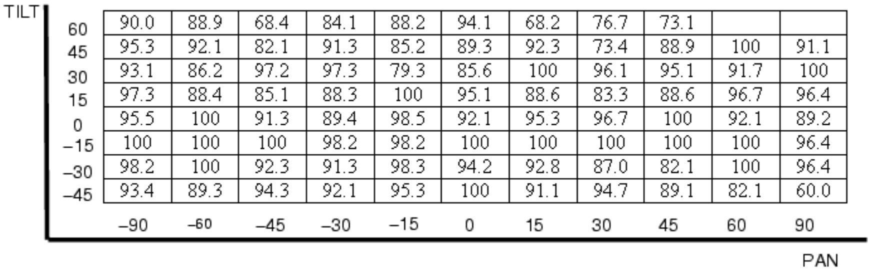


Figure V.14: The performance *w.r.t.* the misalignment. Top row: misalignment in the horizontal direction. Bottom row: misalignment in the vertical direction. Blue curve with x: evaluated on $p = 3$. Red curve with o: evaluated on $p = 1$.

similar performance as the Gaussian kernel, parameter selection for second-order polynomial is not trivial either. The performance for individual poses is shown in Figure V.15.

V.D.2 Stage 2: Refinement

We only use the best results, which is from KDA subspace analysis, as the first-stage output. The pose estimation accuracy after the refinement is summarized in Table Table V.2. For comparison, we compute the refinement from a second stage multi-resolution FDA analysis, which use the poses from the smaller subset to compute the corresponding discriminant subspace and the similar strategy as specified in the first stage to refine the estimate. The results are shown in Tables Table V.3-Table V.6. The comparison shows that by introducing the second-stage structure landmark matching, the estimation accuracy has a marked improvement. This also indicates that the holistic statistical analysis may not be



TILT											
60	90.0	88.9	68.4	84.1	88.2	94.1	68.2	76.7	73.1		
45	95.3	92.1	82.1	91.3	85.2	89.3	92.3	73.4	88.9	100	91.1
30	93.1	86.2	97.2	97.3	79.3	85.6	100	96.1	95.1	91.7	100
15	97.3	88.4	85.1	88.3	100	95.1	88.6	83.3	88.6	96.7	96.4
0	95.5	100	91.3	89.4	98.5	92.1	95.3	96.7	100	92.1	89.2
−15	100	100	100	98.2	98.2	100	100	100	100	100	96.4
−30	98.2	100	92.3	91.3	98.3	94.2	92.8	87.0	82.1	100	96.4
−45	93.4	89.3	94.3	92.1	95.3	100	91.1	94.7	89.1	82.1	60.0
	−90	−60	−45	−30	−15	0	15	30	45	60	90

PAN

Figure V.15: First-stage classification accuracy for FDA/KDA subspaces ($p = 3$) using linear kernel.

sufficient. The accuracy was evaluated by the ratio of samples that were correctly classified. Pose with tilt angle 60^o get poor performance. It is because of the severe occlusion of the salient facial features. Discarding these poses, the overall accuracy can be improved to 84.3%. This is also a limitation of the geometric structure based analysis. The overall performance are summarized in Table Table V.7. The comparison shows that by introducing the second-stage structure landmark matching (FDA subspace), the estimation accuracy has a markable improvement.

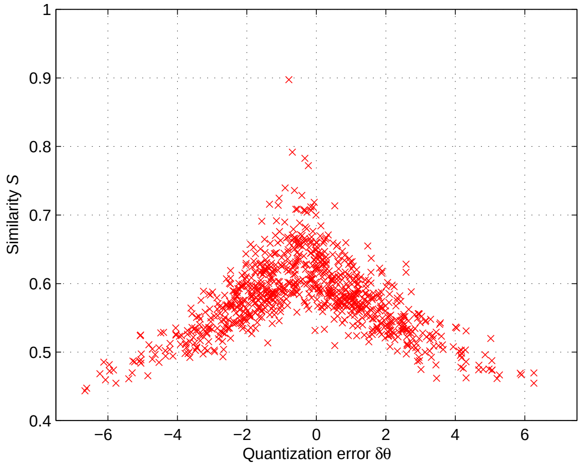


Figure V.16: Relation between the similarity measure and the quantization error.

Table V.2: The overall accuracy (%) using KDA subspace majority voting for the first stage estimation and the semi-rigid bunch graph matching as the second stage refinement. The accuracy is 75.4%.

	-90^o	-60^o	-45^o	-30^o	-15^o	0^o	15^o	30^o	45^o	60^o	90^o
60^o	16.7	30.8	0	50.0	50.0	20.0	30.8	23.1	25.0		
45^o	90.5	87.5	65.6	85.4	89.5	88.2	76.7	53.6	58.3	77.8	67.1
30^o	60.8	44.7	67.5	73.0	76.3	85.3	71.1	62.2	67.6	83.3	86.6
15^o	75.4	71.4	60.5	84.4	94.3	84.0	79.4	82.4	85.7	71.4	87.3
0^o	87.5	73.4	77.0	78.8	77.6	86.0	87.9	70.0	75.0	79.6	79.3
-15^o	67.0	81.2	85.0	78.9	82.5	80.4	89.1	75.9	74.5	84.1	87.0
-30^o	80.9	84.1	92.2	67.7	69.5	64.7	94.2	75.9	87.0	72.1	76.8
-45^o	82.9	65.7	77.1	81.6	76.7	65.6	68.8	81.6	71.1	75.0	30.0

Table V.3: The overall accuracy (%) using PCA subspace majority voting for the first stage estimation and FDA subspace majority voting as the second stage refinement. The accuracy is 43.1%.

	-90^o	-60^o	-45^o	-30^o	-15^o	0^o	15^o	30^o	45^o	60^o	90^o
60^o	30.8	25.0	33.3	7.7	53.8	70.0	58.3	50.0	0		
45^o	76.2	20.8	18.8	19.5	18.4	41.2	37.2	21.4	41.7	36.1	36.8
30^o	54.9	31.6	17.5	35.1	34.2	20.6	28.9	37.8	23.5	83.3	68.3
15^o	36.8	25.7	31.6	12.5	34.3	36.0	41.2	11.8	31.4	71.4	63.6
0^o	71.4	62.5	31.1	40.9	49.3	56.1	51.5	48.3	53.3	60.0	60.7
-15^o	37.2	45.8	43.3	43.9	42.1	67.4	50.9	51.9	36.2	54.5	53.2
-30^o	34.5	54.5	56.9	49.1	45.8	51.0	9.6	18.5	33.3	27.9	41.1
-45^o	3.9	5.7	54.3	68.4	79.1	84.4	45.8	47.4	15.8	10.7	20.0

V.D.3 More Discussion

The above experiments give us a quantized pose classifier. However, in many occasions, we would like to model the pose as a continuous variable. Now the question is whether the above framework is still applicable. The easiest way to verify this is to check whether the similarity measure given by Eq.V.32 and the quantization error from the pose classification has a linear relation. The quantization error from classification is the difference between the actual pose angle from the magnetic sensor and its class label. For example, for a sample with pan

Table V.4: The overall accuracy (%) using FDA subspace majority voting for the first stage estimation and FDA subspace majority voting as the second stage refinement. The accuracy is 44.0%.

	-90^o	-60^o	-45^o	-30^o	-15^o	0^o	15^o	30^o	45^o	60^o	90^o
60^o	30.8	16.7	0	58.3	33.3	60.0	23.1	15.4	25.0		
45^o	90.5	33.3	15.6	24.4	15.8	41.2	30.2	17.9	44.4	38.9	35.5
30^o	52.9	36.8	17.5	32.4	31.6	26.5	31.1	43.2	3.8	69.4	61.0
15^o	36.8	34.3	34.2	25.0	34.3	48.0	52.9	14.7	28.6	60.7	65.5
0^o	69.6	71.9	29.5	33.3	49.3	63.2	54.5	45.0	53.3	63.3	60.0
-15^o	36.2	43.8	41.7	38.6	40.4	69.6	56.4	53.7	36.2	59.1	50.6
-30^o	34.5	54.5	56.9	49.1	54.2	52.9	21.2	25.9	29.6	41.1	45.3
-45^o	5.3	5.7	51.4	68.4	79.1	84.4	62.5	52.6	23.7	25.0	50.0

Table V.5: The overall accuracy (%) using Kernel PCA subspace majority voting for the first stage estimation and FDA subspace majority voting as the second stage refinement. The accuracy is 47.3%.

	-90^o	-60^o	-45^o	-30^o	-15^o	0^o	15^o	30^o	45^o	60^o	90^o
60^o	16.7	23.1	28.6	66.7	33.3	30.0	61.5	30.8	58.3		
45^o	71.4	29.2	25.0	26.8	23.7	52.9	39.5	17.9	33.3	38.9	40.8
30^o	51.0	50.0	20.5	48.6	36.8	8.8	24.4	45.9	35.3	91.7	70.7
15^o	40.4	31.4	23.7	25.0	44.0	28.0	50.0	32.4	51.4	82.1	67.3
0^o	74.1	67.2	44.3	33.3	59.7	71.9	53.0	56.7	58.3	51.7	65.2
-15^o	43.6	43.8	38.3	49.1	43.9	58.7	56.4	53.7	36.2	61.4	63.6
-30^o	39.1	61.4	64.7	52.7	52.5	56.9	11.5	42.6	38.9	44.2	48.2
-45^o	11.8	25.7	48.6	63.2	72.1	90.6	56.3	50.0	25.0	14.3	30.0

angle 33.2^o, since its closest class label would be 30^o, the quantization error would be 3.2. The above procedure would classify the sample as belonging to the class with pose angle 30^o. If there is such a linear relation, it means that the similarity measure can be used to infer the actual pose angle. To get a clearer comparison, we use the samples whose pan angles are exact the same as the class label, or $-60^o, -45^o, -30^o, \cdots, 45^o, 60^o$ as the training sample set. For the testing samples, we denote the similarity measure obtained from the second stage as S; and the quantization error as $\delta\theta$. In Figure V.16, the experimental results answers the

Table V.6: The overall accuracy (%) using KDA subspace majority voting for the first stage estimation and FDA subspace majority voting as the second stage refinement. The accuracy is 53.4%.

	-90^o	-60^o	-45^o	-30^o	-15^o	0^o	15^o	30^o	45^o	60^o	90^o
60^o	25.0	30.8	0	58.3	41.7	50.0	46.2	76.9	58.3		
45^o	81.0	25.0	25.0	36.6	34.2	61.7	30.2	39.3	58.3	41.7	48.7
30^o	60.8	44.7	27.5	32.4	44.7	29.4	40.0	56.8	38.2	91.7	69.5
15^o	49.1	60.0	34.2	21.9	40.0	36.0	47.1	55.9	65.7	85.7	69.1
0^o	83.0	75.0	47.5	31.8	64.2	78.9	65.2	68.3	65.0	55.0	68.1
-15^o	57.4	64.6	46.7	40.4	54.4	63.0	69.1	57.4	44.7	65.9	68.8
-30^o	51.8	52.3	74.5	56.4	62.7	52.9	25.0	38.9	27.8	39.5	51.8
-45^o	25.0	60.0	54.3	57.9	79.1	65.6	60.4	36.8	46.4	25.0	40.0

Table V.7: Comparison of results from different second-stage refinement.

KDA	PCA	FDA	KPCA	KDA
+BG	+FDA	+FDA	+FDA	+FDA
75.4	43.1	44.0	47.3	53.4

question about how much information we can reveal from the similarity measure S. Only the correctly classified samples are shown here for samples with tilt label 0 and pan label from -60^o to 60^o.

It shows that when the quantization error becomes larger, the similarity measure does getting smaller, which mean it becomes less similar to the model bunch graph. In the other word, when the similarity measure gets small, we can make the conclusion that the quantization error is bigger. This indicates that the similarity measure is appropriate to be used to infer the continuous pose angle.

The text of this, in part, is a reprint of the material as it appears in: J. Wu and M. M. Trivedi, "An Integrated Two-stage Framework for Robust Head Pose Estimation", in *the Lecture Notes of Computer Science, IEEE International Workshop on Analysis and Modeling of Faces and Gestures (AMFG), China, in Conjunction with ICCV 2005*, 2005, Beijing and Junwen Wu and Mohan M. Trivedi, "A Two-stage Head Pose Estimation: Framework and Evaluations." to Appear, *Pattern Recognition.* I was the primary researcher of the cited material and the co-author listed in these publication directed and supervised of the research which forms a basis for this chapter.

Chapter VI

Resolution Enhancement

VI.A Inter-Pixel Interference Elimination

In this work the similar idea of high-frequency loss compensation for multi-frame based super-resolution reconstruction is exploited. Resolution enhancement is achieved by iteratively estimating and eliminating inter-pixel interference from neighboring pixels. At a given location, the inter-pixel interference is from an integral effect of several low-pass filtering processes, each determined by one of its neighboring pixels. We propose using a Gaussian mixture to model the probability of the integral inter-pixel interference. The Gaussian mixture is determined by both the local image constraints and the *local variation indicator*(LVI). Local image constraints come from the image derivative priors, which evaluate the similarity between the current pixel and its neighborhood. Larger image derivative prior implies more interference. We use differences between the current pixel and its neighboring pixels as the image derivative priors. The LVI shows the reliability of the neighboring pixels. It is obtained from both the temporal variation and the spatial variation. Neighboring pixels with greater LVI are considered less reliable, hence provide less information for inferring inter-pixel interference. Also we show from the frequency domain representation that inter-pixel interference can actually be related to high-frequency loss. By estimating and compensating the missing high frequency details iteratively, images with higher resolution are recovered. Experimental evaluation on varying inputs, including faces, synthetic text subjects, as well as license plates, validates the algorithm. The diagram is shown in Figure VI.1.

The component Gaussians are confined by the image derivative priors, which are approximated by the differences between the observed neighboring pixels, as well as the Euclidean distance between them. Using the pixel observations to estimate the image derivative priors introduces observation noise. To address this, we introduce a concept of *local variation indicator* (LVI) to describe the confidence level on each neighboring pixel. LVI incorporates variant image degradation

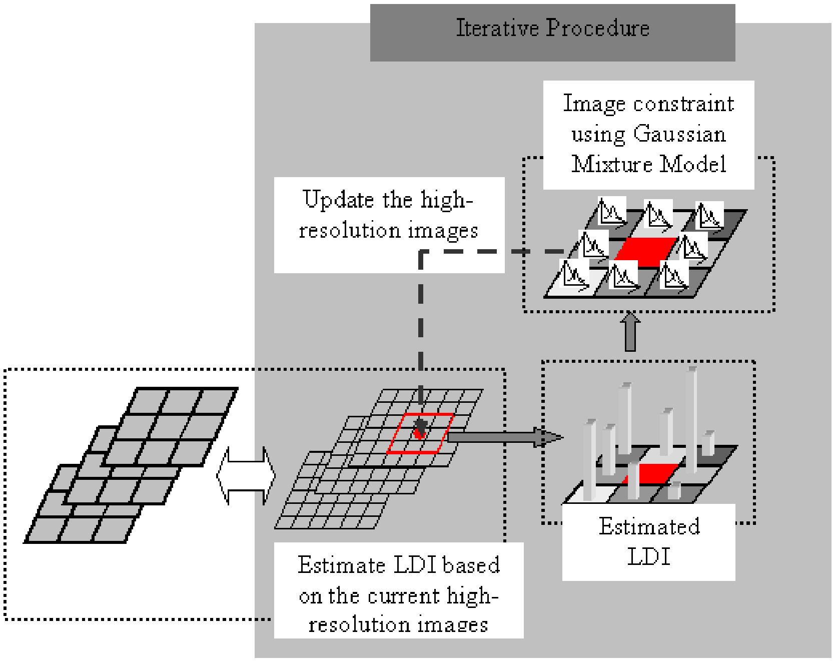

Iterative Procedure
Image constraint using Gaussian Mixture Model
Update the high-resolution images
Estimated LDI
Estimate LDI based on the current high-resolution images

Figure VI.1: The diagram of the proposed super-resolution reconstruction by interpixel interference elimination algorithm.

factors, such as PSF, motion blur as well as the quantization errors as one integral distortion. Neighboring pixels with higher LVI are regarded as less reliable for predicting the interference, since less authentic scene information they may contain. The procedure can be separated into two steps. In the first step LVI is estimated locally from successive frames. In the second step a Gaussian mixture probability model from the image derivative constraints is used to estimate the inter-pixel interference. The Gaussian mixture model uses LVI as the weight constraints. The predicted interference, which is actually the high-frequency loss, is subtracted from the old estimate to refine it. The refined high-resolution estimates are used to re-estimate the LVI. The procedure is repeated until predicted high-frequency loss is negligible.

VI.A.1 Algorithm Framework

Ideally, the images should be acquired through impulse sampling in both spatial and temporal directions. The obtained t_0-th image at location (x_i, y_i) can be represented as:

$$I(x_i, y_i; t_0) = S(x, y; t) * *\delta(x - x_i, y - y_i) * \delta(t - t_0); \tag{VI.1}$$

where S is the 2-d continuous function describing the scene and δ is the impulse function. In spatial domain and temporal domain, δ is respectively defined as follow:

$$\delta(x - x_i, y - y_i) = \begin{cases} 1 & \text{if } x = x_i \quad \text{and} \quad y = y_i, \\ 0 & \text{otherwise.} \end{cases};$$

and

$$\delta(t - t_0) = \begin{cases} 1 & \text{if } t = t_0, \\ 0 & \text{otherwise.} \end{cases}.$$

$**$ represents for the 2D convolution and $*$ is the 1D convolution. However, neither spatial nor temporal sampling can be ideal. The non-zero aperture time of the image acquisition device causes smoothing effect over the temporal direction. With the dynamics of the scene, the non-zero aperture time will contribute to spatial blurring. Also, since the size of the optical sensors can not be infinitely small, the spatial sampling is usually a low-pass procedure instead of the ideal impulse sampling. In frequency domain, the low-pass filters cause lower response over high-frequency component, correspondingly in the spatial domain, inter-pixel interferences occur. Let the low-pass filter at time t_0 be $h_l(x, y; t_0)$, now the image formation procedure can be represented by:

$$I_o(u, v; t_0) = S(x, y; t_0) * *h_l(x - u, y - v; t_0); \tag{VI.2}$$

where (u, v) defines pixels on the low-resolution image coordinate system. Super-resolution problem is to recover $S(x_i, y_i; t_0)$ from $I_o(u, v; t = t_0 - p, \cdots, t_0 + p)$. We use (x_i, y_i) to represent the discrete sampling of (x, y) on high-resolution image

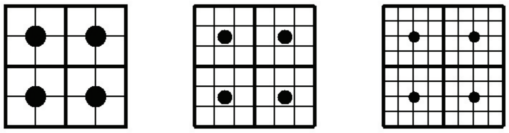


Figure VI.2: Illustration of the original LR image coordinate system and the HR image coordinate system. The dark dots: the low-resolution image grid; the boxes: the high-resolution image grid. The bold box: the mapping neighborhood.

coordinate system. We assume each low-resolution image pixel corresponds to a disjoint high-resolution image neighborhood, as shown in Figure VI.2. The low-resolution images are interpolated first so that two sets of image coordinates are aligned. Also, for simplicity, we use "*low-resolution images*" to refer to the post-interpolated low-resolution images unless otherwise specified. Therefore, there is only one image coordinate system involved. Furthermore, low-resolution images can be written as $I_o(x_i, y_i; t_0 - p, \cdots, t_0 + p)$. The interpolation procedure also introduces smoothing, however, we can assume the spatial smoothing h_l in Eq.VI.2 has already included this. Now Eq. VI.2 can be written as:

$$I_o(x_i, y_i; t_0) = S(x, y; t_0) * *h_l(x - x_i, y - y_i; t_0). \tag{VI.3}$$

Gaussian Mixture model to Predict the Interference

Another way to view Eq. VI.2 is to consider $I_o(x_i, y_i; t_0)$ as a result of an additive procedure, which can be represented as the superposition of the interference from neighboring pixels on top of its true scene value. If denoted the neighborhood as $\mathcal{D}_i = \{i_k : d(\mathbf{x}_i, \mathbf{x}_{i_k}) <= \zeta, \quad \mathbf{x}_i = (x_i, y_i), \quad \mathbf{x}_{i_k} = (x_{i_k}, y_{i_k})\}$, we have:

$$I_o(x_i, y_i; t_0) = S(x_i, y_i; t_0) + \sum_k f\{i, i_k; t_0\}; \quad i_k \in \mathcal{D}_i. \tag{VI.4}$$

$f\{i, i_k; t_0\}$ is interference on pixel (x_i, y_i) from its neighboring pixel (x_{i_k}, y_{i_k}); which is also constrained by other (x_{i_k}, y_{i_k}). For simplification, denote the integral interference as:

$$F(i;t_0) = \sum_k f\{i, i_k; t_0\}. \tag{VI.5}$$

Due to noise from the image acquisition device, the interference is not deterministic. We use a probability model to characterize the statistical property of the integral interference. An iterative scheme is used to reduce the estimation error step by step. At each step, when estimating the interference at location (x_i, y_i), we assume its neighboring pixels are accurate. Inter-pixel's interference $F(i;t_0)$ is determined by the image derivative prior, which is the difference between observations of (x_i, y_i) and $(x_{i_k}, y_{i_k}) \in \mathcal{D}_i$. Probability of the integral interference $F(i;t_0)$ is modeled by the following Gaussian mixture:

$$F(i;t_0) \sim \sum_k \omega_k \mathcal{N}(f; \partial I(i, i_k; t_0), \sigma d_{i,i_k}); \tag{VI.6}$$

where ω_k is the weight for each Gaussian component, which describes the confidence level of each neighboring pixel.

$$\partial I(i, i_k; t_0) = I_o(x_{i_k}, y_{i_k}; t_0) - I_o(x_i, y_i; t_0)$$

defines the image derivative prior between pixel (x_i, y_i) and its neighboring pixel (x_{i_k}, y_{i_k}); and d_{i,i_k} is the Euclidean distance between pixel (x_i, y_i) and (x_{i_k}, y_{i_k}). The size of neighborhood $\mathcal{D}_i$ is determined by the support of the low-pass filters, which also confines parameter σ in this model. $\mathcal{N}(\bullet; \mu, \sigma)$ is a Gaussian characterized by μ and σ; so we have

$$\mathcal{N}(F; \partial I(i, i_k; t_0), \sigma d_{i,i_k}) = \frac{1}{\sqrt{2\pi}\sigma d_{i,i_k}} \exp\{-\frac{[F - \partial I(x_{i_k}, y_{i_k}; t_0)]^2}{2\sigma^2 d_{i,i_k}^2}\}. \tag{VI.7}$$

After obtaining the interference estimate, scene estimate at (x_i, y_i) can be updated accordingly. Let the estimated interference be $\hat{F}(i;t_0)$, from Eq.VI.4 and Eq.VI.5, the scene estimate can be obtained by:

$$\hat{S}(x_i, y_i; t_0) = I_o(x_i, y_i; t_0) - \hat{F}(i; t_0); \tag{VI.8}$$

where $\hat{S}(x_i, y_i; t_0)$ is the scene estimate. The procedure is repeated again to refine the estimate after all frames are processed.

Find the Weights from Prior Knowledge

The image derivative priors contains observation error of the neighboring pixels, which indicates that the Gaussian components are not equally reliable. By determining the temporal variation and spatial variation, we can get the prior knowledge to determine the reliability for each neighboring pixel, i.e., the importance of each Gaussian component. We introduce the concept of local variation indicator (LVI) to incorporate the temporal variation and spatial variation as the prior knowledge. LVI is used as the confidence level for the component Gaussians. For pixels with larger LVI, we consider them as less reliable.

I. Temporal Variation

The temporal variation is used to model the temporal noise of the scene. We assume the noise is i.i.d. Gaussian, and the scene has slow dynamics so that it keeps consistent over the $2p+1$ frames. Based on these assumptions, we have:

$$I_o(x_i, y_i; t_0 + \tau) = S(x_i, y_i; t_0) + n(x_i, y_i; t_0 + \tau); \tau = -p, \cdots, p;$$

where n is the noise. With $2p+1$ observations $I_o(x_i, y_i; t = t_0 - p \cdots, t_0 + p)$, the ML estimate for the scene $S(x_i, y_i; t_0)$ is the mean:

$$\hat{S}_{ML}(x_i, y_i; t_0) = \frac{1}{2p+1} \sum_{\tau=-p}^{p} I_o(x_i, y_i; t_0 + \tau).$$

The temporal variation is defined as the bias from the current observation to the ML estimate:

$$b_i(t_0) = I_o(x_i, y_i; t_0) - \hat{S}_{ML}(x_i, y_i; t_0).$$

II. Spatial Variation

We assume the point spread function (PSF) is unknown and non-uniform over the image domain, which is more general for real data. The space-varying PSF, together with more complicated blurring procedure from atmosphere disturbance, camera motion, interpolation, noise etc, cause a non-uniform low-resolution degradation over the whole image. We use a local model for describing the low-pass filtering process. If we assume the local low-pass filtering process is sufficiently slow so that it keeps constant for successive $2p+1$ frames, and each pixel from the low-resolution image corresponds to a neighborhood $\mathcal{Q}$ on the high-resolution image, as shown in Figure VI.2. Here the low-resolution images refer to the original sampled images, or pre-interpolated low-resolution images. This means that the low-resolution pixels are in the low-resolution image coordinate system. As stated before, the pre-interpolated low-resolution images are represented as $I_o(u, v; t_0+\tau)$. Then the filtering procedure gives:

$$\sum_i S(x_i, y_i; t_0 + \tau)\nu(x_i, y_i; t_0) = I_o(u, v; t_0 + \tau); \quad i \in \mathcal{Q}; \quad \tau = -p, \cdots, p; \tag{VI.9}$$

$\nu(x_i, y_i; t_0)$ are filter coefficients, which evaluate the spatial variations over neighborhood $\mathcal{Q}$ during low-resolution image generation. With $\tau = -p, \cdots, p$ frames, we can have $2p+1$ constraints. Rewrite it into matrix format, we have:

$$\mathbf{S}_i^{\mathrm{T}}(t_0 + \tau)\boldsymbol{\nu}_i(t_0) = \mathbf{I_o}(u, v; t_0); \tag{VI.10}$$

where $\mathbf{S}$ is the matrix whose column vectors are lexicographically ordered pixels from $\mathcal{Q}$, with each column corresponding to one frame; and $\mathbf{I_o}$ is the vector formed by low-resolution pixel $(u, v; t_0 + \tau), \quad \tau = -p, \cdots, p$.

The set of filter parameters evaluates the variations of the high-resolution estimate across space and time. Eq. VI.10 is highly ill-conditioned due to the fact that for slowing changing scene, $\mathbf{S}$ can hardly be a full rank matrix. A regularization term needs to be defined to find a reasonable solution. We define the optimal filtering parameters as those that satisfy the degradation model for the successive $2p+1$ frames from MSE sense. This leads to an objective function as follows:

$$\mathcal{J}(\boldsymbol{\nu}(t_0)) = \sum_{\tau=-p}^{p} \sum_i [K \times I_o(u, v; t_0 - \tau) - \nu(x_i, y_i; t_0)S(x_i, y_i; t_0 - \tau)]^2 + \lambda\nabla\boldsymbol{\nu}(t_0), \tag{VI.11}$$

$$s.t. : \|\boldsymbol{\nu}(t_0)\|_1 = 1. \tag{VI.12}$$

K is the number of pixels inside neighborhood $\mathcal{Q}$. The first term of Eq. VI.11 is from Eq.VI.9, while the second term is a smoothing term, defined as:

$$\nabla\boldsymbol{\nu}(t_0) = \|\partial_x\boldsymbol{\nu}(t_0)\|_2 + \|\partial_y\boldsymbol{\nu}(t_0)\|_2 + \|\partial_{xy}\boldsymbol{\nu}(t_0)\|_2 + \|\partial_{yx}\boldsymbol{\nu}(t_0)\|_2. \tag{VI.13}$$

If pixels on the high-resolution image coordinate system make the same contribution to the corresponding low-resolution pixel, filter parameters $\nu(x_i, y_i; t_0)$ should be uniform over $\mathcal{Q}$. On the other hand, if one filter parameter $\nu(x_i, y_i; t_0)$is far from the average, the corresponding high-resolution estimate has a large difference from the input low-resolution pixel (which is the direct observation of the scene), hence the corresponding high-resolution estimate tends to be less reliable. Based on this observation, the target is to find a set of parameters that describe the high-resolution estimates' bias from the average. A simplified solution is provided for Eq.VI.11: a uniform function is chosen as the initial values for $\boldsymbol{\nu}(t_0)$, which is $\boldsymbol{\nu}(t_0) = \nu_0$, and then the parameters are obtained by one-step steepest descent update as follows:

$$\hat{\nu}(x_i, y_i; t_0) = \nu_0 + \mu \sum_{\tau=-p}^{p} [I_o(u, v; t_0 - \tau) - \nu_0 S(x_i, y_i; t_0 - \tau)] S(x_i, y_i; t_0 - \tau). \tag{VI.14}$$

The solution is normalized to satisfy the constraint in Eq. VI.12, which gives:

$$\begin{aligned} \nu(x_i, y_i; t_0) &= \frac{\hat{\nu}(x_i, y_i; t_0)}{\|\hat{\boldsymbol{\nu}}(t_0)\|_1}; \\ \hat{\boldsymbol{\nu}}(t_0) &= [\hat{\nu}(x_i, y_i; t_0)]_i; \\ \nu(x_i, y_i; t_0) &\leftarrow \nu(x_i, y_i; t_0) - \nu_0. \end{aligned} \tag{VI.15}$$

The updated $\nu(x_i, y_i; t_0)$ evaluates the relative spatial variation. At each iteration, the relative spatial variation of the high-resolution estimate is re-computed.

Overall Algorithm

Local variation indicator is modeled as a function $g(\cdot)$ of both the temporal variation and the spatial variation:

$$w(i; t_0) = g\{-[b_i(x_i, y_i; t_0)\nu(x_i, y_i; t_0)]^2\}.$$

In this work, exponential function is used:

$$w(i;t_0) = \exp\{-[b_i(x_i,y_i;t_0)\nu(x_i,y_i;t_0)]^2\}. \tag{VI.16}$$

As discussed earlier, LVI evaluates how reliable the neighboring pixel is for estimating the inter-pixel interference. Hence, we use it directly as the weights of the Gaussian mixture in Eq. VI.6.

We use Maximum a Posteriori (MAP) criteria to estimate $F(i;t_0)$, which gives:

$$\hat{F}(i;t_0) = \arg\max_F(\mathcal{G}(F)); \tag{VI.17}$$

where:

$$\mathcal{G}(F) = \sum_k w(i_k;t_0)\mathcal{N}(F;\partial I(i,i_k;t_0),\sigma d(i,i_k)).$$

Eq. VI.7, Eq. VI.8, together with Eq. VI.16 and Eq. VI.17 determine the recursive procedure for refining the high-resolution estimate. MAP solution is found numerically by steepest descent method, as follows:

$$F^{j+1}(i;t_0) = F^j(i;t_0) - \alpha\nabla\mathcal{G};$$

$$\nabla\mathcal{G} = \frac{\partial}{\partial F}\mathcal{G}(F)|_{F^j} = \sum_k \frac{w_{i_k}}{\sqrt{2\pi}\sigma d_{i_k}} \frac{\partial I(i,i_k) - F^j}{\sigma^2 d_{i_k}^2} \exp\{-\frac{[F^j - \partial I(i,i_k)]^2}{2\sigma^2 d_{i_k}^2}\}. \tag{VI.18}$$

We omit the time index t_0 in Eq. VI.18 for simplification.

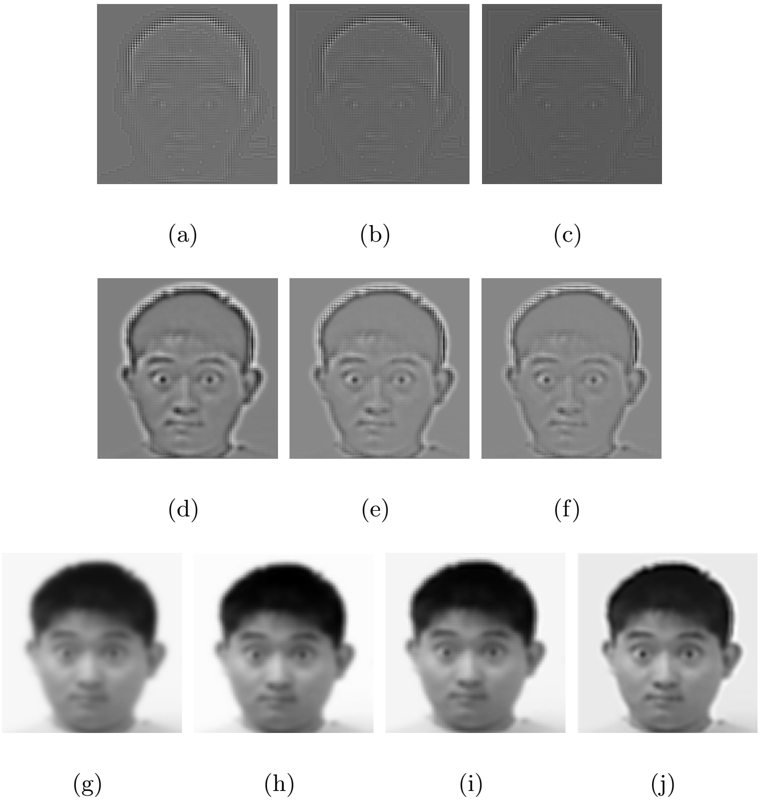
With the prediction of the interference, we can update the scene estimate $\hat{S}(x_i,y_i;t)$ with Eq. VI.8 and VI.18:

$$\hat{S}(x_i,y_i;t_0) = I_o(x_i,y_i;t_0) - \hat{F}(i;t_0). \tag{VI.19}$$

All images are updated using this procedure, and the updated estimates are used as the new given low-resolution input, which gives:

$$I^{(new)}(x_i,y_i;t_0+\tau) \leftarrow \hat{S}(x_i,y_i;t_0+\tau); \tau = -p,\cdots,p.$$

Then the whole process is repeated. The iterative procedure stops when dynamic range of the predicted interference is small enough. Figure VI.3 gives an example of the procedure. Initial estimate of the high-resolution images are bilinear interpolations of the $(2p+1)$ successive frames. After three iterations, the dynamic range of the error is small enough and the iteration stops.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i) (j)

Figure VI.3: The iterative procedure for HR reconstruction. Figure VI.3(g): initial HR frame (bilinear interpolated). Left to right: more iterations. Top row: LVIs. middle row: estimated interferences. bottom row: the reconstructed HR image.

More Discussions in the Frequency Domain

In this section we will look into the algorithm from the frequency domain. The observation is a smoothed version of the original scene. At pixel (x_i, y_i), the observation is given by:

$$I(x_i, y_i; t_0) = (h_l * *S)(x_i, y_i; t_0). \tag{VI.20}$$

$L(x_i, y_i; t_0)$ is a low-pass filter. Suppose the frequency indexes are ω_1 and ω_2, in frequency domain this can be rewritten as:

$$\mathcal{I}(\omega_1, \omega_2; t_0) = \mathcal{F}(I(x_i, y_i; t_0));$$

$$\mathcal{S}(\omega_1, \omega_2; t_0) = \mathcal{F}(S(x_i, y_i; t_0));$$

$$\mathcal{L}(\omega_1, \omega_2; t_0) = \mathcal{F}(h_l(x_i, y_i; t_0));$$

$$\mathcal{I}(\omega_1, \omega_2; t_0) = \mathcal{L}(\omega_1, \omega_2; t_0)\mathcal{S}(\omega_1, \omega_2; t_0); \tag{VI.21}$$

where $\mathcal{F}(\cdot)$ is the Fourier transform. The corresponding high-pass filter is:

$$\mathcal{H}(\omega_1, \omega_2; t_0) = \mathbf{1}(\omega_1, \omega_2; t_0) - \mathcal{L}(\omega_1, \omega_2; t_0),$$

where $\mathbf{1}(\omega_1, \omega_2; t_0)$ is an all-pass filter. Therefore the error is:

$$\varepsilon(\omega_1, \omega_2; t_0) = \mathcal{H}(\omega_1, \omega_2; t_0)\mathcal{S}(\omega_1, \omega_2; t_0); \tag{VI.22}$$

which is actually the suppressed high-frequency component. More details can be recovered if high-frequency loss can be predicted. Therefore, a higher resolution image can be reconstructed by compensating high-frequency loss.

Given unknown true scene $\mathcal{S}(\omega_1, \omega_2; t_0)$, an intuitive way to predict the error is to use the current high-resolution estimates, which is $\mathcal{I}_o(\omega_1, \omega_2; t_0)$. It gives:

$$\varepsilon(\omega_1, \omega_2; t_0) = \mathcal{H}(\omega_1, \omega_2; t_0)\mathcal{I}_o(\omega_1, \omega_2; t_0). \tag{VI.23}$$

Recall that in Eq. VI.6, we use the derivative prior to form a probability model for the inter-pixel interference. The derivative prior is actually the spatial representation of $\mathcal{H}(\omega_1, \omega_2; t_0)\mathcal{I}_o(\omega_1, \omega_2; t_0)$. Comparing Eq. VI.22 with Eq. VI.6, we can see that Eq. VI.6 provides a local probability model for estimating $\epsilon(\omega_1, \omega_2; t_0)$, which is high-frequency loss. Therefore, the inter-pixel interference we estimate above can be related with the missing high-frequency details. Inter-pixel interference elimination can also be related to high-frequency loss compensation. In frequency domain, the procedure can be summarized as follows:

- Predict high-frequency component, or function of the inter-pixel interference, using estimate of the HR images:

$$\hat{\varepsilon}_k(\boldsymbol{\omega}; t) = \mathcal{H}(\omega_1, \omega_2; t_0)\mathcal{I}^{(old)}(\omega_1, \omega_2; t_0). \qquad \text{(VI.24)}$$

- Refine the HR estimates using the predicted high-frequency component obtained from above:

$$\mathcal{I}^{(new)}(\omega_1, \omega_2; t_0) = \mathcal{I}^{(old)}(\omega_1, \omega_2; t_0) + \hat{\varepsilon}(\omega_1, \omega_2; t_0). \qquad \text{(VI.25)}$$

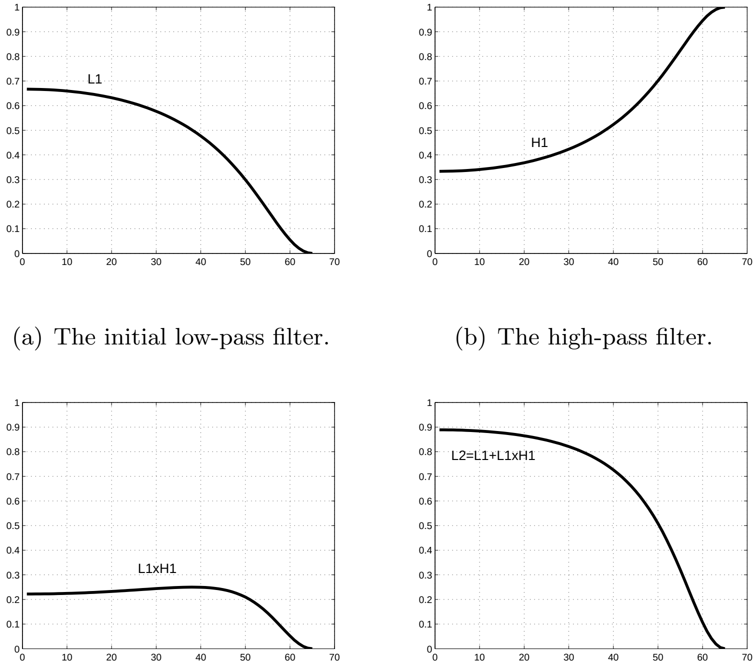
In whole, the procedure is:

$$\mathcal{I}^{(new)}(\omega_1, \omega_2; t_0) = [\mathcal{L}(\omega_1, \omega_2; t_0) + \mathcal{L}(\omega_1, \omega_2; t_0)\mathcal{H}(\omega_1, \omega_2; t_0)]\mathcal{S}(\omega_1, \omega_2; t_0). \quad \text{(VI.26)}$$

The second term has a higher band-width, which means that the refined HR images contain more high-frequency details as desired. Figure VI.4 gives a 1-D illustration. The refined high-resolution estimate is the output from the filter after high-frequency component compensation, which is shown in Figure VI.4(d). Comparing with the initial low-pass filter (Figure VI.4(a)), more high-frequency components are preserved.

VI.A.2 Experimental Evaluation

In this section, the performance of the proposed inter-pixel interference elimination based resolution enhancement algorithm is presented. The focus of the experiments in this work is on the high-resolution reconstruction for human faces. Human faces are different from other subjects, like text and scenery, due to its non-planarity and non-rigidity. Besides the global motion, the dominant motion is the local motion caused by changes in expression, self-shadow, head rotation etc. Many single frame based face super-resolution algorithms have been proposed to facilitate face recognition [70, 71, 72, 73, 74]. We apply the proposed algorithm over substantive face videos collected under different conditions. Performance is



(a) The initial low-pass filter.

(b) The high-pass filter.

(c) The estimation of the high-pass filter.

(d) The refined low-pass filter.

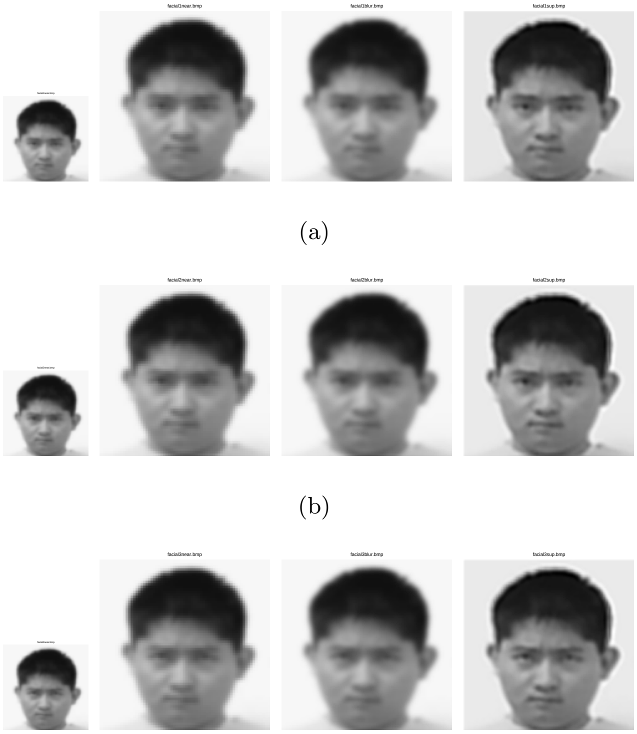
Figure VI.4: 1D illustration of the refinement procedure. Top row: left: the initial low-pass filter; right: the high-pass filter. Bottom row: left: the estimated high-pass filter; right: the refined low-pass filter.

evaluated and compared with other algorithms. Examples of the experimental evaluation are presented and discussed in this section. The experimental results show substantial improvement. In all the experiments, the input frames are registered using a perspective projection assumption.

Face Videos with Changes in Expression

We first evaluate the proposed algorithm over face videos with changing facial expressions. Subtle changes in facial features are not visible in the low-resolution images, while in the reconstructed high-resolution images the details become perceivable. We sub-sampled the original images to get the low-resolution video, which is further blurred by a box blur filter and then used as the input observation.

In Figure VI.5, some results from the super-resolution reconstruction are shown and compared with the standard widely used interpolation techniques. Three successive frames are used for the reconstruction. The first column shows the input low-resolution images. The second column shows zoomed images from the nearest neighbor interpolation. The third column shows the bilinear interpolation results. The fourth column shows the super-resolution reconstruction results. From the experimental results, we can see that the proposed algorithm recovers more facial feature details, which is significantly better than the blurry interpolated images.



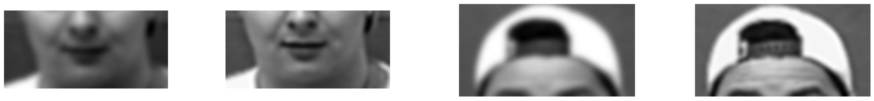
(a)

(b)

(c)

Figure VI.5: Examples of the experimental results. First Column: LR images. Second column: nearest neighbor interpolation. Third columns: bilinear interpolation. Fourth column: resolution enhanced results from the proposed algorithm.

Figure VI.6 provides another example for the super-resolution reconstruction results. The original sequences are from Cohn-Kanade database [1], which contains face videos with substantive facial expression change. Due to the copyright issue, only partial faces are shown here for performance illustration. Similarly, the original sequence are sub-sampled and box blurred, and then used as the input low-resolution observation. The left images in Figure VI.6(a) and Figure VI.6(b) give the bilinear interpolated images for the low-resolution input. The corresponding right images show the results from the proposed super-resolution algorithm. We can clearly see that the blurry images are improved after the resolution enhancement.



(a) Lip area.

(b) Cap area.

Figure VI.6: Reconstruction for the lip area and the cap area (sequences from Cohn-Kanade facial expression database [1]). In both examples, the left images: results from bilinear interpolation; the right images: high-resolution reconstruction results.

Videos with Large Head Motion

In Figure VI.7 we compare our results with the super-resolution optical flow algorithm proposed by Dr. Baker et al. [63]. We use the same video as in [63] for comparison. The results of super-resolution optical flow are courtesy of Dr. Baker. For a fair comparison, in both algorithms, 5 frames were used for the reconstruction in the proposed algorithm. Super-resolution optical flow requires sufficient textures for accurate flow estimation. Also, the algorithm requires motion within a certain range so that the flow estimation can be accurate enough. Therefore, in those areas that the two conditions cannot be satisfied, there will be undesirable artifacts. For face video, one occasion artifacts may appear is in reconstructing the blinking eyes. This can be seen from the third image in the rightmost column of Figure VI.7, where the proposed inter-pixel interference elimination based algorithm can achieve better performance. Figure VI.7 also gives some other examples, where similar perceptual performance can be achieved for both algorithms.

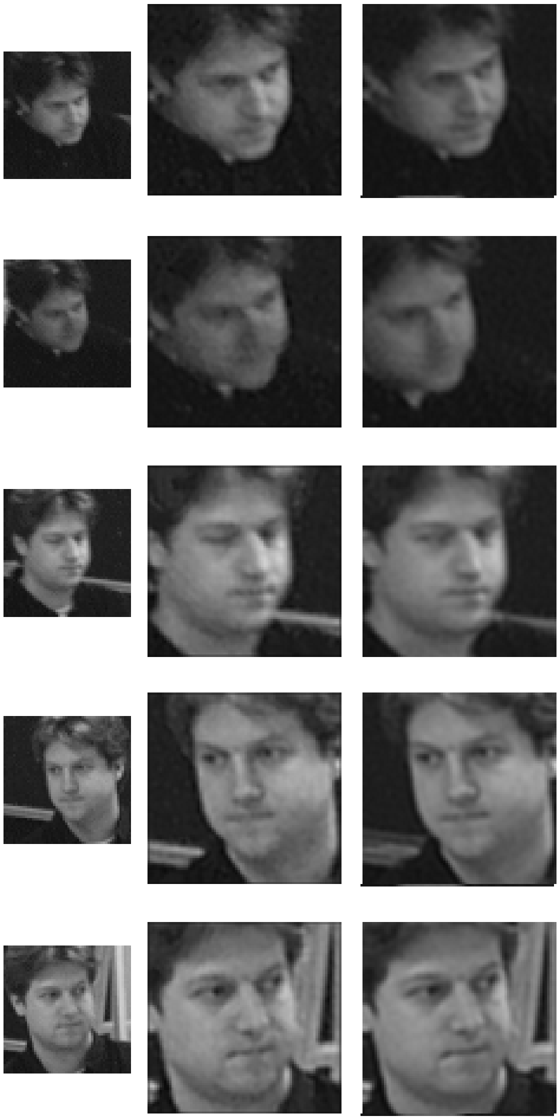


Figure VI.7: Examples from sequence with large head motion. First Column: original LR images. Second column: super-resolved results. Third columns: high resolution frames from [2]. The sequence is the courtesy of Dr. Baker.

Usually there is a higher requirement for the video capturing system when fast motion presents. Motion blur might occur if the shuffler speed is not high enough. In this experiment, we study the performance for face videos with motion blur and want to measure the performance of the high-resolution reconstruction algorithm. For a clear comparison with the ground-truth high-resolution images, we use videos acquired in the following way as the input: we first synthetically introduce motion blur into the original video sequence. And then each frame is sub-sampled by a factor of 2 in both directions. We use linear motion blur filter: first the motion direction is estimated and then pixels along the moving direction in the successive frames are superimposed onto the current video frame at a lowered intensity to get the blurry effect. Examples of the experimental results are shown in Figure VI.8. We use a color video from OSU database. In both examples, the first image is the frame from the original video sequence, and the second image is the motion blurred image. The second image is sub-sampled by a factor of 2 in both directions to give the observation image we use for super-resolution reconstruction, which is shown in the third column. The fourth image gives the reconstructed results. Successive 5 frames are used. Both results show that the resolution-enhanced images give perceptually favorable results.

Videos from Omni-directional Video Camera (ODVC)

Omni-directional video cameras (ODVC) are widely used for their 360-degree field of view [10, 140, 11]. Pseudo-perspective video can be generated by using the ODVC geometry [140, 11] so that existing algorithms for rectilinear cameras can be applied. However, the transformed images are typically low resolution and suffer from non-uniform distortion across the images [141]. VI.A.2 gives an example of the omni-directional image. Although stereo pair can be used to alleviate the problem [142], the expenses are increased and applications are limited by the deployment of the omni-directional cameras. This motivates our research on enhancement for images from omni-directional camera video stream.

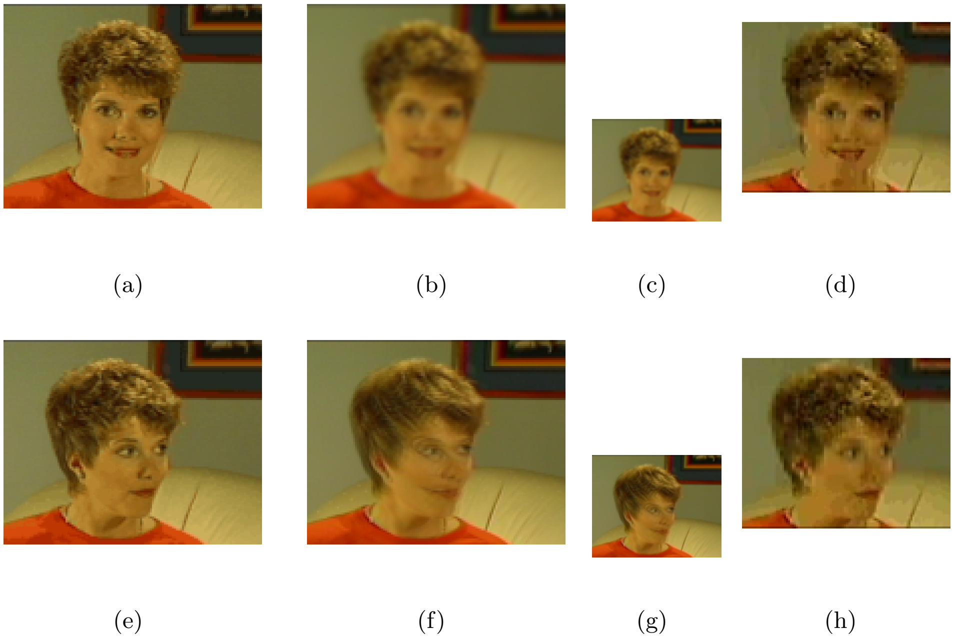


Figure VI.8: HR reconstruction for motion-blurred color face video. The first column: the original frames. The second column: corresponding motion blurred frames. The third column: sub-sampled blurry images (input). The fourth column: results.

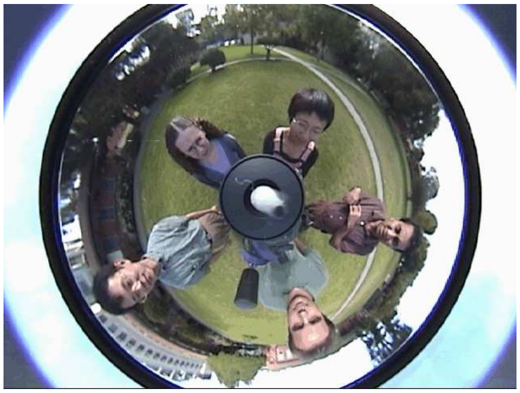
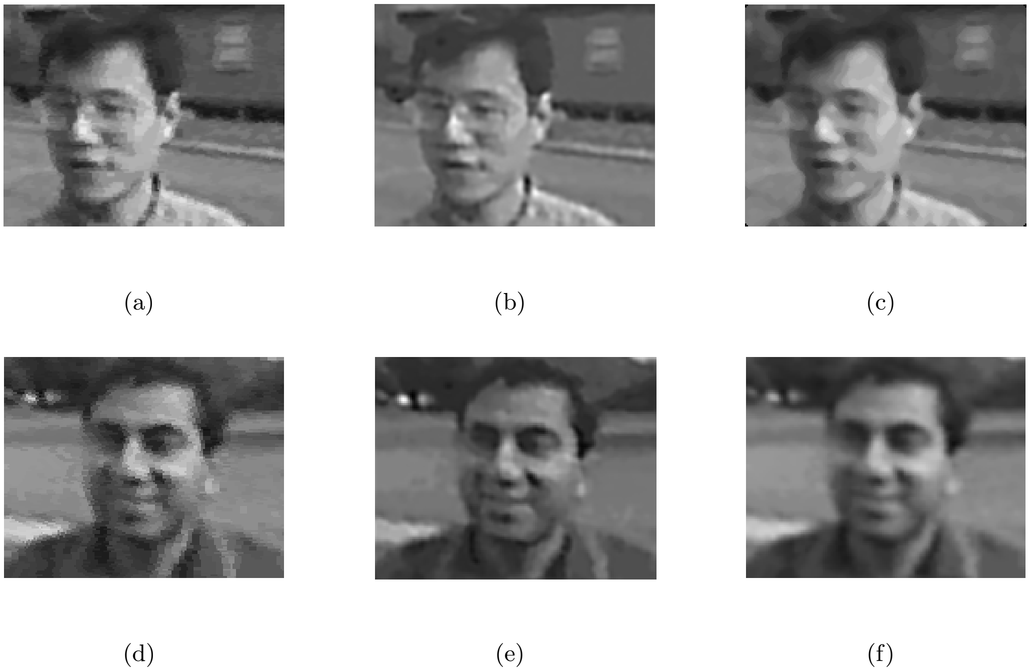


Figure VI.9: Example image from the ODVC.

We study the performance for face video from omni-directional video camera, which is from our NOVA system [10, 140]. The ODVC images are projected onto a pseudo-perspective plane. Face areas are segmented by *skintone* and the tracking procedure gives a stable output for faces. Output from ODVC is low-resolution and non-uniformly distorted, as shown in Figure VI.10(a) and Figure VI.10(b). We first us an error suppression algorithm (see Appendix) to remove the salt-and-pepper noise and the boundaries from non-uniform distortion. No over-smoothing effect appears from the proposed error suppression algorithm. Results from this step are used as the low-resolution observation. Figure VI.10(b) and Figure VI.10(e) show results from the proposed algorithm. As a widely used filter, median filter is usually used for salt-and-pepper noise removal. However, the results are always over-smoothed. For comparison, the results from the median filter is shown in Figure VI.10(c) and Figure VI.10(f). For each reconstructed HR image, three successive images are used. Experimental results show that the resolution enhanced algorithm provide perceptually favorable results.



(a) (b) (c)

(d) (e) (f)

Figure VI.10: Examples of the two-level resolution enhancement. First column: original images. Second column: integrated results. Third column: results from 5 × 5 median filter noise elimination.

Quantitative Comparison

We compare the performance of the proposed inter-pixel interference elimination algorithm with algorithms proposed in [3] and [4]. To be able to compare with the ground-truth data, we synthesize the low-resolution observation: the video frames are down-sampled by a factor of 2 in each direction and the sub-sampled frames are used as the input. The original high resolution face videos are from the database in [1]. Videos in the database contain substantive facial expressions from various subjects. We use the original video sequences to provide the ground-truth for quantitatively performance evaluation. Examples of the results are shown in Figure VI.11 for perceptual evaluation. Due to copyright, only part of the images are shown. Perceptually, more details are resolved by our algorithm and Borman's algorithm than variants of IBP algorithms. However, Borman's algorithm produces blobby images that are perceptually unappealing. The examples in [65] also exhibit the same problems with the Huber *a-priori*, possibly due to such prior leading to excessive constraints on the high-frequency components. Also, both Borman's algorithm and Zomet's algorithms may cause aliasing due to the assumption on the PSF function, which can be observed from the ability to catch the lip changes. The PSNR is computed for each algorithm on a frame-by-frame basis. Figure VI.12 shows the PSNR curve for the first video sequence in the database. The experiment indicates that the IBP algorithms, including Zomet's robust Super-resolution algorithm, have lower performance. But as indicated in [76], Zomet's algorithm requires that the input low-resolution frames have enough difference to make the median estimate accurate enough. This requirement needs to be satisfied either by using more frames or with more diversity samples. For face videos, one major variation if the change of the facial expressions, which is not in favor of Zomet's algorithm.

The mean PSNR over all frames is computed and displayed as well. Table Table VI.1 gives the mean PSNR over all frames. It indicates that the proposed inter-pixel interference estimation and elimination based algorithm exhibits the

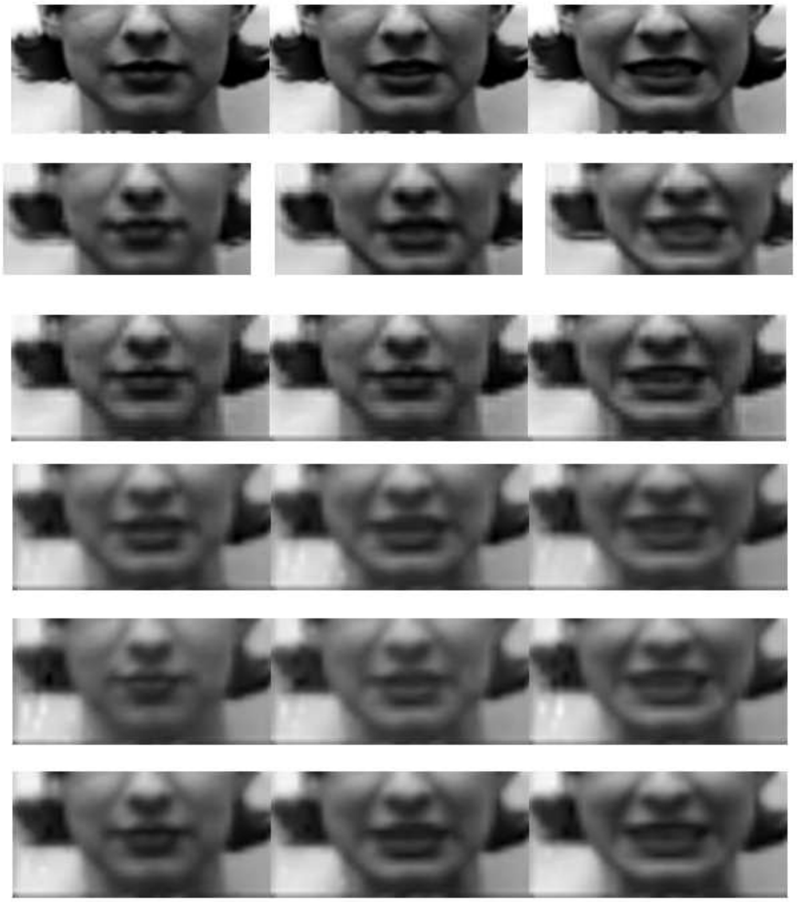


Figure VI.11: First row: the original HR images; second row: from the proposed inter-pixel interference elimination based algorithm; third row: from Borman's algorithm; fourth sixth rows: from different variant of the back-projection algorithm.

least distortion. To get a better understanding of the reconstruction accuracy, we compute the mean MSE over all frames with respect to the frequency. The results in Figure VI.13 show that the proposed inter-pixel interference algorithm has less low-frequency distortion, however, in the high-frequency range, the distortion is greater. Overall, these comparative results show the effectiveness of our algorithm. Also, our algorithm has a lower computational cost than the alternatives.

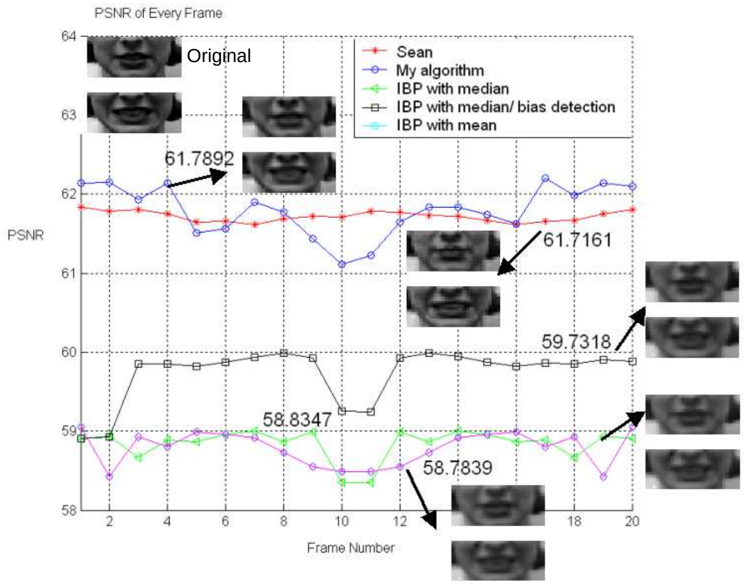


Figure VI.12: Comparison of the PSNR of different algorithms. Blue line: the proposed algorithm; Red line: Borman's algorithm [3]; others: variants of IBP algorithms [4] (Median with bias detection: black. Median; green. Mean: magenta.)

Table VI.1: The mean of each frame's PSNR for different algorithms.

Algorithm	Interpixel Interference Elimination	Borman *et.al.* [3]	Zomet *et.al.* [4], IBP (mean)	Zomet *et.al.* [4], IBP (median)	Zomet *et.al.* [4], IBP (mean+bias)
Mean PSNR	61.80dB	61.72dB	58.78dB	58.84dB	59.73dB

Other Example

Although the motivation of the proposed algorithm is for face video resolution enhancement, the algorithm is not limited to human faces. We apply the proposed algorithm over text subjects to evaluate the performance. First we apply the algorithm over synthetic example. We add random noise to the example and then Gaussian blur the images. Five images are used for the reconstruction. The result is shown in Figure VI.14. Figure VI.14(a) shows initial clean image. Figure VI.14(b) shows the low-resolution input after adding random noise and Gaussian blur. Figure VI.14(c) shows the interpolated observation, which is the

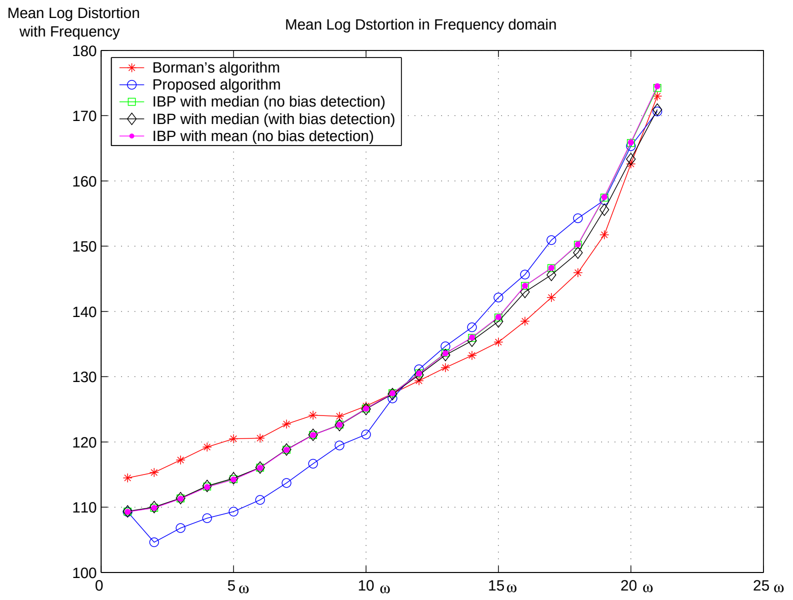
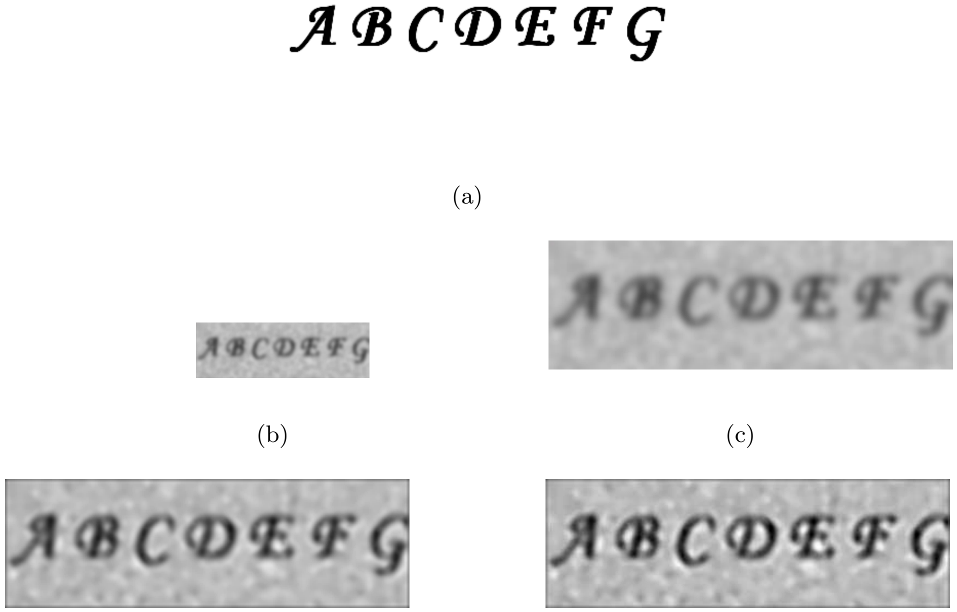


Figure VI.13: Mean log-distortion at different frequencies. Blue line: the proposed algorithm; Red line: Borman's algorithm [3]; others: variants of IBP algorithms [4] (Median with bias detection: black. Median; green. Mean: magenta.)

initial high-resolution estimate. Figure VI.14(d) shows the result after 3 iterations and Figure VI.14(e) shows the result after 6 iterations.

We also compare the performance of the proposed inter-pixel interference elimination based algorithm with Borman's algorithm over real-world text examples. The image sequence is obtained by a webcam. Similarly, the input images are sub-sampled by a factor of 4 in each direction, and then box blurring is applied to get more deteriorate low-resolution images. One example is shown in Figure VI.15. In Figure VI.15(b), the observed low-resolution image is shown. In Figure VI.15(c), the interpolated image is shown, which is the initial high-resolution estimate. In Figure VI.15(d), the reconstructed result from the proposed algorithm is shown. In Figure VI.15(e) and Figure VI.15(f), the reconstructed results from Borman's algorithm and Zomet's algorithm are shown respectively. Experimental evaluation shows that the proposed algorithm performs good over text objects too.

The proposed algorithm is also evaluated over license plate data. The



(a)

(b)

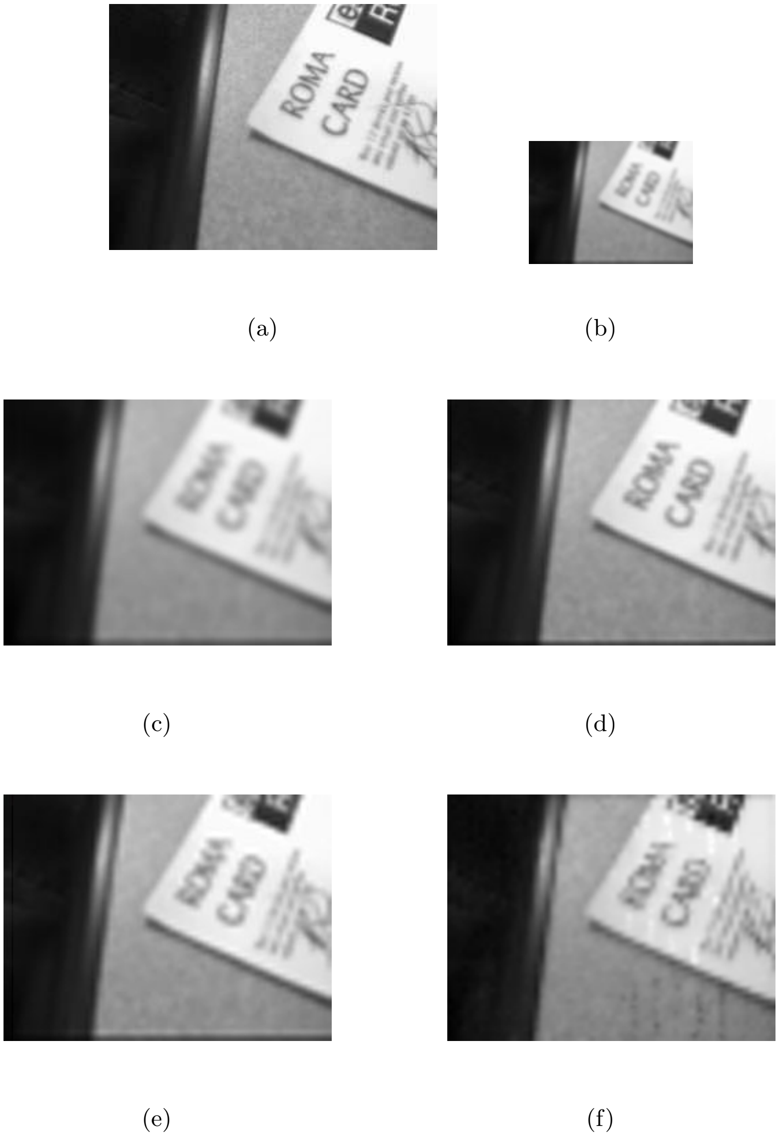
(c)

(d)

(e)

Figure VI.14: The 1st image: original image. The 2nd image: the LR input. The 3rd image: initial high-resolution estimate (interpolation). The 4th one: reconstruction after 3 iterations. The last image: reconstruction after 6 iterations.

license plate data is collected from a video stream. Two examples with different magnification factors are presented in Figure VI.16. Neighboring 3 frames are used for the reconstruction and 4 iterations are used. The super-reconstructed results provides more high-frequency details, which also indicate the potential applications. However, as a reconstruction-based algorithm, the algorithm has a limit on the magnification factor. It can be noticed that the unreadable characters, such as the state, are still not being able to be reconstructed. In [2, 143], more discussions have been made on the limit of the reconstruction based algorithm. One possible solution is to combine with the recognition based approaches [73]; where more details can be recovered while the aliasing can also be suppressed. In the next section we proposed an approach that learns the relevance model between the projected subspace coefficients from the low-resolution images and their counterpart from

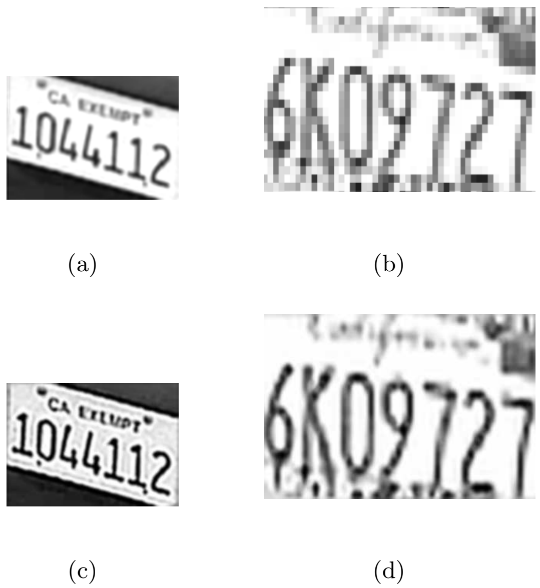


(a) (b)

(c) (d)

(e) (f)

Figure VI.15: Figure VI.15(a)-Figure VI.15(f): the original frame, the LR observation, the initial HR estimate (interpolation); reconstruction from proposed algorithm; reconstruction from Borman's algorithm; reconstruction from Zomet's algorithm.



(a) (b)

(c) (d)

Figure VI.16: Example for license plate. 3 successive images are used for the reconstruction. Figure VI.16(a) and Figure VI.16(b) are the original license plate data; Figure VI.16(c) and Figure VI.16(d) are the corresponding super-resolved results.

the high-resolution images, where this relevance model is used to reconstruct the high-resolution image from the test image's low-resolution subspace coefficients.

VI.B A Relevance Model in TensorPCA Subspace

TensorPCA as a compact and efficient feature subspace has attracted many research interests. In this work, we propose to use tensorPCA subspace as the face representation, which is a specific case of the Concurrent Subspace Analysis described in [85]. The review of tensorPCA is given in Section IV.C.2. The relevance model between the projected tensors from the low-resolution images and their counterpart from the high-resolution images is studied. A regression model is proposed for this purpose based on a Maximum-likelihood estimation framework. This is still based on a global face representation, which makes the algorithm capable for partially occluded images.

VI.B.1 Algorithm Framework: Regression in Subspace

Given a training image set $\{\mathcal{X}^h, \mathcal{X}^l\}$, where:

$$\mathcal{X}^h = \{\mathbf{X}_i^h\}_{i=1,\cdots,n} \quad \text{and} \quad \mathcal{X}^l = \{\mathbf{X}_i^l\}_{i=1,\cdots,n}.$$

$\mathbf{X}_i^h \in \mathcal{R}^{M_1 \otimes M_2}$ is a high-resolution face image and $\mathbf{X}_i^l \in \mathcal{R}^{N_1 \otimes N_2}$ is its corresponding low-resolution image. The tensor subspace for $\mathcal{X}^h$ and $\mathcal{X}^l$ can be obtained individually, where the details for the tensorPCA subspace computation are given in Section IV.C.2. $\check{\mathbf{U}}^h$ and $\check{\mathbf{U}}^l$ are the left projection matrices, while $\check{\mathbf{V}}^h$ and $\check{\mathbf{V}}^l$ are the right projection matrices for $\mathcal{X}^h$ and $\mathcal{X}^l$ respectively. We have $\check{\mathbf{U}}^h \in \mathcal{R}^{M_1 \otimes P_1}$, $\check{\mathbf{V}}^h \in \mathcal{R}^{M_2 \otimes P_2}$, $\check{\mathbf{U}}^l \in \mathcal{R}^{N_1 \otimes Q_1}$ and $\check{\mathbf{V}}^l \in \mathcal{R}^{N_2 \otimes Q_2}$; where P_1, P_2, Q_1, Q_2 are the rank for each projection matrix controlled by a parameter α. The tensorPCA gives:

$$\mathbf{Y}_i^h = \check{\mathbf{U}}^{h\mathrm{T}} \mathbf{X}_i^h \check{\mathbf{V}}^h;$$

$$\mathbf{Y}_i^l = \check{\mathbf{U}}^{l\mathrm{T}} \mathbf{X}_i^l \check{\mathbf{V}}^l. \tag{VI.27}$$

Therefore, two sets of tensorPCA subspace projection are obtained, which are $\mathbf{Y}_i^h \in \mathcal{R}^{P_1 \otimes P_2} = [y_{s,t}^h]_i$ and $\mathbf{Y}_i^l \in \mathcal{R}^{Q_1 \otimes Q_2} = [y_{s,t}^l]_i$ respectively. Here we use $[y_{s,t}]$ to represent a matrix with $y_{s,t}$ as its (s,t)-th entry.

We model the co-occurrent model between the high-resolution image and its low-resolution counterpart in the tensorPCA subspace projection domain. The model is denoted as: $\mathbf{Y}_i^h = f(\mathbf{Y}_i^l)$. When a generative model is used, f is actually a probability. Instead of a general probability model, we consider the conditional probability: $\mathrm{P}(\mathbf{Y}_i^h | \mathbf{Y}_i^l)$. Then the reconstruction problem becomes a Maximum-Likelihood (ML) estimation problem. When a new testing image $\mathbf{X}^l$ is provided, the high-resolution tensorPCA subspace projection is given by:

$$\tilde{\mathbf{Y}}^h = \arg\max_{\mathbf{Y}} \mathrm{P}(\mathbf{Y} | \mathbf{Y}^l). \tag{VI.28}$$

The high-resolution image can be reconstructed by back-projection from the tensorPCA subspace into the image tensor space using:

$$\tilde{\mathbf{X}}^h = (\check{\mathbf{U}}^{h\mathrm{T}})^+ \tilde{\mathbf{Y}}^h (\check{\mathbf{V}}^h)^+; \tag{VI.29}$$

where $(\bullet)^+$ is the pseudo-inverse.

Linear Component

TensorPCA subspace projection is actually formed by the coefficients of decomposing the original tensor onto the set of tensor basis. Since the set of basis are formed from two disjoint sets of ortho-normal vectors $\check{\mathbf{U}}$ and $\check{\mathbf{V}}$, it is reasonable to assume that the correlation between the decomposition coefficients can be suppressed. Thus each individual coefficient can be estimated separately. We have:

$$\hat{y}^h_{s,t} = \arg\max_{y_{s,t}} \mathrm{P}(y_{s,t}|\mathbf{Y}^l).$$

This probability can be further simplified from the assumption of low-correlation between the coefficients in $\mathbf{Y}^l$:

$$\begin{aligned} \mathrm{P}(y_{s,t}|\mathbf{Y}^l) &= \mathrm{P}(y_{s,t}|y^l_{1,1}, \cdots, y^l_{Q_1,Q_2}) \\ &\approx \mathrm{P}(y_{s,t}|y^l_{1,1}) \cdots \mathrm{P}(y_{s,t}|y^l_{Q_1,Q_2}). \end{aligned} \tag{VI.30}$$

Rewrite the tensor $\mathbf{Y} = [y_{s,t}]$ and $\mathbf{Y}^l = [y^l_{s,t}]$ into vector form by rastering order, Eq. VI.30 now becomes:

$$\mathrm{P}(y_s|\mathbf{Y}^l) \approx \prod_{p=1}^{Q_1Q_2} \mathrm{P}(y_s|y^l_p).$$

We use the following Gaussian to model the probability $\mathrm{P}(y_s|y^l_p)$:

$$\mathrm{P}(y_s|y^l_p) = c_s \exp\{-\frac{(y_s - w_{s,p}y^l_p)^2}{2}\}; \tag{VI.31}$$

where c_s is a normalization factor. This Gaussian model evaluates the weighted distance between the projection coefficients. Then we have

$$\mathrm{P}(y_s|\mathbf{Y}^l) \approx c \exp\{-\sum_{p=1}^{Q_1Q_2} \frac{(y_s - w_{s,p}y^l_p)^2}{2}\}. \tag{VI.32}$$

The ML estimate gives:

$$\hat{y}^h_s = \arg\max_{y_s} \log \mathrm{P}(y_s|\mathbf{Y}^l), \tag{VI.33}$$

From Eq. VI.32 and Eq. VI.33, we can get:

$$\hat{y}_s^h = \sum_{p=1}^{Q_1Q_2} w'_{s,p} y_p^l; \tag{VI.34}$$

where

$$w'_{s,p} = \frac{w_{s,p}}{Q_1Q_2}.$$

This is actually a linear regression model.

Each training image provides one equation to find $w'_{s,p}(s = 1, \cdots, P_1P_2, p = 1, \cdots, Q_1Q_2)$. Let 1). the column vector formed by the s-th projection coefficients for the high-resolution images be $\mathbf{y}_s^h$; 2). the column vector formed by the p-th projection coefficients for the corresponding low-resolution set be $\mathbf{y}_p^l$; then n images can give:

$$\mathbf{y}_s^h = [\mathbf{y}_1^l, \cdots, \mathbf{y}_{Q_1Q_2}^l]\mathbf{w}_s;$$

where $\mathbf{w}_s = [w'_{s,1}, \cdots, w'_{s,Q_1Q_2}]^{\mathrm{T}}$. Then $\mathbf{w}_s$ can be given by an Least-Square (LS) estimate:

$$\mathbf{w}_s = \mathbb{Y}^{l+}\mathbf{y}_s^h; \tag{VI.35}$$

$$\mathbb{Y}^l = [\mathbf{y}_1^l, \cdots, \mathbf{y}_{Q_1Q_2}^l].$$

The dimension of the projection matrices is a trade-off between the computational cost and the information preserved. Greater P_1, P_2, Q_1 and Q_2 preserve more training data information at the cost of more computations.

Nonlinear Component

After the approximation step using the independent assumption, the ML estimator is simply a linear model. Higher-order statistics are not used. To refine the approximation, we compensate the the nonlinear components back as y^h's residue estimation. Inspired from the linear regression model, we use Gaussians to model it. Therefore, y_s^h's estimation can be rewritten as:

$$y_s^h = \hat{y}_s^h + e_s^h, j = 1, \cdots, P_1P_2;$$

$$e_s^h = \sum_{p=1}^{Q_1Q_2} \omega_{s,p} g(y_p^l); \tag{VI.36}$$

where

$$g(y_p^l) = \exp\{-\frac{(y_p^l - \mu_p)^2}{2}\};$$

and μ_p is the sample mean of the p-th projection coefficients for all low-resolution training images. This is an RBF-type regression model. However, the center of the each node is fixed so as to reduce the number of the unknown parameters. Sample mean is a reasonable assumption of the center.

The vector set $\{\omega_{s,p}\}$ model the high-order statistical relationships between the the low-resolution input and the high-resolution estimate. Similarly, n training images give:

$$\mathbf{e}_s^h = [\mathbf{g}_1^l, \cdots, \mathbf{g}_{Q_1Q_2}^l]\boldsymbol{\omega}_s;$$

where $\mathbf{g}_p^l$ is the column vector formed by the function $g(y_p^l)$ from all the low-resolution training samples and $\boldsymbol{\omega}_s = [\omega_{s,1}, \cdots, \omega_{s,Q_1Q_2}]^{\mathrm{T}}$. Then LS estimation gives:

$$\boldsymbol{\omega}_s = \mathbb{G}^{l+}\mathbf{e}_s^h; \tag{VI.37}$$

$$\mathbb{G}^l = [\mathbf{g}_1^l, \cdots, \mathbf{g}_{Q_1Q_2}^l].$$

Altogether, the estimate of the s-th high-resolution projection coefficient is:

$$\tilde{y}_s^h = \sum_{p=1}^{Q_1Q_2} w'_{s,p} y_p^l + \sum_{p=1}^{Q_1Q_2} \omega_{s,p} g(y_p^l). \tag{VI.38}$$

And the high-resolution image can be reconstructed using Eq. VI.29, while $\tilde{\mathbf{Y}}^h$ is a tensor with entry $\tilde{y}_s^h$.

VI.B.2 Experimental Evaluation

In this section, the performance of the proposed regression in tensorPCA based face super-resolution algorithm is evaluated. We first evaluate the performance of the super-resolution reconstruction both qualitively and quantitatively using face images from FERET database [91]. In the second section, we compare the performance using the same regression model with the traditional PCA projection. In the third section, the performance for super-resolution reconstruction using partially occluded input is given.

Evaluation on Images from FERET Database

Face images from the FERET database are cropped and normalized. Each image is resized to 160 × 160. These cropped and normalized images are divided into two parts, the training sample set and the testing sample set. For both sets, the images are down-sampled by a factor of 4 in both directions; which are the low-resolution input.

For the training set, we first compute the tensorPCA subspace projections for both the high-resolution images and the low-resolution images. The proposed regression model is learned. In Figure VI.17, some example results from the testing set are shown. We use different α to control the information preserved. Larger α can preserve more information at the expense of more computational cost. We use PSNR to evaluate the performance of the reconstruction. PSNR under different α are computed for comparison, which is shown in Figure VI.19.

Comparison with Traditional PCA

In this section, we compare the performance of using the same regression model for the tensorPCA as well as the traditional PCA. Same protocol is used for comparison with the same α. The reconstruction in the traditional PCA requires high-dimension matrix operation; hence the computational cost and the memory demand is much larger than that of the tensorPCA. Figure VI.21 shows the performance comparison. When $\alpha = 0.999$, the regression model in the tensorPCA gives reconstruction with low aliasing while the results from the traditional PCA are still noisy. In Figure VI.19, the mean PSNR for the traditional PCA using the

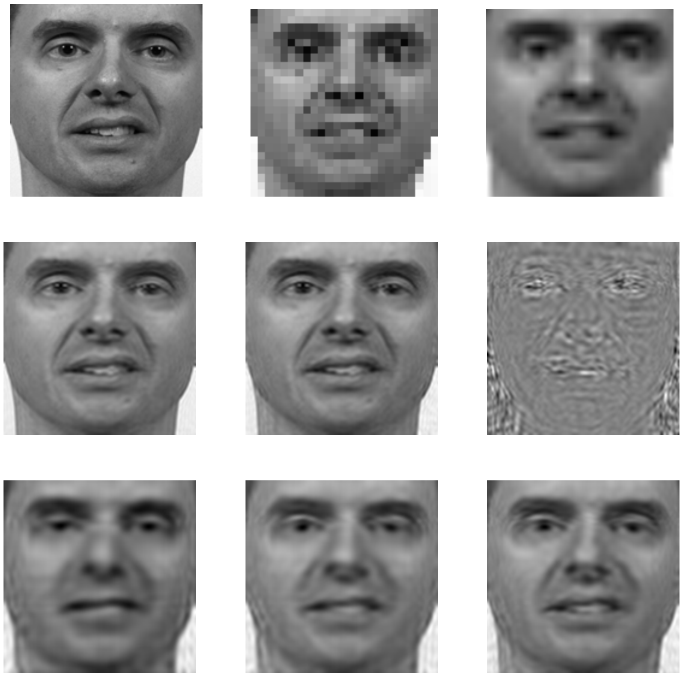


Figure VI.17: Top (left to right): original image, LR input and bilinear interpolation; middle: reconstruction results($\alpha = 0.99$). From left to right: linear + RBF; linear only; and RBF only. Bottom: reconstruction results ($\alpha = 0.96, 0.97, 0.98$).

proposed regression model is also shown. The PSNR is defined as:

$$\text{PSNR} = \frac{(I_o - I_s)^2}{(I_o + 1)^2}; \qquad \text{(VI.39)}$$

where I_o is the original pixel value and I_s is the corresponding super-resolved image pixel value.

Performance for Partially Occluded Face Images

The regression model in the tensorPCA subspace also provides an ability to deal with the partially occluded face image. Since the tensorPCA subspace analysis is still a holistic analysis method, we use a block as well as a mosaic effect for the partially occluded image patch. The mosaic effect is obtained by a severely down-sampling of the occluded area. The occluded image is down-sampled and used as the low-resolution input. Example of the reconstructed result is shown in

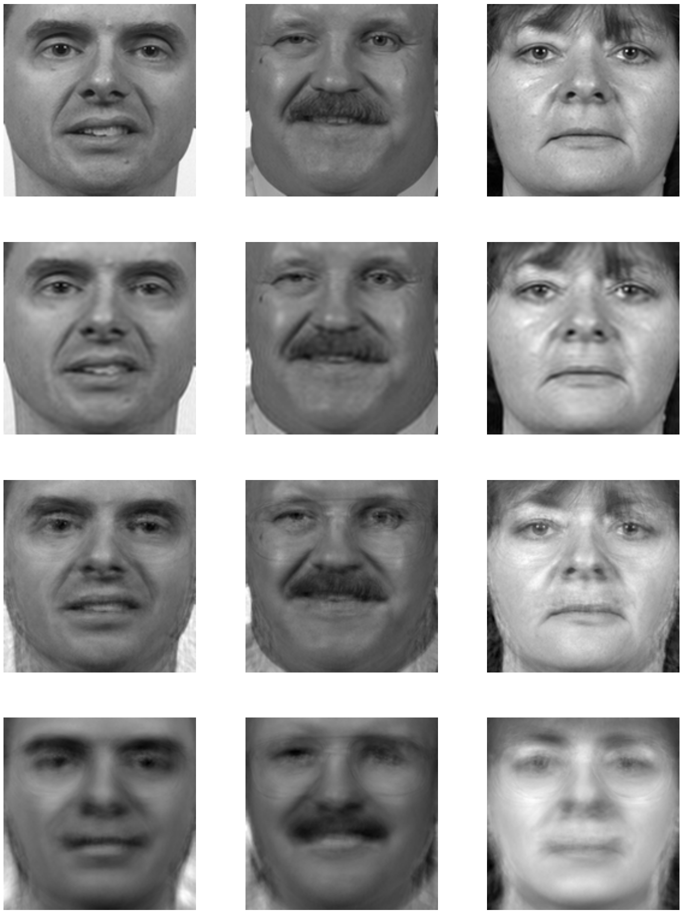


Figure VI.18: TensorPCA vs Traditional PCA. 1st row: original face images; 2nd row: reconstruction with tensorPCA ($\alpha = 0.99$); 3rd row and 4th row: reconstruction with traditional PCA ($\alpha = 0.999, 0.99$).

Figure VI.20. The top row shows result from a block occlusion, while the bottom one shows result from a mosaic occlusion. Good reconstruction can be obtained. However, the occlusion also cause a uniform deteriorate over the whole image; which is reasonable because of the global property of the subspace analysis.

Appendix: Noise Removal for OVDC Output

Due to the non-uniform sampling characteristic of the omni-directional video camera, the generated face video suffers from blocky effect, as well as salt-and-pepper type noises. We use a two-level scheme to enhance the resolution. The

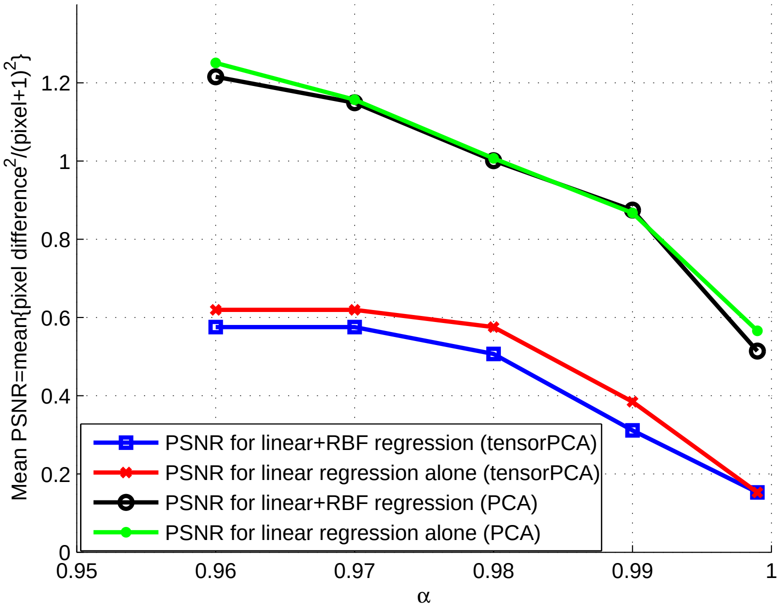


Figure VI.19: Mean PSNR for the testing samples under different α.

first level is a preprocessing step, which is used to suppress the false high-frequency component from the salt-and-pepper noise as well as the blocky effect. Although some traditional filters, for example, median filter, can reduce such noise, the useful high frequency information can also be filtered out; therefore the generated images are over-smoothed. We propose to use a statistical approach instead. The blocky effect of the perspective-projected images introduces the inconsistency of the local statistics, which can be rectified by the the statistics from clean face images. The inconsistent boundary has low probability, which will be removed during the iteration procedure.

We assume that the transformed pseudo-perspective images $I_P(x, y)$ bear the same local statistical property as the clean face images $I_{ref}(x, y)$. The inconsistent boundary can be suppressed by correcting $I_P(x, y)$ using statistics from $I_{ref}(x, y)$. We use pixels' posterior conditioned on their neighborhoods as the local statistics descriptor. The procedure is done iteratively per-pixel basis.

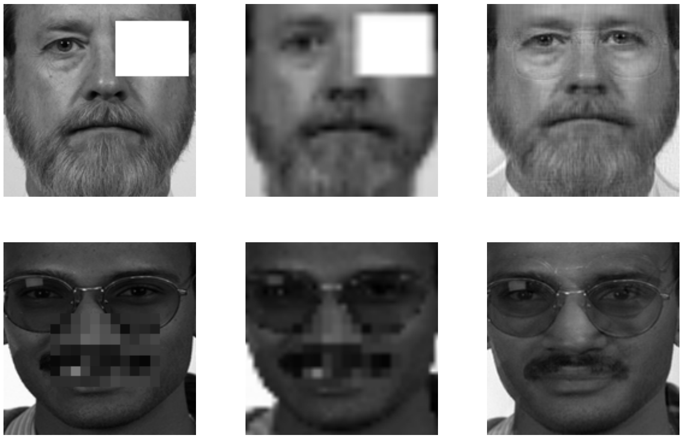


Figure VI.20: Top row: left eye occluded. Bottom row: mouth occluded. Left column: Original images; Middle column: bilinear interpolated partially occluded LR input using the mosaic effect; Right column: reconstructed with $\alpha = 0.99$.

Given (x_i, y_i) as the current pixel to be processed,

$$\mathbf{X} = \{(x_{i_1}, y_{i_1}), \cdots, (x_{i_K}, y_{i_K})\}$$

is the set of its neighborhood. When processing (x_i, y_i), we assume the pixels at $\mathbf{X}$ have the correct intensities. The posterior of $I_P(x_i, y_i)$ conditioned on $\mathbf{X}$ can be written as :

$$\Pr(I_P(x_i, y_i) = g|\mathbf{X}) = \frac{\Pr(\mathbf{X}|I_P(x_i, y_i) = g)\Pr(I_P(x_i, y_i) = g)}{\Pr(\mathbf{X})}. \qquad \text{(VI.40)}$$

We estimate the probability of $I_P(x_i, y_i)$ according to the probability learned from the reference face images I_{ref} so that the posterior estimated from I_P can be as similar as that from I_{ref}. We sample the high-resolution clean face images randomly to get the training set for probability learning. Each sample is formed by a neighborhood containing K samples. K-means clustering is used first to group the samples into similar patterns. For each pattern, a Gaussian mixture is used to model the conditional probabilities: $\Pr(\mathbf{X}_{ref}|I_{ref}(x_i, y_i) = g)$, where: $\mathbf{X}_{ref} = [I_{ref}(x_{i_1}, y_{i_1}), \cdots, I_{ref}(x_{i_K}, y_{i_K})]$.

We use a semi-exhaustive strategy to save the computational cost. For each pattern, we use a Gaussian mixture model for samples with the central pixel's intensity $I_{ref}(x_i, y_i) = g$ in a given range: $g \in \mathcal{B}_m = [a_m, b_m)$. Accordingly, $\Pr(I_{ref}(x_i, y_i) = g)$ becomes $\Pr(\mathcal{B}_m)$. Here we use $a_m = 16(m-1); b_m = 16m - 1; m = 1, \cdots, 16$. The probability is:

$$\Pr(\mathbf{X}_{ref} | I_{ref}(x_i, y_i) = g \in \mathcal{B}_m) \propto \sum_{i_m=1}^{C_m} \beta_{i_m} \mathrm{GNN}(K_{i_m}, \boldsymbol{\alpha}_{i_m}, \boldsymbol{\mu}_{i_m}, \boldsymbol{\Sigma}_{i_m}); \qquad \text{(VI.41)}$$

where $\mathrm{GNN}(K_{i_m}, \boldsymbol{\alpha}_{i_m}, \boldsymbol{\mu}_{i_m}, \boldsymbol{\Sigma}_{i_m})$ is a Gaussian mixture with K_{i_m} Gaussians. $\boldsymbol{\alpha}_{i_m}$, $\boldsymbol{\mu}_{i_m}$, $\boldsymbol{\Sigma}_{i_m}$ are the Gaussian mixture parameters. C_m is the number of clusters from K-means clustering. β_{i_m} is the weight for the i-th cluster. Expectation-maximization (EM) algorithm is applied afterwards to estimate the parameters. The order of the model, which is represented by K_{i_m}, is determined by checking the singularity of the covariance matrices. If the model is over-fitting, the covariance matrices of some Gaussian will become singular. For the Gaussian whose covariance matrix is close to singular, it is merged to the Gaussian with the closest mean. K_{i_m} is decreased by 1 accordingly. The procedure is repeated until there is no ill-conditioned $\boldsymbol{\Sigma}_{m,k}$ and EM converges. Equation (VI.40) can be written as:

$$\Pr(I_{ref}(x_i, y_i) \in \mathcal{B}_m | \mathbf{X}_{ref}) = \Pr(\mathbf{X}_{ref} | I_{ref}(x_i, y_i) \in \mathcal{B}_m) \Pr(\mathcal{B}_m). \qquad \text{(VI.42)}$$

The most probable bin will be given by MAP estimate:

$$\hat{m} = \arg\max_m \Pr(I_{ref}(x_i, y_i) \in \mathcal{B}_m | \mathbf{X}_{ref}). \qquad \text{(VI.43)}$$

The most probable intensity value in the bin $\mathcal{B}_{\hat{m}}$ can be estimated afterwards. We use Pazen window method to model the distribution of the pixel intensity in a given bin:

$$\Pr(g | g \in \mathcal{B}_{\hat{m}}) = \sum_{i=0}^{15} f_{\hat{m},i} \mathcal{W}(g). \qquad \text{(VI.44)}$$

$\mathcal{W}(g)$ is the window function. $f_{\hat{m},i}$ is the normalized frequency:

$$f_{\hat{m},i} = \frac{\#(g = a_m + i)}{\#(g \in \mathcal{B}_{\hat{m}})};$$

where # represents the frequency of an event. $\mathcal{W}(x)$ is a Gaussian window function $\mathcal{N}(g; a_m + i, 1)$. The most probable intensity is: $\arg\max_g \Pr(g | g \in \mathcal{B}_{\hat{m}})$.

With the most probable intensity, the image $I_P(x,y)$ can be corrected. Given the intensity of the current pixel $I_P(x_i,y_i)$: $I_P(x_i,y_i) = g \in \mathcal{B}_{m_0}$. If the pixel value has a large deviation from its estimated most probable intensity, the pixel is most likely on the boundaries between blocks. Therefore, we correct its intensity as follows:

$$\hat{g} = \begin{cases} g + 2\Delta, & \text{if } \hat{m} > m_0, \\ g - 2\Delta, & \text{if } \hat{m} < m_0, \\ g + \Delta, & \text{if } \hat{m} = m_0 \ \& \arg\max_g \mathcal{J}(g|\hat{m}) > g, \\ g - \Delta, & \text{if } \hat{m} = m_0 \ \& \arg\max_g \mathcal{J}(g|\hat{m}) < g, \\ g, & \text{otherwise.} \end{cases} \tag{VI.45}$$

Δ is the updating step. In our implementation, Δ is taken as 1. The whole image is scanned and updated pixel-by-pixel in a raster-scan order. The procedure is repeated for a given times.

The false high-frequency component from the blocky effect is suppressed effectively. The proposed algorithm is tested on gray-scale human face videos obtained from our NOVA system. Figure VI.21 gives some examples of the experimental results. In the second step, the above high-frequency component compensation algorithm is applied on the false high-frequency component suppression results to enhance the resolution.

The text of this chapter, in part, is a reprint of the material as it appears in: Junwen Wu, Mohan M. Trivedi and Bhaskar Rao, “High Frequency Component Compensation based Super-Resolution Algorithm for Face Video Enhancement.” in *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, pp598-601, Aug. 2004, and Junwen Wu and Mohan M. Trivedi, “Enhancement for Face Video from Omni-directional Video Camera”, in *Proceedings of the 38th Asilomar Conference on Signals, Systems and Computers*, Asilomar, CA, November, 2004, and Junwen Wu and Mohan M. Trivedi. “Resolution Enhancement by Inter-Pixel Interference Elimination”, In Press, *Journal of Electronic Imaging*,

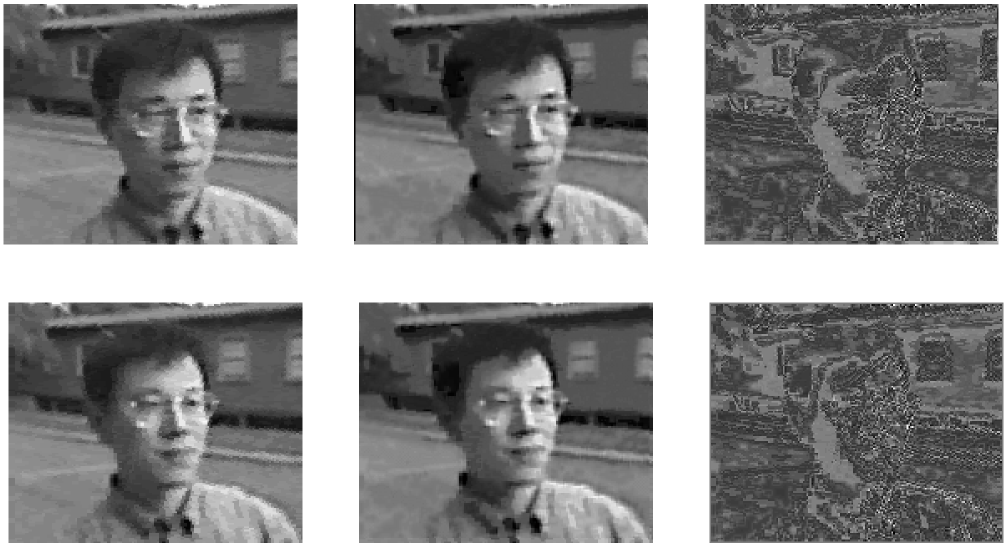


Figure VI.21: The 1st column : original unwarpped face; the 2nd column : results after the false high-frequency component suppression; the 3rd column : difference between the original image and processed result.

16(1), 2007, and Junwen Wu and Mohan M. Trivedi, “A Regression Model in TensorPCA Subspace for Face Image Super-Resolution Reconstruction”, in *Proceedings of the IEEE International Conference of Pattern Recognition (ICPR)*, August, 2006. I was the primary researcher of the cited material and the co-author listed in these publication directed and supervised of the research which forms a basis for this chapter.

Chapter VII

Discussion and Concluding Remarks

“Smart environment” has attracted lots of research interest from the computer vision community. Among these research efforts, head gesture recognition and activity analysis is one of the most important research topics. The recognition of head gesture and behaviors usually requires the knowledge of the individual visual cues, which appear to have a multi-level structure. The study includes the finer detail analysis like facial feature detection and tracking, blink pattern recognition and eye gaze analysis, as well as the coarser level head pose estimation. Depending on the application context, the most appropriate visual module should be used. For example, in a coarser level, only the head orientation is obtainable; while in a finer level, eye dynamics can be observed for providing additional useful information. Correspondingly, this motivates the study of the better models for accurate and robust visual cue extraction, as well as the necessity of defining the most appropriate visual modules and interpreting the obtained visual information for head gesture and behavior analysis. In this work, we devoted to solving the first task: robust visual cue extraction algorithms. We studied the problems of facial feature detection, tracking, blink pattern recognition and head pose estimation.

The facial feature detection and localization problem can be solved by a general object representation and detection algorithm, whose performance is evaluated by the problem of eye detection and localization. A binary tree is used to model the statistical structure of the object’s feature space, so as to obtain a more accurate probability model. After the eyes are automatically located, a particle filter based approach is used to simultaneously track eyes and detect blinks. We used two interactive particle filters for this purpose, each particle filter serves to track the eye localization by exploiting AR models for describing the state transition and a classification based model in tensor subspace for measuring the observation. The performance is evaluated from both the blink detection rate and the tracking accuracy perspectives. Data collected under varying scenarios are used to evaluate the blink detection accuracy, and experimental set-up for acquiring benchmark data to evaluate the accuracy is presented and the experimental results are shown, which show that the proposed algorithm is able to accurately track eye locations, detect both voluntary long blinks and involuntary short blinks and accurately recover blink durations.

Head pose as a most informative visual cue for subjects' attentiveness analysis has attracted enormous attention. In this work we discussed a two-stage approach for estimating face pose from a single static image. The rationale for this approach is the observation that visual cues characterizing facial pose has unique multi-resolution spatial frequency and structural signatures. For effective extraction of such signatures, we use statistical subspaces analysis in Gabor wavelet domain. For systematic analysis of the finer structural details associated with facial features, we employ semi-rigid bunch graphs. It solves the internal problem of the statistical analysis approach that requires a well-aligned and properly-sized data set, as well as introducing the methodology of decomposing a large classification problem into smaller sub-problem for better performance. Extensive series of experiments were conducted to evaluate the pose estimation approach. Experimental results show that this framework also has the potential to infer the continuous pose angles.

Physical constraints, such as the bandwidth limit, impose additional difficulties to the problem of analyzing head and face visual activities. By working on revealing more facial details from the low-resolution images, resolution enhancement algorithms have potential applications for "smart environment" applications, such as facial expression recognition and face recognition [14, 70, 71, 72, 73, 74].

The work summarizes our research accomplishment for a general multilevel visual cues extraction scheme. However, in order to realize a real head gesture recognition system, such visual cues need to be interpreted in a proper way. The following problems still need to be solved:

1. Find the most appropriate visual cues for the current application context;
2. Define the semantics between these visual cues and the temporal evolution rules of these visual cues;

3. Interpret the visual cues according to the predefined semantics and the temporal evolution rules.

Further studies should be conducted with such requirements as a guideline.

Bibliography

[1] T. Kanade, J. F. Cohn, and Y. Tian. Comprehensive database for facial expression analysis. In *The 4th IEEE International Conference on Automatic Face and Gesture Recognition (FG'00).*, March 2000.

[2] S. Baker and T. Kanade. Limits on super-resolution and how to break them. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 24(9):1167–1183, September. 2002.

[3] S. Borman and R. Stevenson. Simultaneous multi-frame map super-resolution video enhancement using spatio-temporal priors. In *Proceedings of IEEE International Conference on Image Processing.*, October. 1999.

[4] A. Zomet, A. Rav-Acha, and S. Peleg. Robust super resolution. In *Proceedings of International Conference on Computer Vision and Pattern Recognition (CVPR)*, December. 2001.

[5] M. Pantic, A. Pentland, A. Nijholt, and T. S. Huang. Human computing and machine understanding of human behavior: a survey. In *ICMI '06: Proceedings of the 8th international conference on Multimodal interfaces*, pages 239–248, 2006.

[6] K. Grauman, M. Betke, J. Lombardi, J. Gips, and G. Bradski. Communication via eye blinks and eyebrow raises: Video-based human-computer interfaces. *Universal Access in the Information Society*, 2(4):359–373, November 2003.

[7] N. Kojima, K. Kozuka, T. Nakano, and S. Yamamoto. Detection of consciousness degradation and concentration of a driver for friendly information service. In *Proceedings of the IEEE International Vehicle Electronics Conference 2001*, pages 31–36, 2001.

[8] X. Liu, F. Xu, and K. Fujimura. Real-time eye detection and tracking for driver observation under various light conditions. In *Proceedings of IEEE Intelligent Vehicle Symposium*, pages 18–20, June 2002.

[9] L. Fletcher, L. Petersson, N. Apostoloff, and A. Zelinsky. Vision in and out of vehicles. *IEEE Intelligent Systems Magazine*, June 2003.

[10] K. Huang and M. M. Trivedi. Driver head pose and view estimation with single omnidirectional video stream. In *Proceedings of the 1st International Workshop on In-Vehicle Cognitive Comptuer Vision Systems, in conjunction with the 3rd International Conference on Computer Vision Systems.*, 2003.

[11] Y. Onoe, N. Yokoya, K. Yamazawa, and H. Takemura. Visual surveillance and monitoring system using an omnidirectional video camera. In *Proceedings of the Fourteenth International Conference on Pattern Recognition.*, volume 1, pages 588–592, August. 1998.

[12] M. M. Trivedi, K. S. Huang, and I. Mikic. Dynamic context capture and distributed video arrays for intelligent spaces. *IEEE Transaction on Systems, Man and Cybernetics, Part A*, 35(1):145–163, Jan 2005.

[13] K. Huang and M.M.Trivedi. Networked omnivision arrays for intelligent environment. In *Proceedings of the Applications and Science of Soft Computing IV.*, August. 2001.

[14] K. Huang and M.M.Trivedi. Streaming face recognition using multicamera video arrays. In *Proceedings of the 16th International Conference on Pattern Recognition.*, pages 213–216, August. 2002.

[15] P. Smith, M. Shah, and N. da V. Lobo. Monitoring head/eye motion for driver alertness with one camera. In *Proceedings of the Fifteenth IEEE International Conference on Pattern Recognition*, September 2000.

[16] K. Grauman, M. Betke, J. Gips, and G. R. Bradski. Communication via eye blinks-detection and duration analysis in real time. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, December 2001.

[17] M. Chau and M. Betke. Real time eye tracking and blink detection with usb cameras. Technical report, Boston University Computer Science, April 2005. No. 2005-12.

[18] P.A. Beardsley. Qualitative approach to classifying gaze direction. In *Proceedings of the IEEE Conf on Automatic Face and Gesture Recognition.*, 1998.

[19] R. Stiefelhagen. Tracking focus of attention in meetings. In *Proceedings of the IEEE International Conference on Multimodal Interfaces (ICMI'02).*, 2002.

[20] B. Braathen, M. S. Bartlett, and J. R. Movellan. 3-d head pose estimation from video by stochastic particle filtering. In *Proceedings of the 8th Annual Joint Symposium on Neural Computation.*, 2001.

[21] Y.Li, S.Gong, and H. Liddell. Support vector regression and classification based multi-view face detection and recognition. In *Proceeding of IEEE International Conference on Automatic Face and Gesture Recognition.*, pages 300–305, July 2000.

[22] S.Z. Li, Q.D. Fu, L. Gu, B. Scholkopf, Y.M. Cheng, and H.J. Zhang. Kernel machine based learning for multi-view face detection and pose estimation. In *Proceedings of 8th IEEE International Conference on Computer Vision.*, July 2001.

[23] K. Jia and S. Gong. Hallucinating multiple occluded cctv face images of different resolutions. In *Proceedings of the IEEE International Conference on Advanced Video and Signal based Surveillance*, September 2005.

[24] Rurainsky and P. Eisert. Template-based eye and mouth detection for 3d video conferencing. In *Proceedings of the International Workshop on Very Low Bitrate Video*, pages 23–31, Sep. 2003.

[25] H. Veeraraghavan and N. P. Papanikolopoulos. Detecting driver fatique through the use of advanced face monitoring techniques. Technical report, University of Minnesota, Center for Transportation Studies, 2001.

[26] Q. Ji and X. Yang. Real-time eye, gaze, and face pose tracking for monitoring driver vigilance. *Real-Time Imaging*, 8(5):357–377, October 2002.

[27] M. M. Trivedi, S. Cheng, E. Childers, and S. Krotosky. Occupant posture analysis with stereo and thermal infrared video: Algorithms and experimental evaluation. *IEEE Transactions on Vehicular Technology, Special Issue on In-Vehicle Vision Systems*, 53(6), 2004.

[28] T. Moriyama, T. Kanade, J. F. Cohn, J. Xiao, Z. Ambadar, J. Gao, and H. Imamura. Automatic recognition of eye blinking in spontaneously occurring behavior. In *Proceedings of the International Conference on Pattern Recognition (ICPR'2002)*, volume IV, pages 78–81, 2002.

[29] T. Morris, F. Zaidi, and P. Blenkhorn. Blink detection for real-time eye tracking. *Journal of Network and Computer Applications*, 25(2):129–143, 2002.

[30] J. F. Cohn, J. Xiao, T. Moriyama, Z. Ambadar, and T. Kanade. Automatic recognition of eye blinking in spontaneously occurring behavior. *Behavior Research Methods, Instruments, and Computers*, 2007 (in Press).

[31] R. L. Hsu, M. Abdel-Mottaleb, and A. K. Jain. Face detection in color images. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24(5):696–706, May 2002.

[32] P. Wang, B. G. Matthew, J. Qiang, and W. J. Wayman. Automatic eye detection and its validation. In *Proceedings of the IEEE Workshop on Face Recognition Grand Challenge Experiments (in Conjunction With CVPR)*, June 2005.

[33] K. Nguyen. Differences in the infrared bright pupil response of human eyes. In *Proceedings of the Eye Tracking Research and Applications Symposium*, pages 133–156, Mar 2002.

[34] A. Haro, M. Flickner, and I. Essa. Detecting and tracking eyes by using their physiological properties, dynamics, and appearance. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR 2000)*, 2000.

[35] D. Cooper. Maximum likelihood estimation of markov process blob boundaries in noisy images. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 1:372–384, 1979.

[36] U. Grenander. *Lectures in Pattern Theory I, II, and III.* Springer. 1976-1981.

[37] K.S. Fu. *Syntactic Pattern Recognition.* Prentice-Hall, 1982.

[38] W. Richards and A. Jepson. What makes a good feature? In *Proceedings of the 1991 York Conference on Spatial Vision in Humans and Robots*, pages 89–125, Jan. 1994.

[39] E. Seemann, B. Leibe, K. Mikolajczyk, and B. Schiele. An evaluation of local shape-based features for pedestrian detection. In *Proceedings of the British Machine Vision Conference (BMVC'05)*, September 2005.

[40] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):696–710, Jul. 1997.

[41] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[42] T. Kadir and M. Brady. Scale, saliency and image description. *International Journal on Computer Vision*, 45(2):83–105, 2001.

[43] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 193–199, June 1997.

[44] P. Viola and M. Jones. Robust real-time object detection. In *Proceedings of the International Conference on Computer Vision(ICCV2001)*, 2001.

[45] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(4):349–361, 2001.

[46] H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *International Journal of Computer Vision*, 56(3):151–177, Feb. 2004.

[47] H. Schneiderman. Learning a restricted bayesian network for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2004.

[48] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1), 2005.

[49] S. C. Zhu. Statistical modeling and conceptualization of visual patterns. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 25(6):691–712, 2003.

[50] D. Gorodnichy. Second order change detection, and its application to blink-controlled perceptual interfaces. In *Proceedings of the International Association of Science and Technology for Development (IASTED) Conference on Visualization, Imaging and Image Processing (VIIP 2003)*, pages 140–145, September 2001.

[51] M. Cordea, E. Petriu, N. Georganas, D. Petriu, , and T. Whalen. Real-time 2.5d head pose recovery for model-based video-coding. In *Proceedings of the IEEE Instrumentation and Measurement Technology Conference.*, 2000.

[52] T. Horprasert, Y. Yacoob, and L. S. Davis. An anthropometric shape model for estimating head orientation. In *Proceedings of the 3rd International Workshop on Visual Form*, 1997.

[53] L. Morency, P. Sundberg, and T. Darrell. Pose estimation using 3d view-based eigenspaces. In *Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures, in Conjunction with ICCV2003*, pages 45–52, 2003.

[54] L. Chen, L. Zhang, Y. Hu, M. Li, and H. Zhang. Head pose estimation using fisher manifold learning. In *Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures, in Conjunction with ICCV2003*, 2003.

[55] J. Sherrah, S. Gong, and E. Ong. Understanding pose discrimination in similarity space. In *Proceedings of the The Eleventh British Machine Vision Conference (BMVC1999)*, 1999.

[56] Y. Wei, L. Fradet, and T. Tan. Head pose estimation using gabor eigenspace modeling. In *Proceedings of the IEEE International Conference on Image Processing (ICIP2002)*, volume 1, pages 281–284, 2002.

[57] S. Srinivasan and K.L. Boyer. Head pose estimation using view based eigenspaces. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 4, pages 302–305, 2002.

[58] Y. Fu and T. S. Huang. Graph embedded analysis for head pose estimation. In *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition.*, April 2006.

[59] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models . In *Proceedings of the European Conference on Computer Vision 1998*, volume 2, pages 484–498, 1998.

[60] M. Potzsch, N. Kruger, and C. von der Malsburg. Determination of face position and pose with a learned representation based on labeled graphs. Technical report, Institute for Neuroinformatik, RuhrUniversitat, Bochum, 1996. Internal Report.

[61] V. Kruger and G. Sommer. Efficient head pose estimation with gabor wavelet networks. In *Proceedings of the The Eleventh British Machine Vision Conference (BMVC2000)*, 2000.

[62] L. Wiskott, J. Fellous, N. Krger, and C von der Malsburg. Face recognition by elastic bunch graph matching. In *Proceedings of the 7th International Conference on Computer Analysis of Images and Patterns(CAIP'97)*, 1997.

[63] S. Baker and T. Kanade. Super-resolution optical flow . Technical report, Carnegie Mellon University., 1999.

[64] R. Y. Tsai and T. S. Huang. Multiframe image restoration and registration. *Advances in Computer Vision and Image Processing.*, 1984.

[65] D. P. Capel and A. Zisserman. Super-resolution from multiple views using learnt image models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*, 2001.

[66] A. J. Patti, M. I. Sezan, and A. M. Tekalp. Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time. *IEEE Transactions on Image Processing,*, 6(10):1064–1076, 1997.

[67] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *International Journal of Computer Vision.*, 40(1):24–57, October. 2000.

[68] A. J. Storkey. Dynamic structure super-resolution. In *Advances in Neural Information Processing Systems 15 (NIPS2002).*, pages 1295–1302, 2003.

[69] J. Wu, M. M. Trivedi, and B. Rao. Resolution enhancement by adaboost. In *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR).*, pages 893 – 896, 2004.

[70] D. D. Muresan and T. W. Parks. Optimal face reconstruction using training. In *Proceedings of 8th IEEE International Conference on Image Processing*, volume 3, pages 373–376, 2002.

[71] K. Jia and S. Gong. Multi-modal tensor face for simultaneous super-resolution and recognition. In *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)*, volume 2, pages 1683–1690, 2005.

[72] B. K. Gunturk, A. U. Batur ad Y. Altunbasak, M. H. Hayes III, and R. M. Mersereau. Eigenface-domain super-resolution for face recognition. *IEEE Transactions on Image Processing*, May 2003.

[73] W. Liu, D. Lin, , and X. Tang. Hallucinating faces: Tensorpatch super-resolution and coupled residue compensation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05)*, 2005.

[74] D. Lin, W. Liu, , and X. Tang. Layered local prediction network with dynamic learning for face super-resolution. In *Proceedings of the IEEE International Conference on Image Processing (ICIP05)*, 2005.

[75] S. Borman and R. L. Stevenson. Super-resolution from image sequences - a review. *Midwest Symposium on Circuits and Systems*, 1998.

[76] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar. Fast and robust multi-frame super resolution. *IEEE Transaction on Image Processing*, 13(10):1327–1344, October 2004.

[77] D. P. Capel and A. Zisserman. Super-resolution enhancement of text image sequences. In *Proceedings of IEEE International Conference on Pattern Recognition (ICPR),*, 2000.

[78] R. R. Schultz and R. L. Stevenson. Extraction of highresolution frames from video sequences. *IEEE Transactions on Image Processing,*, 5(6):996–1011, June 1996.

[79] X. Li and M. T. Orchard. New edge directed interpolation. *IEEE Transactions on Image Processing.*, 10(10):1521–1527, October 2001.

[80] D. Ramanan and K. E. Barner. Non-linear image interpolation through extended permutation rank selection filters. In *Proceedings of the IEEE International Conference on Image Processing (ICIP),*, pages 912–915, September 2000.

[81] C. Liu, H. Shum, and C. Zhang. A two-step approach to hallucinating faces: Global parametric model and local non-parametric model. In *Proceedings of the International Conference on Computer Vision(ICCV)*, pages 192–198., 2001.

[82] G. Dedeoglu, T. Kanade, , and J. August. High-zoom video hallucination by exploiting spatio-temporal regularities. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, volume 2, pages 151–158, June 2004.

[83] M. Alex, O. Vasilescu, and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *Proceedings of the European Conference on Computer Vision*, pages 447–460, 2002.

[84] D. Cai, X. He, and J. Han. Subspace learning based on tensor analysis. Technical report, University of Illinois at Urbana-Champaign (UIUCDCS-R-2005-2572), 2005. Department of Computer Science Technical Report No. 2572.

[85] D. Xu, S. Yan, L. Zhang, H. Zhang, Z.Liu, and H. Shum. Concurrent subspace analysis. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 203 – 208.

[86] R. E. Bellman. *Adaptive Control Processes.* Princeton University Press, 1961.

[87] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559 – 572, 1901.

[88] R.O. Duda, P.E. Hart, and D.H. Stork. *Pattern Classification (2nd ed.).* Wiley Interscience, 2000.

[89] A. Bell and T.J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

[90] T. M. Cover and J. A. Thomas. *Elements of Information Theory.*, pages 18–26. New York: Wiley, 1991.

[91] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The feret evaluation methodology for face recognition algorithms. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22:1090–1104, Oct. 2000.

[92] T. D. Rikert, M. J. Jones, and P. Viola. A cluster-based statistical model for object detection. In *Proceedings of the International Conference on Computer Vision*, volume 2, page 1046, 1999.

[93] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995.

[94] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Transactions on Signal Processing*, 50:425–435, Feb 2002.

[95] Y. Rui and Y. Chen. Better proposal distributions: Object tracking using unscented particle filter. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001).* IEEE Computer Society, 2001.

[96] M. Lee, I. Cohen, and S. Jung. Particle filter with analytical inference for human body tracking. In *Proceedings of the IEEE Workshop on Motion and Video Computing*, 2002.

[97] M. Bolic, S. Hong, and P. M. Djuric. Performance and complexity analysis of adaptive particle filtering for tracking applications. In *Proceedings of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 853–857, 2002.

[98] C. Chang and R. Ansari. Kernel particle filter: iterative sampling for efficient visual tracking. In *Proceedings of the International Conference on Image Processing*, volume 3, pages 977–980, 2003.

[99] C. Chang and R. Ansari. Kernel particle filter for visual tracking. *IEEE Signal Processing Letters*, 12(3):242–245, 2005.

[100] A. Giremus, A. Doucet, V. Calmette, and J. Tourneret. A Rao-Blackwellized Particle Filter for INS/GPS Integration . In *Proceedings of the International Conference on Audio, Speech and Signal Processing*, pages 964–967, 2004.

[101] J. S. Liu and R. Chen. Blind deconvolution via sequential imputation. *Journal of the American Statistical Association*, 90:567–576, 1995.

[102] A. Doucet, de Freitas, J.F.G., and N. J. Gordon. *Sequential Monte Carlo Methods in Practice.* New York: Springer-Verlag, 2001.

[103] K. Heine. Unified framework for sampling/importance resampling algorithms. In *Proceedings of the IEEE International Conference on Information Fusion*, volume 2, 2005.

[104] J. S. Liu and R. Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93:1032–1044, 1998.

[105] R. Karlsson. *Particle Filtering for Positioning and Tracking Applications.* PhD thesis, Link02ping University, Link02ping, Sweden, 2005. Dissertation No. 924.

[106] N. J. Gordon, D. J. Salmond, and A.F.M. Smith. A novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings on Radar and Signal Processing*, 140:107–113, 1993.

[107] M. K. Pitt and N. Shephard. Filtering via simulation: auxiliary particle filter. *Journal of the American Statistical Association*, 94:590–599, 1999.

[108] Z. Khan, T. Balch, and F. Dellaert. A rao-blackwellized particle filter for eigen tracking. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, volume 02, pages 980–986, 2004.

[109] S. Sarkka, A. Vehtari, and J. Lampinen. Rao-blackwellized particle filter for multiple target tracking. *Information Fusion*, 2006. In press.

[110] J. Carpenter, P. Clifford, and P. Fernhead. An improved particle filter for non-linear problems. Technical report, Department of Statistics, University of Oxford, 1997.

[111] A. Doucet, S. J. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[112] J. Hol, T. Schon, and F. Gustafsson. On resampling algorithms for particle filters. In *Nonlinear Statistical Signal Processing Workshop*, Cambridge, United Kingdom, Sep 2006.

[113] M Isard and A. Blake. Visual tracking by stochastic propagation of conditional density. In *Proceedings of the 4th European Conference on Computer Vision*, pages 343– 356, April 1996.

[114] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1), 1998.

[115] K. Nishiyama. Fast and effective generation of the proposal distribution for particle filters. *Signal Process.*, 85(12):2412–2417, 2005.

[116] Y. Guan, R. Fleissner, P. Joyce, and S. M. Krone. Markov chain monte carlo in small worlds. *Statistics and Computing*, 16(2):193–202, 2006.

[117] C. Shen, M. J. Brooks, and A. van den Hengel. Augmented particle filtering for efficient visual tracking. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 856–859, 2005.

[118] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *Proceedings of the European COnference on Computer Vision*, pages 28–39, 2004.

[119] X. Xu and B. Li. Head tracking using particle filter with intensity gradient and color histogram. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 888–891, 2005.

[120] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *Proceedings of the European Conference on Computer Vision*, 2002.

[121] C. Yang, R. Duraiswami, and L. Davis. Fast multiple object tracking via a hierarchical particle filter. In *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)*, volume 1, pages 212–219, 2005.

[122] M. Pupilli and A. Calway. Real-time camera tracking using a particle filter. In *Proceedings of the British Machine Vision Conference*, pages 519–528, September 2005.

[123] X. He, D. Cai, and P. Niyogi. Tensor subspace analysis. In *Proceedings of the Neural Information Processing Systems*, 2005.

[124] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290. 22 December 2000.

[125] X. He, S. Yan, Y. Hu, P. Niyogi, and H-J. Zhang. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3), March 2005.

[126] S. Esaki, Y. Ebisawa, A. Sugioka, and M. Konishi. Quick menu selection using eye blink for eye-slaved nonverbal communicator with video-based eye-gaze detection. In *Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology society*, 1997.

[127] D. Goryn and S. Hein. On the estimation of rigid body rotation from noisy data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1219–1220, 1995.

[128] B. Scholkopf, A. Smola, and K.-R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[129] Y. Li, S. Gong, and H. Liddell. Recognising trajectories of facial identities using kernel discriminant analysis. In *Proceedings of the British Machine Vision Conference (BMVC2001)*, pages 613–622, 2001.

[130] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. Muller. Fisher discriminant analysis with kernels. In *Proceedings of the IEEE Neural Networks for Signal Processing Workshop*, pages 41–48, 1999.

[131] J.MacLennan. Gabor representations ofspatiotemporal visual images. Technical report, Computer Science Department, University of Tennessee, Knoxville., 1991. CS-91-144. Accessible via URL: http://www.cs.utk.edu/ mclennan.

[132] J. Ham, D.D. Lee, S. Mika, and B. Scholkopf. A kernel view of dimensionality reduction of manifolds. In *Proceedings of the International Conference on Machine Learning.*, 2004.

[133] L. Shen and L. Bai. Gabor feature based face recognition using kernel methods. In *Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FGR'04).*, 2004.

[134] G. Dai and Y.T. Qian D.Y. Yeung. Face recognition using a kernel fractional-step discriminant analysis algorithm. *Pattern Recognition*, 40(1):229–243, 2007.

[135] M. Savvides, R. Abiantun, J. Heo, S. Park, C. Xie, and B.V.K. Vijayakumar. Partial and holistic face recognition on frgc-ii data using support vector machine. In *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop.*, page 48. IEEE Computer Society, 2006.

[136] S. Yang, S.Yan, D. Xu, X. Tang, and C. Zhang. Fisher+kernel criterion for discriminant analysis. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05).*, volume 2, pages 197–202, 2005.

[137] N. Cristianini, J. Kandola, A. Elisseeff, and J. Shawe-Taylor. On kernel-target alignment. In *Proceedings of the Neural Information Processing System (NIPS).*, pages 367–373, 2001.

[138] R. Kondor and T. Jebara. A kernel between sets of vectors. In *Proceedings of the International Conference of Machine Learning (ICML).*, 2003.

[139] M. Yang. Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. In *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition.*, page 0215. IEEE Computer Society, 2002.

[140] K. Huang and M.M.Trivedi. Video arrays for real-time tracking of person, head, and face in an intelligent room. *Machine Vision and Applications.*, 14(2):103–111, June. 2003.

[141] K. Yamada, T. Ito, and H. Masuda. The omnidirectional vision sensor for in-vehicle image processing applications. In *Proceedings of 1999 International Conference on Image Processing.*, volume 4, pages 11–15, October. 1999.

[142] R. Bajcsy S. Lin. High resolution catadioptric omni-directional stereo sensor for robot vision. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation.*, pages 1694–1699, Sep. 2003.

[143] Z. Lin and H. Shum. Fundamental limits of reconstruction-based superresolution algorithms under local translation. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 26(1):1–15, January. 2004.