Head Pose Estimation and Attentive Behavior Detection

Nan Hu

B.S.(Hons.), Peking University

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF

ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2005

Acknowledgements

I express sincere thanks and gratefulness to my supervisor Dr. Weimin Huang, Institute for Infocomm Research, for his guidance and inspiration throughout my graduate career at National University of Singapore. I am truly grateful for his dedication to the quality of my research, and his insightful prospectives on numerous prospectives on numerous technical issues.

I am very much grateful and indebted to my co-supervisor Prof. Surendra Ranganath, ECE department of Nationl University of Singapore, for his suggestions on the key points of my projects and the helpful comments during my paper work.

Thanks are also due to the I^2R Visual Understanding Lab, Dr. Liyuan Li, Dr. Ruihua Ma, Dr. Pankaj Kumar, Mr. Ruijiang Luo, Mr. Lee Beng Hai, to name a few, for their help and encouragement.

Finally, I would like to express my deepest gratitude to my parents, for the continuous love, support and patience given to me. Without them, this thesis could not have been accomplished. I am also very thankful to friends and relatives with whom I have been staying. They never failed to extend their helping hand whenever I went through stages of crisis.

Contents

Summary

Attentive behavior detection is an important issue in the area of visual understanding and video surveillance. In this thesis, we will discuss the problem of detecting a frequent change in focus of human attention(FCFA) from video data. People perceive this kind of behavior(FCFA) as temporal changes of human head pose, which can be achieved by rotating the head or rotating the body or both. Contrary to FCFA, an ideally focused attention implies that the head pose remains unchanged for a relatively long time. For the problem of detecting FCFA, one direct solution is to estimate the head pose in each frame of the video sequence, extract features to represent FCFA behavior, and finally detect it. Instead of estimating the head pose in every frame, another possible solution is to use the whole video sequence to extract features such as a cyclic motion of the head, and then devise a method to detect or classify it.

In this thesis, we propose two methods based on the above ideas. In the first method, called the head pose estimation(HPE) method, we propose to find a 2-D manifold for each head image sequence to represent the head pose in each frame. One way to build a manifold is to use a non-linear mapping method called the ISOMAP to represent the high dimensional image data in a low dimensional space. However, the ISOMAP is only suitable to represent each person individually; it cannot find a single generic manifold for all the person's low dimensional embeddings. Thus, we normalize the 2-D embeddings of different persons to find a unified head pose embedding space, which is suitable as a feature space for person independent head pose estimation. These features are used in a non-linear person-independent mapping system to learn the

parameters to map the high dimensional head images into the feature space. Our non-linear person-independent mapping system is composed of two parts: 1) Radial Basis Function (RBF) interpolation, and 2) an adaptive local fitting technique. Once we get these 2-D coordinates in the feature space, the head pose is very simply calculated based on these coordinates. The results show that we can estimate the orientation even when the head is completely turned back to the camera. To extend our HPE method to detect FCFA behavior, we propose to use an entropy-based classifier. We estimate the head pose angle for every frame of the sequence, and calculate the head pose entropy over the sequence to determine whether the sequence exhibits either FCFA or focused attention behavior. The experimental results show that the entropy value for FCFA behavior is very distinct from that for the focused attention behavior. Thus by setting an experimental threshold on the entropy value we can successfully detect FCFA behavior. In our experiment, the head pose estimate is very accurate compared with the "ground truth". To detect FCFA, we test the entropy-based classifier on 4 video sequences, by setting an easy threshold, we classify FCFA from focused attention by an accuracy of 100%.

In a second method, which we call the cyclic pattern frequency analysis (CPFA) method, we propose to use features extracted by analyzing a similarity matrix of head pose obtained from the head image sequence. Further, we present a fast algorithm which uses the principal components subspace instead of the original image sequence to measure the self-similarity. An important feature of the behavior of FCFA is its cyclic pattern where the head pose repeats its position from time to time. A frequency analysis scheme is proposed to find the dynamic characteristics of persons with frequent change of attention or focused attention. A nonparametric classifier is used to classify these two kinds of behaviors (FCFA and focused attention). The fast algorithm discussed in this work yields less computational time (from 186.3s to 73.4s for a sequence of 40s in Matlab) as well as improved accuracy in classification of the two types of attentive behavior (improved from 90.3% to 96.8% in average accuracy).

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Recent advancements in the technologies of video data acquisition and computer hardware, both in terms of speed and memory for processing information together with the rapidly growing demand for video data analysis has made intelligent, computer-based visual monitoring an active area of research. In public sites, surveillance systems are commonly used by security or local authorities to monitor events that involve unusual behaviors. The main aim of the video surveillance system is the early detection of unusual situations that may lead to undesirable emergencies and disasters.

The most commonly used surveillance system is the Closed Circuit Television (CCTV) system, which can record the scenes on tapes for the past 24 to 48 hours to be retrieved "after the event". In most of the cases, the monitoring task is done by human operators. Undeniably, human labor is accurate for a short period, and difficult to be replaced by an automatic system. However, the limited attention span and reliability of human observers have led to significant problems in manual monitoring. Besides, this kind of monitoring is very tiring and tedious for human operators, for they have to deal with a wall of split screens continuously and simultaneously to look for suspicious events. In addition, human labor is also costly, slow, and its performance deteriorates when the

amount of data to be analyzed is large. Therefore, intelligent monitoring techniques are essential.

Motivated by the demand of intelligent video analysis system, our work focuses on an important aspect of this kind of system, i.e. attentive behavior detection. Human attention is a very important cue which may lead to better understanding of human's intrinsic behavior, intention or mental status. One example discussed in [24] is about the students' attentive behavior relationship to the teaching method. An interesting, flexible method will attract more attention from students while a repeated task will make it difficult for students to remain attentive. Human's attention is a means to express their mental status [25], from which an abserver can infer their beliefs and desires. The attentive behavior analysis is such a way to mimic the observer's perception to the inference.

In this work, we propose to classify these two kinds of human attentive behaviors, i.e. a frequent change in focus of attention (FCFA) and focused attention. We would expect that FCFA behavior requires a frequent change of head pose, while focused attention means that the head pose will approximately be constant for a relatively long time. Hence, this motivates us to detect the head pose in each frame of a video sequence, so that the change of head pose can be analyzed and subsequently classified. We call this the Head Pose Estimation (HPE) method and present it in the first part of this dissertation. On the other hand, in terms of head motion, FCFA behavior will cause the head to change its pose in a cyclic motion pattern, which motivates us to analyze cyclic motion for classification. In the second part of this dissertation, we propose a Cyclic Pattern Analysis (CPA) method to detect FCFA.

1.2 Applications

In video surveillance and monitoring, people are always interested in the attentive behavior of the observer. Among the many possible attentive behaviors, the most

important one is a frequent change in focus of attention (FCFA). Correct detection of this behavior is very useful in everyday life. Applications can be easily found in, e.g. a remote education environment, where system operators are interested in the attentive behavior of the learners. If they are being distracted, one possible reason may be that the content of the material is not attractive and useful enough for the learners. This is a helpful hint to change or modify the teaching materials.

In cognitive science, scientists are always interested in the response to salient objects in the observer's visual field. When salient objects are spatially widely distributed, however, visual search for the objects will cause FCFA. For example, the number of salient objects to a shopper can be extremely large, and therefore, in a video sequence, the shopper's attention will change frequently. On the other side, when salient objects are localized, visual search will cause human attention to focus on one spot only, resulting in focused attention. Successful detection of this kind of attentive motion can be a useful cue for intelligent information gathering about objects which people are interested in.

In building intelligent robots, scientists are interested in making robots understand the visual signals arising from movements of the human body or parts of the body, e.g. a hand waving and a head nodding, which is a cyclic motion. Therefore, our work can be applied in these areas of research also.

In computer vision, head pose estimation is a research area of current interest. Our HPE method explained later is shown to be successful in estimating the head pose angle even when the person's head is totally or partially turned back to the camera.

In the following we give an overview of our approaches to recognizing human attentive behavior through head pose estimation and cyclic pattern analysis.

1.3 Our Approach

1.3.1 HPE Method

Since head pose will change during FCFA behavior, FCFA can be detected by estimating head pose in each frame of a video sequence and looking at the change of head pose as time evolves. Different head pose images of a person can be thought of as lying on some manifold in high dimensional space. Recently, some non-linear dimensionality reduction techniques have been introduced, including Isometric Feature Mapping (ISOMAP) [18], Locally Linear Embedding (LLE) [20]. Both methods have been shown to be able to successfully embed the hidden manifold in high dimensional space onto a low dimensional space.

In our head pose estimation (HPE) method, we first employ the ISOMAP algorithm to find the low dimensional embedding of the high dimensional input vectors from images. ISOMAP tries to preserve (as much as possible according to some cost function) the geodesic distance on the manifold in high dimensional space while embedding the high dimensional data into a low dimensional space (2-D in our case). However, the biggest problem of ISOMAP as well as LLE is that it is person-dependent, i.e., it provides individual embeddings for each person's data but cannot embed multiple persons' data into one manifold as is described in Chapter 3. Besides, although the appearance of the 2-D embedding of a person's head data is ellipse-like, for different persons, the shape, scale and orientation of the ellipse is different.

To find a person-independent feature space, for every person's 2-D embedding we use an ellipse fitting technique to find an ellipse that can best represent the points. After we obtain the parameters of every person's ellipse, we further normalize these ellipses into a unified embedding space so that similar head poses of different persons are near each other. This is done by first rotating the axes of every ellipse to lie along the X and Y axes, and then scaling every ellipse to a unit circle. Further, by identifying frames which are frontal or near frontal and their corresonding points in

the 2-D unified embedding, we rotate all the points so that those corresponding to the frontal view lie at the 90 degree angle in the X-Y plane. Moreover, since the ISOMAP algorithm can embed the head pose data into the 2-D embedding space either clockwise or anticlockwise, we will take a mirror image along the Y-axis for all the points if the left profile frames of a person are at around 180 degree. This process yields the final embedding space, or a 2-D feature space which is suitable for person independent head pose estimation.

After following the above process for all training data, we propose a non-linear person-independent mapping system to map the original input head images to the 2-D feature space. Our non-linear person-independent mapping system is composed of two parts: 1) a Radial Basis Fucntion (RBF) interpolation, and 2) an adaptive local fitting algorithm. RBF interpolation here is used to approximate the non-linear embedding function from high dimensional space into the 2-D feature space. Furthermore, in order to correct for possible unreasonable mappings and to smooth the output, an adaptive local fitting algorithm is then developed and used on sequences under the assumption of the temporal continuity and local linearity of the head poses. After obtaining the corrected and smoothed 2-D coordinates, we transform the coordinate system from X-Y coordinate to R-Θ coordinate and take the value of θ as the output pose angle.

To further detect FCFA behavior, we propose an entropy classifier. By defining the head pose angle entropy of a sequence, we calculate the entropy value for both FCFA sequences and focused attention sequences. Examining the experimental results, we set a threshold on the entropy value to classify FCFA and focused attention behavior, as discussed later.

1.3.2 CPFA Method

FCFA can be easily perceived by humans as temporal changes of head pose which keeps repeating itself in different orientations. However, as human beings, we probably do not recognize this behavior by calculating the head pose at each time instant but

by treating the whole sequence as one pattern. Contrary to FCFA, an ideally focused attention implies that head pose remains unchanged for a relatively long time, i.e., no cyclicity is demonstrated. This part of work, which we call cyclic pattern frequency analysis (CPFA) method, therefore, is to mimic human perception of FCFA as a cyclic motion of a head and to present an approach for the detection of this cyclic attentive behavior from video sequences. In the following, we give the definition of cyclic motion.

The motion of a point $\overline{X}(t)$, at time t, is defined to be cyclic if it repeats itself with a time varying period $p(t)$, i.e.,

$$\overline{X}(t + p(t)) = \overline{X}(t) + \overline{T}(t), \tag{1.1}$$

where $\overline{T}(t)$ is a translation of the point. The period $p(t)$ is the time interval that satisfies (1.1). If $p(t) = p_0$, i.e., a constant for all t, then the motion is exactly periodic as defined in [1]. A periodic motion has a fixed frequency $1/p_0$. However, the frequency of cyclic motion is time varying. Over a period of time, cyclic motion will cover a band of frequencies while periodic motion covers only a single frequency or at most a very narrow band of frequencies.

Most of the time, the attention of a person can be characterized by his/her head orientation [80]. Thus, the underlying change of attention can be inferred by the motion pattern of head pose changes with time. For FCFA, the head keeps repeating the poses, which therefore demonstrates cyclic motion as defined above. An obvious measurement for the cyclic pattern is the similarity measure of the frames in the video sequence.

By calculating the self-similarities between any two frames in the video sequence, a similarity matrix can be constructed. As shown later, a similarity matrix for cyclic motion differs from that of one with smaller motion such as a video of a person with focused attention.

Since the calculation of the self-similarity matrix using the original video sequence is

very time consuming, we further improved the algorithm by using a principal components subspace instead of the original image sequence for the self-similarity measure. This approach saves much computation time as well as an improved classification accuracy.

To analyze the similarity matrix we applied a 2-D Discrete Fourier Transform to find the characteristics in the frequency domain. A four dimensional feature vector of normalized Fourier spectral values in the low frequency region is extracted as the feature vector.

Because of the relatively small size of training data, and the unknown distribution of the two classes, we employ a nonparametric classifier, i.e., k-Nearest Neighbor Rule (K-NNR), for the classification of the FCFA and focused attention.

1.4 Contributions

The main contribution of our HPE method is an innovative scheme for the estimation of head orientation. Some prior works have considered head pose estimation, but they require either the extraction of some facial features or depth information to build a 3-D model. Facial feature based methods require finding the features while 3-D model-based methods requires either a stereo or multiple calibrated cameras. However, our algorithm works with an uncalibrated, single camera, and can give correct estimate of the orientation even when the person's head is turned back to the camera.

The main contribution of our CPFA method is the introduction of a scheme for the robust analysis of cyclic time-series image sequences as a whole rather than using individual images to detect FCFA behavior. Although there were some works presented by other researchers for periodic motion detection, we believe our approach is new to address the cyclic motion problem. Different from the works in head pose detection, this approach requires no information of the exact head pose. Instead, by extracting the global motion pattern from the whole head image sequence and combining with

a simple classifier, we can robustly detect FCFA behavior. A fast algorithm is also proposed with improved accuracy for this type of attentive behavior detection.

The rest of the dissertation is organized as follows:

- Chapter 2 will discuss the related work, including works on attention analysis, dimensionality reduction, head pose estimation, and periodic motion analysis.
- Chapter 3 will describe our HPE method.
- Chapter 4 will explain our CPFA method.
- Chapter 5 will show the experimental results and give a brief discussion on the robustness and performance of our proposed methods.
- Chapter 6 will present the conclusion and future work.

Chapter 2

Related Work

2.1 Attention Analysis

Computation for detecting attentive behavior has long been focusing on the task of selecting salient objects or short-term motion in images. Most of the research works tried to detect low level salient objects with local features such as edges, corners, color and motion etc.[27, 28, 35, 26]. In contrast, our work deals with the issue of detecting high level salient objects from long-term video sequences, i.e. the attention of an observer when the salient objects to the observer is widely distributed in space. Attentive behavior analysis is an important part of attention analysis, however, it is believed not to have been researched much.

Koch and Itti have built a very sophisticated saliency-based spatial attention model [43, 44]. The saliency map is used to encode and combine information about each salient or conspicuous point (or location) in an image or a scene to evaluate how different a given location is from its surrounding. A Winner-Take-All (WTA) neural network implements the selection process based on the saliency map to govern the shifts of visual attention. This model performs well on many natural scenes and has received some support from recent electrophysiological evidence [55, 56]. Tsotsos et al. [26] presented a selective tuning model of visual attention that used inhibition of

irrelevant connections in a visual pyramid to realize spatial selection and a top-down WTA operation to perform attentional selection. In the model proposed by Clark et al. [30, 31], each task-specific feature detector is associated with a weight to signify the relative importance of the particular feature to the task and WTA operates on the saliency map to drive spatial attention (as well as the triggering of saccades). In [39, 50], color and stereo are used to filter images for attention focus candidates and to perform figure/ground separation. Grossberg proposed a new ART model for solving the attention-preattention (attention-perceptual grouping) interface and stability-plasticity dilemma problems [37, 38]. He also suggested that both bottom-up and top-down pathways contain adaptive weights that may be modified by experience. This approach has been used in a sequence of models created by Grossberg and his colleagues (see [38] for an overview). In fact, the ART Matching Rules suggested in his model tend to produce later selection of attention and is partly similar to Duncan's integrated competition hypothesis [35] which is an object-based attention theory and different from the above models.

Some researchers have exploited neural network approaches to model selective attention. In [27, 28], the saliency maps which are derived from the residual error between the actual input and the expected input are used to create the task-specific expectations for guiding the focus of attention. Kazanovich and Borisyu proposed a neural network of phase oscillators with a central oscillator (CO) as a global source of synchronization and a group of peripheral oscillators (PO) for modelling visual attention [42]. Similar ideas have also been found in other works [33, 34, 45, 46, 47] and are supported by many biological investigations [45, 57, 58]. There are also some models of selective attention based on the mechanisms of gating or dynamic routing information flow by dynamically modifying the connection strengths of neural networks [37, 41, 48, 49].

In some models, mechanisms for reducing the high computational burden of selective attention have been proposed based on space-variant data structures or multiresolution pyramid representations and have been embedded within foveation systems for robot vision [29, 51, 32, 36, 52, 53, 54]. But it is noted that these models developed the overt

attention systems to guide fixations of saccadic eye movements and partly or completely ignored the covert attention mechanisms. Fisher and Grove [40] have also developed an attention model for a foveated iconic machine visual system based on an interest map. The low-level features are extracted from the currently foveated region and top-down priming information are derived from previous matching results to compute the salience of the candidate foveate points. A suppression mechanism is then employed to prevent constantly re-foveating the same region.

2.2 Dimensionality Reduction

The basis for our HPE method is our belief that different head poses of a person will lie on some high dimensional manifold (in the original image space) and can be visualized by embedding it into a 2- or 3-D space, which is also useful to find the features to represent different poses. In recent years, scientists have been working on non-linear dimensionality reduction methods, since classical techniques such as Principal Component Analysis (PCA) and Multidimensional Scaling (MDS) [21, 22, 23] cannot find meaningful low dimensional structures hidden in high-dimensional observations when their intrinsic structures are non-linear or locally linear. Some non-linear dimensionality reduction methods, such as topology representing network [16], Isometric Feature Mapping (ISOMAP) [17, 18, 19], locally linear embedding (LLE) [20], can successfully find the intrinsic structure given that the data set is representative enough. This section will review some of these linear/non-linear dimensionality reduction techniques.

Multidimensional Scaling The classic Multidimensional Scaling (MDS) method tries to find a set of vectors in d-dimensional space such that the matrix of Euclidean distances among them corresponds as closely as possible to the distances between their corresponding vectors in the original measurement space (D-dimensional, where $D >> d$) by minimizing some cost function. Different MDS methods, such as [21, 22, 23], use different cost functions to find the low dimensional space. MDS is a global minimization

method; it tries to preserve the geometric distance. However, in some cases, when the intrinsic geometry of the graph is nonlinear or locally linear, MDS fails to reconstruct a graph in a low dimensional space.

Topology representing networks Martinetz and Schulten showed [16] how the simple competitive Hebbian rule (CHR) forms topology representing networks. Let us define $Q = \mathbf{q}_1, \cdots, \mathbf{q}_k$ as a set of points, called quantizers, on a manifold $M \subset R^D$. With each quantizer $\mathbf{q}_i$ a Voronoi set V_i is associated in the following manner: $V_i = x \in R^D : \|\mathbf{q}_i - \mathbf{x}\| = \min_j \|\mathbf{q}_j - \mathbf{x}\|$, where $\|\cdot\|$ denotes the vector norm. The Delaunay triangulation $\mathcal{D}_Q$ associated with Q is defined as the graph that connects quantizers with adjacent Voronoi sets (two Voronoi sets are called adjacent if their intersection is non-empty.). The masked Voronoi sets $V_i^{(M)}$ are defined as the intersection of the original Voronoi sets with the manifold M. The Delaunay triangulation $\mathcal{D}_Q^{(M)}$ on Q induced by the manifold M is the graph that connects quantizers if the intersection of their masked Voronoi sets is non-empty.

Given a set of quantizers Q and a finite data set $\mathbf{X}_n$, the CHR produces a set of edges as follows: (i) For every $\mathbf{x}_i \in X_n$ determine the closest and second closest quantizer, respectively $\mathbf{q}_{i_0}$ and $\mathbf{q}_{i_1}$. (ii) Include (i_0, i_1) as an edge in E. A set of quantizers Q on M is called dense if for each $\mathbf{x}$ on M the triangle formed by x and its closest and second closest quantizer lies completely on M. Obviously, if the distribution of the quantizer over the manifold is homogeneous (the volumes of the associated Voronoi regions are equal), the quantization can be made dense simply by increasing the number of quantizers.

Martinetz and Schulten showed that if Q is dense with respect to M, the CHR produces the induced Delaunay triangulation.

ISOMAP The ISOMAP algorithm [18] finds coordinates in R^d of data that lie on a d dimensional manifold embedded in a $D >> d$ dimensional space. The aim is to preserve the topological structure of the data, i.e. the Euclidean Distances in R^d should correspond to the geodesic distances (distances on the manifold). The

algorithm makes use of a neighborhood graph to find the topological structure of the data. The neighborhood graph can be obtained either by connecting all points that are within some small distance of each other (ϵ-method) or by connecting each point to its k nearest neighbors. The algorithm is then summarized as follows: (i) Construct neighborhood graph. (ii) Compute the graph distance (the graph distance is defined as the minimum distance among all paths in the graph that connect the two data points. The length of a path is the sum of the lengths its edges.) between all data points using a shortest path algorithm, for example Dijkstra's algorithm. (iii) Find low dimensional coordinates by applying MDS on the pairwise distances.

The run time of the ISOMAP algorithm is dominated by the computation of the neighborhood graph, costing $O(n^2)$, and computing the pairwise distances, which costs $O(n^2 \log n)$.

Locally Linear Embedding The idea underpinning the Locally Linear Embedding (LLE) algorithm [20] is the assumption that the manifold is locally linear. It follows that small patches cut out from the manifold in R^D should be approximately equal (up to a rotation, translation and scaling) to small patches on the manifold in R^d. Therefore, local relations among data in R^D that are invariant under rotation, translation and scaling should also be (approximately) valid in R^d. Using this principle, the procedure to find low dimensional coordinates for the data is simple: Express each data point $\mathbf{x}_i$ as a linear (possibly convex) combination of its k nearest neighbors $\mathbf{x}_{i_1}, \cdots, \mathbf{x}_{i_k} : \mathbf{x}_i = \sum_{j=1}^{k} \omega_{i_j} \mathbf{x}_{i_j} + \epsilon$, where ϵ is the approximation error whose norm is mininmized by the weights that are used. Then we find coordinates $\mathbf{y}_i \in R^d$ such that $\sum_{i=1}^{n} \left\| \mathbf{y}_i - \sum_{j=1}^{k} \omega_{i_j} \mathbf{y}_{i_j} \right\|^2$ is minimized. It turns out that the $\mathbf{y}_i$ can be obtained by finding d eigenvectors of a $n \times n$ matrix.

2.3 Head Pose Estimation

In recent years, a lot of research work has been done on head pose estimation [69, 70, 71, 72, 73, 74, 79, 80]. Generally, head pose estimation methods can be categorized into two classes, 1) feature-based approaches, 2) view-based approaches.

Feature-based techniques try to find facial feature points in an image from which it is possible to calculate the actual head orientation. These features can be obvious facial characteristics like eyes, nose, mouth etc. View-based techniques, on the other hand, try to analyze the entire head image in order to decide in which direction a person's head is oriented.

Generally, feature-based methods have the limitation that the same points must be visible over the entire image sequence, thus limiting the range of head motions they can track [59]. View-based methods do not suffer from this limitation. However, view-based methods normally require a large dataset of training sample.

Matsumoto and Zelinsky [60] proposed a template-matching technique for feature-based head pose estimation. They store six small image templates of eye and mouth corners. In each image frame they scan for the position where the templates fit best. Subsequently, the 3D position of these facial features are computed. By determining the rotation matrix M which maps these six points to a pre-defined head model, the head pose is obtained.

Harvile et al. [63] used the optical flow in an image sequence to determine the relative head movement from one frame to the next. They use the brightness change constraint equation (BCCE) to model the motion in the image. Moreover they added a depth change constraint equation to incorporate the stereo information. Morency et al. [64] improved this technique by storing a couple of key frames to reduce drift.

Srinivasan and Boyer [61] proposed a head pose estimation technique using view-based eigenspaces. Monrency et al. [62] extended this idea to 3D view-based eigenspaces,

where they use additional depth information. They use a Kalman filter to calculate the pose change from one frame to the next. However, they reduce drift by comparing the images to a number of key frames. These key frames are created automatically from a single view of the person.

Stiefelhagen et al. [65] estimated the head orientation with neural networks. They use normalized gray value images as input patterns. They scaled the images down to 20×30 pixels. To improve performance they added the image's horizontal and vertical edges to the input patterns. In [66], they further improved the performance by using the depth information.

Gee and Cipolla have presented an approach for determining the gaze direction using a geometrical model of the human face [67]. Their approach is based on the computation of the ratios between some facial features like nose, eyes, and mouth. They present a real-time gaze tracker which uses simple methods to extract the eye and mouth points from the gray-scale images. These points are then used to determine the facial normal. They do not report the accuracy of their system, but they show some example images with a little pointer for visualization of the head direction.

Ballard and Storkman [68] built a system for sensing the face direction. They showed two different approaches for detecting facial feature points. One approach relies on the eye and nose triangle, the other one uses a deformable template. The detected feature points are then used for the computation of the facial normal. The uncertainty in the feature extraction results in a major error of 22.5% in the yaw angle and 15% in the pitch angle. Their system is used in a human-machine interface to control a mouse pointer on a computer screen.

Wu and Toyama [75] proposed to use a probabilistic model approach to detect the head pose. They used four image-based features—convolution with a coarse scale Gaussian and convolution with rotation-invariant Gabor templates at four scales—to build the probabilistic model for each pose and determine the pose of an input image by computing the maximum *a posteriori* pose. Their algorithm uses an 3D ellipsoidal

model of the head to represent the pose information. Brown and Tian [76] used the same probabilistic model but instead of a 3D model they used 2D images directly to determine the coarse pose by computing the maximum *a posteriori* probability.

Rae and Ritter [77] used three neural networks to do color segmentation, face localization, and head orientation estimation respectively. The inputs of their neural network for head orientation estimation are a set of heuristically parameterized Gabor filters extracted from the head region (80×80). Their system is user-dependent, i.e., it works well for a person included in the training data but performance degrades for unseen persons. Zhao & Pingali [78] also presented a head orientation estimation system using neural networks. They used two neural networks to determine pan and tilt angles separately. Brown and Tian [76] use a three layer NN to estimate the head pose. They propose to histogram equalize the input image to reduce the effects of variable lighting conditions.

2.4 Periodic Motion Analysis

Recently, a lot of work has been done in segmenting and analyzing periodic motion. Existing methods can be categorized as those requiring point correspondences [13, 15]; those analyzing periodicities of pixels [8, 12]; those analyzing features of periodic motion [11, 6, 7]; and those analyzing the periodicities of object similarities [1, 4, 5, 13]. Related work has been done in analyzing the rigidity of moving objects [14, 9]. Below we review and critique each of these methods.

Cutler and Davis [1] compute the image self-similarity S of a sequence of motion images using absolute correlation. These motion images used are first Gaussian filtered and stabilized to segment the motion area. Then, morphological operation is performed to reduce motion due to image noise. They merge the large connected components of motion area and eliminate small ones. The motion sequences that demonstrate periodicity are walking or running persons from airborne video. A Fisher's test is

utilized to detect the periodic motions from nonperiodic ones. Fisher's test rejects the null hypothesis if the self-similarity shows only white noise by testing whether the power spectrum $P(f_i)$ is substantially larger than the average value. If the periodicity is non-stationary, the normal Fourier Analysis will not be appropriate to find the correct periodicity. Instead, they propose to use a Short-Time Fourier Transform (STFT). They use a short-time analysis window (Hanning windowing function) in the Fourier Transform to find the "local" spectrum of the signal. Their method is useful when motions like walking and running demonstrate strong peroidicity or at least "local" periodicity, i.e. periodic in several periods. However, their method will fail significantly when the motion is nonperiodic but cyclic.

Seitz and Dyer [13] compute a temporal correlation plot for repeating motions using different image comparison functions, d_{A} and d_{I}. The affine comparison function d_{A} allows for view-invariant analysis of image motion, but requires point correspondences (which are achieved by tracking reflectors on the analyzed objects). The image comparison function d_{I} computes the sum of absolute differences between images. However, the objects are not tracked and, thus, must have nontranslational periodic motion in order for periodic motion to be detected. Cyclic motion is analyzed by computing the period-trace, which are curves that are fit to the surface d. Snakes are used to fit these curves, which assumes that d is well-behaved near zero so that near-matching configurations show up as local minima of d. The K-S test is utilized to classify periodic and nonperiodic motion. The samples used in the K-S test are the correlation matrix M and the hypothesized period-trace PT. The null hypothesis is that the motion is not periodic, i.e., the cumulative distribution function M and PT are not significantly different. The K-S test rejects the null hypothesis when periodic motion is present. However, it also rejects the null hypothesis if M is nonstationary. For example, when M has a trend, the cumulative distribution function of M and PT can be significantly different, resulting in classifying the motion as periodic (even if no periodic motion present). This can occur if the viewpoint of the object or lighting changes significantly during evaluation of M. The basic weakness of this method is it uses a one-sided

hypothesis test which assumes stationarity and works for periodic motion only.

Polana and Nelson [12] recognize periodic motions in an image sequence by first aligning the frames with respect to the centroid of an object. Reference curves, which are lines parallel to the trajectory of the motion flow centroid, are then extracted and the spectral power is estimated for the image signals along these curves. The periodicity measure of each reference curve is defined as the normalized difference between the sum of the spectral energy at the highest amplitude frequency and its multiples and the sum of the energy at the frequencies half way between.

Tsai et al. [15] analyze the periodic motion of a person walking parallel to the image plane. Both synthetic and real walking sequences were analyzed. For the real images, point correspondences were achieved by manually tracking the joints of the body. Periodicity was detected using Fourier analysis of the smoothed spatio-temporal curvature function of the trajectories created by specific points on the body as it performs periodic motion. A motion-based recognition application is described in which one complete cycle is stored as a model and a matching process is performed using one cycle of an input trajectory.

Allmen [2] used spatio-temporal flow curves of edge image sequences (with no background edges present) to analyze cyclic motion. Repeating patterns in the ST flow curves are detected using curvature scale-space. A potential problem with this technique is that the curvature of the ST flow curves is sensitive to noise. Such a technique would likely fail on very noisy sequences.

Niyogi and Adelson [11] analyze human gait by first segmenting a person walking parallel to the image plane using background subtraction. A spatio-temporal surface is fit to the XYT pattern created by the walking person. This surface is approximately periodic and reflects the periodicity of the gait. Related work [10] used this surface (extracted differently) for gait recognition.

Liu and Picard [8] assume a static camera and use background subtraction to segment motion. Foreground objects are tracked and their path is fit to a line using a Hough

transform (all examples have motion parallel to the image plane). The power spectrum of the temporal histories of each pixel is then analyzed using Fourier analysis and the harmonic energy caused by periodic motion is estimated. An implicit assumption in [8] is that the background is homogeneous (a sufficiently nonhomogeneous background will swamp the harmonic energy). Our work differs from [8] and [12] in that we analyze the periodicities of the image similarities of large areas of an object, not just individual pixels aligned with an object. Because of this difference (and the fact that we use a smooth image similarity metric), our Fourier analysis is much simpler since the signals we analyze do not have significant harmonics of the fundamental frequency. The harmonics in [8] and [12] are due to the large discontinuities in the signal of a single pixel; our self-similarity metric does not have such discontinuities.

Fujiyoshi and Lipton [6] segment moving objects from a static camera and extract the object boundaries. From the object boundary, a "star" skeleton is produced, which is then Fourier analyzed for periodic motion. This method requires accurate motion segmentation, which is not always possible. Also, objects must be segmented individually; no partial occlusions are allowed. In addition, since only the boundary of the object is analyzed for periodic change (and not the interior of the object), some periodic motions may not be detected (e.g., a textured rolling ball, or a person walking directly toward the camera).

Selinger and Wixson [14] track objects and compute self-similarities of that object. A simple heuristic using the peaks of the 1D similarity measure is used to classify rigid and nonrigid moving objects, which in our tests fails to classify correctly for noisy images.

Heisele and Wohler [7] recognize pedestrians using color images from a moving camera. The images are segmented using a color/position feature space and the resulting clusters are tracked. A quadratic polynomial classifier extracts those clusters which represent the legs of pedestrians. The clusters are then classified by a time delay neural network, with spatio-temporal receptive fields. This method requires accurate

object segmentation. A 3-CCD color camera was used to facilitate the color clustering and pedestrians are approximately 100 pixels in height. These image qualities and resolutions are typically not found in surveillance applications.

There has also been some work done in classifying periodic motion. Polana and Nelson [12] use the dominant frequency of the detected periodicity to determine the temporal scale of the motion. A temporally scaled XYT template, where XY is a feature based on optical flow, is used to match the given motion. The periodic motions include walking, running, swinging, jumping, skiing, jumping jacks, and a toy frog. This technique is view dependent and has not been demonstrated to generalize across different subjects and viewing conditions. Also, since optical flow is used, it will be highly susceptible to image noise.

Cohen et al. [3] classify oscillatory gestures of a moving light by modeling the gestures as simple one-dimensional ordinary differential equations. Six classes of gestures are considered (all circular and linear paths). This technique requires point correspondences and has not been shown to work on arbitrary oscillatory motions.

Area-based techniques, such as our method, have several advantages over pixel-based techniques, such as [12, 8]. Specifically, area-based techniques allow the analysis of the dynamics of the entire object, which is not achievable by pixel-based techniques. This allows for classification of different types of periodic motion. In addition, area-based techniques allow detection and analysis of periodic motion that is not parallel to the image plane. All examples given in [12, 8] have motion parallel to the image plane, which ensures there is sufficient periodic pixel variation for the techniques to work. However, since area-based methods compute object similarities which span many pixels, the individual pixel variations do not have to be large. A related benefit is that area-based techniques allow the analysis of low S/N images, since the S/N of the object similarity measure is higher than that of a single pixel.

Chapter 3

Head Pose Estimation

In this chapter, we will describe our method of head pose estimation (HPE). The algorithm for HPE method is composed of two parts: i) unified embedding to find the 2-D feature space; ii) parameter learning to find a person-independent mapping. This is then used in an entropy-based classifier to detect FCFA behavior. Here, we propose to use foreground segmentation and edge detection to extract the head in each frame of the sequence for further experiments. However, our algorithm can be used with head sequences extracted by other different head tracking algorithms (see a review in [84]). Head tracking is a step before FCFA detection. It is related while not within the scope of our discussion.

All the data we used in the HPE method are image sequences obtained from a fixed video camera. To simplify the problem, we obtain the video such that the heads only rotate horizontally without any upward or downward rotation, i.e., a pan rotation only. A sample sequence is shown in Fig. 3.1. Since the size of the head in each image of a sequence and between different sequences could be different, we normalize them to a fixed size of $n_1 \times n_2$.

Figure 3.1: A sample sequence used in our HPE method.

3.1 Unified Embedding

3.1.1 Nonlinear Dimensionality Reduction

Since the image sequences primarily exhibit head pose changes, we believe that even though the images are in high dimensional space, they must lie on some manifold with dimensionality much lower than the original. Recently, several new non-linear dimensionality reduction techniques have been proposed, such as Isometric Feature Mapping (ISOMAP) [18] and locally linear embedding (LLE) [20]. Both methods have been shown to successfully embed manifolds in high dimensional space onto a low dimensional space in several examples. In our work, we adapt the ISOMAP framework. Table 3.1 details the three steps in the ISOMAP algorithm. The algorithm takes as input the distances $d_x(i,j)$ between all pairs i,j from N data points in the high-dimensional input space X, measured either in the standard Euclidean metric or in some domain-specific metric. The algorithm outputs coordinate vectors $\mathbf{y}_i$ in a d-dimensional Euclidean space Y that best represents the intrinsic geometry of the data. The only free parameter (ϵ or K) appears in Step 1.

Fig. 3.2(a) shows the 2-D embedding of the sequence sampled in Fig. 3.1 using the K-ISOMAP ($K = 7$ in our experiments) algorithm. Since we rotate the head so that there is almost no tilt angle change, i.e., it is a pan rotation (1-D circular motion physically) only, we believe a good choice of the embedding space is a 2-D plane. If

Table 3.1: A complete description of the ISOMAP algorithm.

Step	Operation	Description
1	Construct neighborhood graph	Define the graph G over all N data points by connecting points i and j if they are [as measured by $d_x(i,j)$] closer than ϵ (ϵ-ISOMAP),or if i is one of the K nearest neighbors of j (K-ISOMAP). Set edge lengths equal to $d_x(i,j)$.
2	Compute shortest paths	Initialize $d_G(i,j) = d_x(i,j)$ if i,j are linked by an edge; $d_G(i,j) = \infty$ otherwise. Then for each value of $k = 1, 2, \cdots, N$ in turn, replace all entries $d_G(i,j)$ by $\min\{d_G(i,j), d_G(i,k) + d_G(k,j)\}$. The matrix of final values $D_G(i,j)$ will contain the shortest path distances between all pairs of points in G.
3	Construct d-dimensional embedding	Let λ_p be the p-th eigenvalue (in decreasing order) of the matrix $\tau(D_G)$ (The matrix τ is defined by $\tau(D) = -HSH/2$, where S is the matrix of squared distances $\{S_{ij} = {D_{ij}}^2\}$, and H is the centering matrix $\{H_{ij} = \delta_{ij} - 1/N\}$.), and v_p^i be the i-th component of the p-th eigen vector. Then set the p-th component of the d-dimensional coordinate vector $\mathbf{y}_i$ equal to $\sqrt{\lambda_p} v_p^i$.

1-D space is chosen here, it will cause a discontinuity at head pose angles of 0° and 360°. However, by choosing a 2-D plane, this problem can be solved, which as can be seen later is very important for the non-linear person-independent mapping. As can be noticed from Fig. 3.2(a), the embedding can discriminate different pan angles. The outline of the embedding can be seen to be ellipse-like. The frames with head pan angles close to each other in the images are also close in the embedded space. One point that needs to be emphasized is that we do not use the temporal relationships to achieve the embedding, since the goal is to obtain an embedding that preserves the geometry

of the manifold. Temporal relation can be used to determine the neighborhood of each frame but it was found to lead to erroneous, artificial embedding.

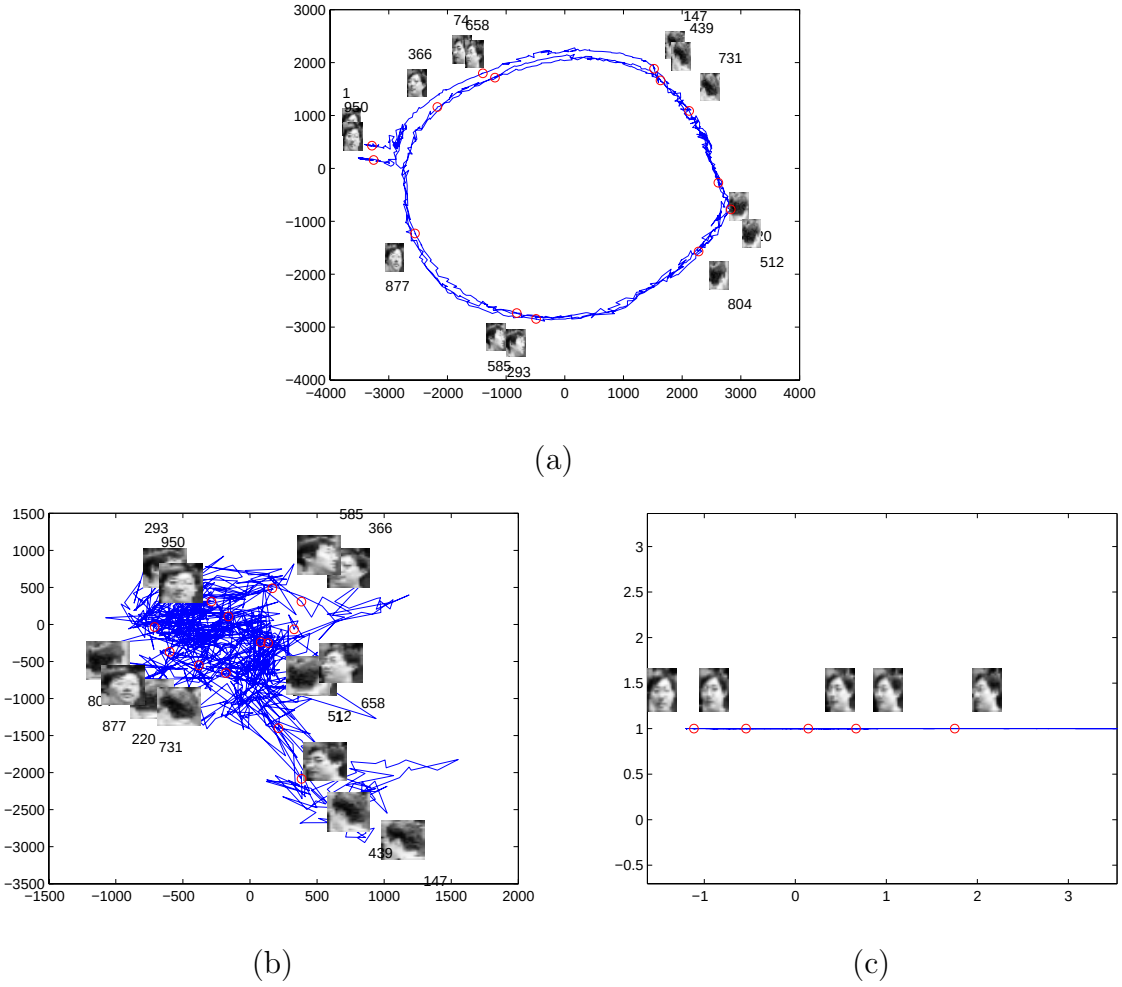


Figure 3.2: 2-D embeding of the sequence sampled in Fig. 3.1 (a) by ISOMAP, (b) by PCA, (c) by LLE.

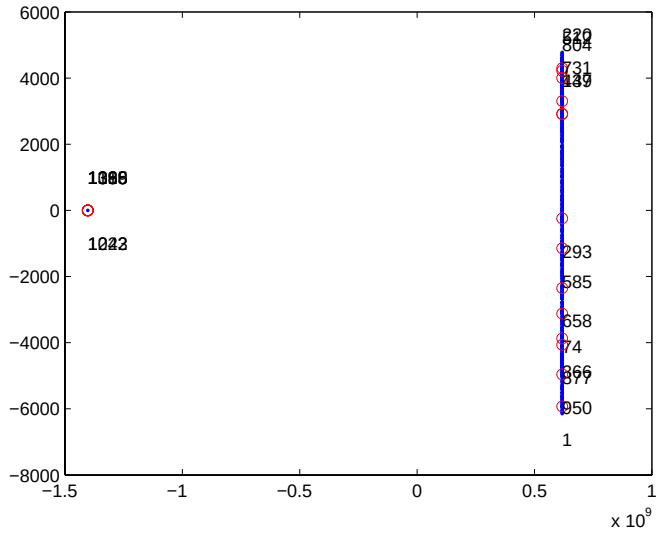
Fig. 3.2(b) and (c) show corresponding results using the classic linear dimensionality reduction method of principal component analysis (PCA) and the non-linear dimensionality reduction method of LLE on the same sequence. We choose also a 2-D embedding to make them comparable. As can be seen, PCA leads to an embedding that cannot differentiate head poses in our case. Using LLE makes the 1-D circular motion degenerate into a line in a 2-D plane, which correctly shows the intrinsic dimensionality of this motion. However, the points at the leftmost and the rightmost end of the line

correspond to similar poses, which, however, are far away in the embedded space. This characteristic is not suitable for our non-linear person-independent mapping method, and will cause large error as shown later.

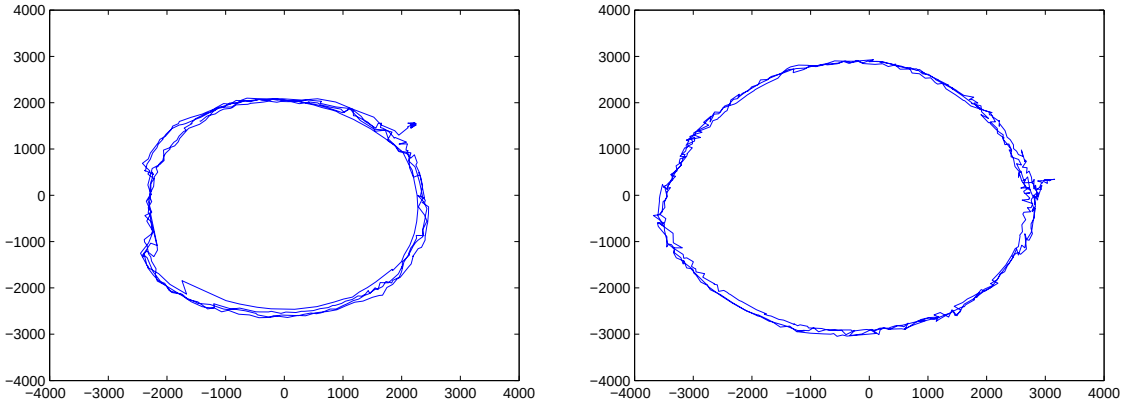
3.1.2 Embedding Multiple Manifolds

Although the ISOMAP can very effectively represent a hidden manifold in high dimensional space into a low dimensional embedded space as shown in Fig. 3.2(a), it fails to embed multiple people's data together into one manifold. Since typically intra-person differences are much smaller than inter-person differences, the residual variance minimization technique used in ISOMAP, therefore, tries to preserve large contributions from inter-person variations. This is shown in Fig. 3.3(a) where ISOMAP is used to embed two people's manifolds (care has been taken to ensure that all the inputs are spatially registered). Here, the embedding shows separate manifolds (note one manifold has degenerated into a point because the embedding is dominated by inter-person distances which are much larger than intra-person distances.) Besides, another fundamental problem is that different persons will have different shape of manifold. This can be seen in Fig. 3.3(b).

To embed multiple persons' data to find a useful, common 2-D feature space, each person's manifold is first embedded separately using ISOMAP. An interesting point here is that, although the appearance (shape) of the manifold for each person differs, they are all ellipse-like (different parameters for different manifolds). We then find a best fitting ellipse [85] to represent each manifold before we further normalize it. Fig. 3.4 shows the results of the ellipse fitted on the manifold of the sequence sampled in Fig. 3.1. The parameters of each ellipse were then used to scale the coordinate axes of each embedded space to obtain a unit circle. After we normalize the coordinates in every person's embedded space into a unit circle, we find an interesting property that on every person's unit circle the angles between any two points are roughly the same as the difference between their corresponding pose angles in the original images.



(a)



(b)

Figure 3.3: (a) Embedding obtained by ISOMAP on the combination of two person's sequences. (b) Separate embedding of two manifolds for two people's head pan images.

However, when using ISOMAP to embed each person's manifold individually, it cannot be ensured that different person's frontal faces are close in angle in each embedded space. Thus, further normalization is needed to make all person's frontal images to be located at the same angle in the manifold so that they are comparable and meaningful to build a unified embedded space. To do this, we first manually label the frames in each sequence with frontal views of the head. To reduce the labelling error, we label all the frames with a frontal or near frontal view, take the mean of the corresponding

coordinates in the embedded space, and rotate it so that the frontal images are located at the 90 degree angle. In this way, we align all the person's frontal view coordinates to the same angle.

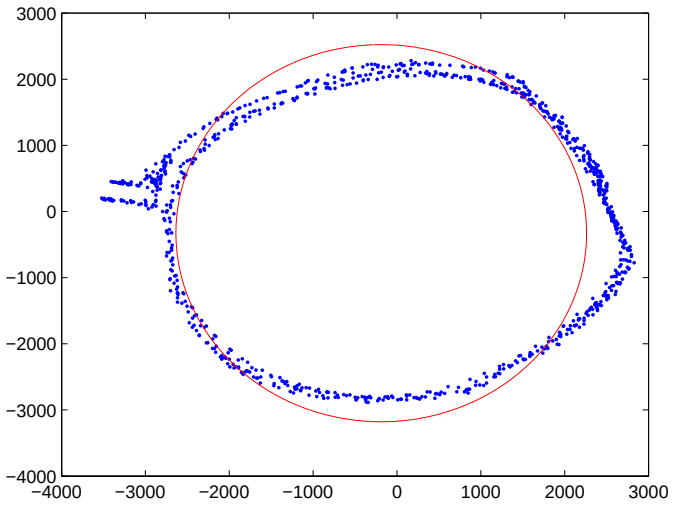


Figure 3.4: The results of the ellipse (solid line) fitted on the sequence (dotted points).

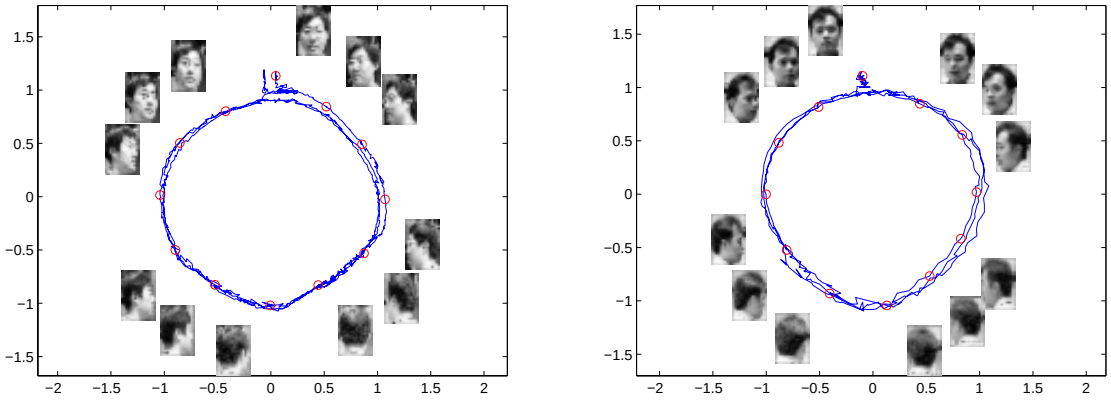


Figure 3.5: Two sequences whose low-dimensional embedded manifolds have been normalized into the unified embedding space (shown separately).

After we rotate every person's normalized unit circle so that the frontal view frames are at the 90 degree angle, the left profile frames are automatically located at about

either 0° or 180°. Since the embedding can turn out to be either clockwise or anti-clockwise, we form a mirror image along the Y-axis for those unit circles where the left profile faces are at around 180 degrees, i.e., anticlockwise embeddings. Finally, we have a unified embedded space where different persons' similar head pose images are close to each other on the unit circle, and we call this unified embedding space the feature space. Fig. 3.5 shows two of the sequences normalized to obtain a unified embedding space. The details of obtaining the unified embedded space are given in Table 3.2.

Table 3.2: A complete description of our unified embedding algorithm.

Step	Operation	Description
1	Individual Embedding	Define $Y^P = \{\mathbf{y}_1^P, \cdots, \mathbf{y}_{n_P}^P\}$ the vector sequence of length n_P in the original measurement space for person P. ISOMAP is used to embed Y^P to a 2-D embedded space. $Z^P = \{\mathbf{z}_1^P, \cdots, \mathbf{z}_{n_P}^P\}$ are the corresponding coordinates in the 2-D embedded space for person P.
2	Ellipse Fitting	For person P, we use an ellipse to fit Z^P, resulting in the ellipse with parameters: center $\mathbf{c}_e^P = (c_x^P, c_y^P)^T$, major and minor axes a^P and b^P respectively, and orientation Φ_e^P.
3	Multiple Embedding	For person P, let $\mathbf{z}_i^P = (z_{i1}^P, z_{i2}^P)^T$, $i = 1, \cdots, n_P$. We rotate and reshape every $\mathbf{z}_i^P$ to obtain $\mathbf{z}_i^{*^P} = \begin{pmatrix} 1/a^P & 0 \\ 0 & 1/b^P \end{pmatrix} \left(\begin{pmatrix} cos\Phi_e^P & -sin\Phi_e^P \\ sin\Phi_e^P & cos\Phi_e^P \end{pmatrix} \mathbf{z}_i^P - \mathbf{c}_e^P \right)$. Identify the frontal face frames for Peron P, and the corresponding $\{\mathbf{z}_i^{*P}\}$ of these frames. The mean of these points is calculated, and the embedded space is rotated so that this mean value lies at the 90 degrees angle. After that, we choose a frame l showing left profile and test whether $\mathbf{z}_l^{*^P}$ is close to 0 degrees. If not, we set $\mathbf{z}_i^{*^P} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \mathbf{z}_i^{*^P}$.

3.2 Person-Independent Mapping

3.2.1 RBF Interpolation

As described in Table 3.2, let the input images of person P from a sequence are $Y^P = \{\mathbf{y}_1^P, \cdots, \mathbf{y}_{n_P}^P \in R^D\}$ and the sets of corresponding points in the feature space, i.e. the unified embedded space, are $Z^{*^P} = \{\mathbf{z}_1^{*^P}, \cdots, \mathbf{z}_{n_P}^{*^P}\}$, where n_P is the number of frames for person P. We can then learn a nonlinear interpolative mapping from the input images to the corresponding coordinates in the feature space by using Radial Basis Functions.

We combine all the persons' sequences together, $\Gamma = \{Y^{P_1}, \cdots, Y^{P_k}\} = \{\mathbf{y}_1, \cdots, \mathbf{y}_{n_0}\}$, and their corresponding coordinates in the feature space, $\Lambda = \{Z^{*^{P_1}}, \cdots, Z^{*^{P_k}}\} = \{\mathbf{z}_1^*, \cdots, \mathbf{z}_{n_0}^*\}$, where $n_0 = n_{P_1} + \cdots + n_{P_k}$ is the total number of input images. For every single point in the feature space, we take the interpolative mapping function in the form of

$$f(\mathbf{y}) = \omega_0 + \sum_{i=1}^{M} \omega_i \cdot \psi(|\mathbf{y} - \mathbf{c}_i|). \tag{3.1}$$

where $\psi(\cdot)$ is a real-valued basis function, ω_i are real coefficients, $\mathbf{c}_i$, $i = 1, \cdots, M$ are centers of the basis functions on R^D, $|\cdot|$ is the norm on R^D (original input space). Choices for basis functions include thin-plate spline ($\psi(u) = u^2 log(u)$), the multi-quadric ($\psi(u) = \sqrt{u^2 + a^2}$), Gaussian ($\psi(u) = e^{-\frac{u^2}{2\sigma^2}}$), etc..

In our experiment, we use Gaussian basis functions and employ k-means clustering [82] algorithm to find the corresponding centers. Once basis centers have been determined, the widths σ_i^2 are set equal to the variances of the points in the corresponding cluster.

To decide the number of basis functions to use, we experimentally tested various values of M and calculated the mean squared error of the RBF output. For every

value of M, we used a leave-one-out cross-validation method, i.e., we take out in turn one person's data for testing, and combine all the remaining persons' data to learn the parameters of the RBF interpolation system. Fig. 3.6 shows the results of our test for different number of basis functions (from 2 to 50). As can be seen in Fig. 3.6, to avoid both underfitting and overfitting, a good choice of the number of basis functions is $M = 8$.

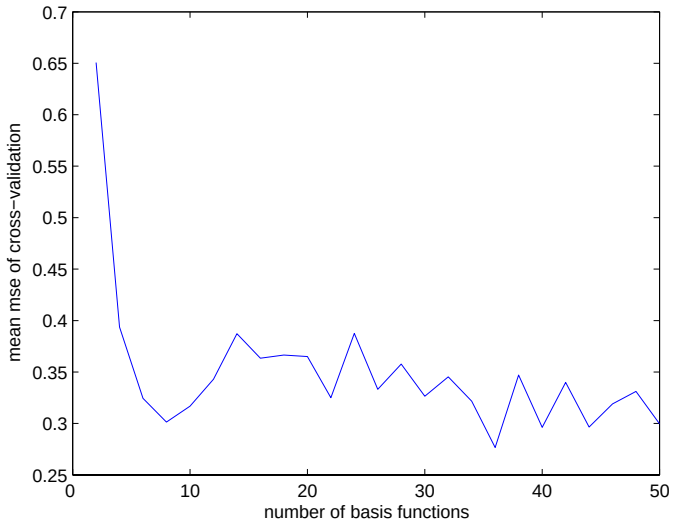


Figure 3.6: Mean squared error on different values of M.

Let $\psi_i = \psi(|\mathbf{y} - \mathbf{c}_i|)$ and by introducing an extra basis function $\psi_0 = 1$, (3.1) can be written as

$$f(\mathbf{y}) = \sum_{i=0}^{M} \omega_i \psi_i. \tag{3.2}$$

Let points in the feature space be written as $\mathbf{z}_i^* = (z_{i1}^*, z_{i2}^*)$. After obtaining the centers $\mathbf{c}_1, \cdots, \mathbf{c}_M$, and determining the width σ_i^2, to determine the weights ω_i, we merely have to solve a set of simple linear equations

$$f_l(\mathbf{y}_i) = \sum_{j=0}^{M} \omega_{lj} \cdot \psi(|\mathbf{y}_i - \mathbf{c}_j|) = z_{il}^*, \quad i = 1, \cdots, n_0, \tag{3.3}$$

where $l = 1, 2$.

By defining matrices $\Omega = \begin{pmatrix} \omega_{10} & \cdots & \omega_{1M} \\ \omega_{20} & \cdots & \omega_{2M} \end{pmatrix}$, $\Psi = \begin{pmatrix} \psi_{11} & \cdots & \psi_{n_0 1} \\ \vdots & \psi_{ij} & \vdots \\ \psi_{1M} & \cdots & \psi_{n_0 M} \end{pmatrix}$, $Z = \begin{pmatrix} z_{11}^* & \cdots & z_{n_0 1}^* \\ z_{12}^* & \cdots & z_{n_0 2}^* \end{pmatrix}$, where $\psi_{ij} = \psi(|\mathbf{y}_i - \mathbf{c}_j|)$, (3.3) can be written in matrix form as

$$\Omega \cdot \Psi = Z. \tag{3.4}$$

The least square solution for Ω is then given by

$$\Omega = Z\Psi^{\Delta}, \tag{3.5}$$

where $\Psi^{\Delta} = \Psi^T(\Psi\Psi^T)^{-1}$ is the pseudo inverse of Ψ.

3.2.2 Adaptive Local Fitting

The RBF interpolation can map an image or a video sequence into the 2-D feature space and find the corresponding coordinate or sequence of coordinates. Specially, when processing video sequences, such as in the case of attentive behavior detection, temporal continuity requirement and temporal local linearity assumption can be applied to correct unreasonable mappings, if any, in individual frames, and to smooth the outputs of RBF interpolation. We propose an adaptive local fitting (ALF) technique. Our ALF algorithm is composed of two parts: 1) adaptive outlier correction; 2) locally linear fitting.

In adaptive outlier correction, assuming temporal continuity of the head video sequence and their corresponding 2-D features, estimates which are far away from those of their S (an even number and let $S = 2s_0$) temporally nearest neighbor (S-TNN) frames are defined as outliers. Let $\mathbf{z}_t$ be the output of the RBF interpolation system for the t-th frame, and D_t^S be the mean distance between $\mathbf{z}_t$ and the points

$\{\mathbf{z}_{t-k}| - s_0 \leq k \leq s_0, k \neq 0\}$:

$$D_t^S = \frac{1}{S} \sum_{k=-s_0, k\neq 0}^{s_0} |\mathbf{z}_t - \mathbf{z}_{t-k}|\,, \tag{3.6}$$

where $|\cdot|$ is the norm on the 2-D feature space.

For the t-th frame, we wait until the $(t + s_0)$-th image (to obtain all S-TNNs) to make update. We adaptively calculate D_t^S and update the mean M_t and the variance V_t of the sequence $\{D_{s_0+1}^S, \cdots, D_t^S\}$ as follows

$$\begin{aligned} M_t &= \frac{1}{t - s_0}[(t - s_0 - 1)M_{t-1} + D_t^S], \\ V_t &= \frac{1}{t - s_0 - 1}(\sum_{j=s_0+1}^{t} D_j^{S^2} - (t - s_0){M_t}^2). \end{aligned}$$

To check for outliers, we set a threshold $h = \lambda\sqrt{V_t}$, where λ is a tolerance coefficient. Using different values of λ can make the system tolerant to different degrees of sudden change in the head pose. If $D_t - M_t > h$, we deem point $\mathbf{z}_t$ an outlier, and set $\mathbf{z}_t = \frac{1}{S}\sum_{j=t-s_0, j\neq t}^{t+s_0} \mathbf{z}_j$.

In locally linear fitting, we assume the local linearity within a temporal window of the length of L. We employed the technique suggested in [86] for linear fitting to smooth the output of RBF interpolation.

After the above process, the head pose angle can be very easily estimated as

$$\theta_t = \tan^{-1}(\frac{z_{t2}}{z_{t1}}). \tag{3.7}$$

3.3 Entropy Classifier

Here we propose a simple method to detect FCFA behavior in a video sequence, given the head pose angle estimated for each frame as discussed above. The head pose angle range of 0°-360° is divided into Q equally spaced angular regions. Given a video sequence of length N, a pose angle histogram with Q bins is calculated as

$$p_i = \frac{n_i}{N}, \quad i = 1, 2, \cdots, Q \tag{3.8}$$

where n_i is the number of pose angles which fall into the i-th bin. The head pose entropy E of the sequence is then estimated as

$$E = -\sum_{i=1}^{Q} p_i \log p_i. \tag{3.9}$$

For focused attention, we expect that the entropy will be low, and become high for FCFA behavior. Hence we set a threshold on E to detect FCFA.

A block diagram of our HPE algorithm as discussed above is shown in Fig. 3.7.

As shown in Fig. 3.7, in the offline learning process, we first use ISOMAP to find the individual 2-D embedding for each person in the training data, then a coordinate normalizer is proposed to find a unified embedding (2-D feature space) for multiple persons. Following this, we use the original images and the corresponding coordinates in the 2-D feature space to train and learn the parameters of the RBF interpolator.

In the online head pose estimation scheme, we use the trained RBF interpolator to map new head images or sequence of head images into the 2-D feature space. For video sequence of head images, we propose an adaptive local fitting technique to correct unreasonable mapping and smooth the output. The head pose angle is then obtained as a simple trigonometric function of the 2-D coordinates. To extend our HPE method to detect FCFA behavior, we designed an entropy-based classifier. Giving the sequence

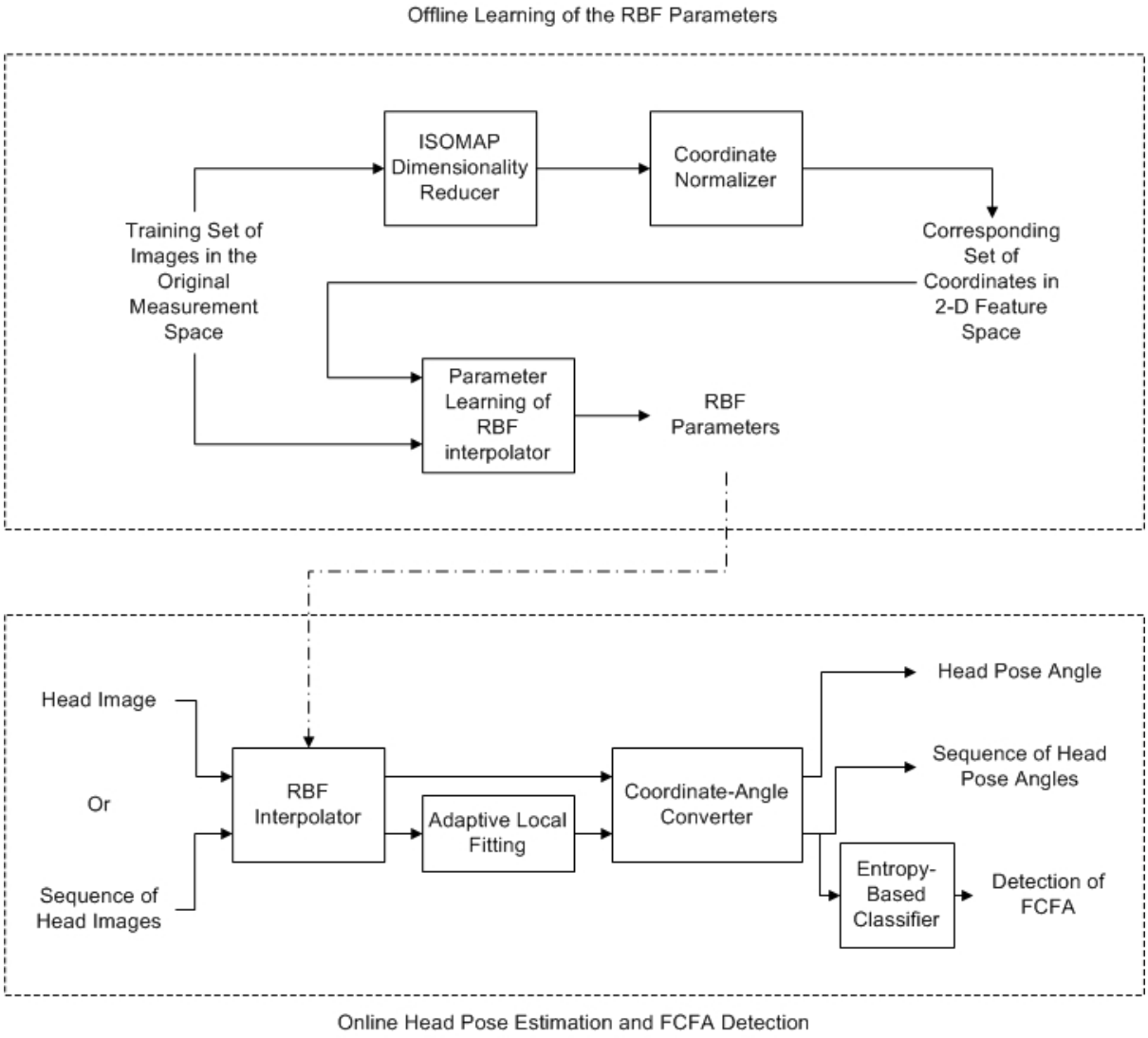


Figure 3.7: Overview of our HPE algorithm.

of head pose angles, we calculate the head pose angle entropy of the sequence and compare it with a preset threshold to detect FCFA behavior.

Chapter 4

Cyclic Pattern Frequency Analysis

In this chapter, we present another technique for cyclic pattern frequency analysis (CPFA) to differentiate between two types of attentive behaviors, i.e., focused attention and frequent change in focus of attention (FCFA) based on detecting non-cyclic or cyclic head motion, respectively. The algorithm for cyclic motion detection consists of three parts: (1) linear dimensionality reduction of head images; (2) head pose similarity computation as it evolves in time; (3) frequency analysis and classification. To extact the head from images, we use the same technique discussed in Chapter 3. However, head tracking is by itself a research area with several prior works[83, 69]. Hence, our algorithm can also be used with head sequences extracted from other different head tracking algorithms (see a review in [84]).

In the following sections, video sequences of a person looking around (called "watcher"), i.e., exhibiting FCFA behavior as shown in Fig. 4.1(a), and a person talking to others (called "talker"), i.e., exhibiting focused attention as shown in Fig. 4.1(b), will be used to illustrate the algorithms and methods used.

(a) watcher (b) talker

Figure 4.1: A sample of extracted heads of a watcher (FCFA behavior) and a talker (focused attention).

4.1 Similarity Matrix

The input data here is a sequence of images given head centers $\mathbf{c}_i$ located. Before we calculate the similarity, we first normalize the head in each frame of the sequence to be a fixed size of $n_1 \times n_2$. To characterize the cyclicity of the head, we first compute the head H's similarity in images t_1 and t_2. While many image similarity metrics can be used, we used the absolute difference [1, 13], as it is computationally simple:

$$S_{t_1,t_2} = \sum_{(x,y)\in B} |O_{t_1}(x,y) - O_{t_2}(x,y)|, \tag{4.1}$$

where $O_t(x,y)$ is the image intensity at the pixel (x,y) of the t-th image, B is the bounding box $n_1 \times n_2$ of head H centered at the head center $\mathbf{c}_i$. In order to reduce sensitivity to head location errors, the minimal S is found by computing similarities over a small square search window, to obtain the best similarity match S'_{t_1,t_2} as below:

$$S'_{t_1,t_2} = \min_{|\mathrm{d}x|,|\mathrm{d}y|<a} \sum_{(x,y)\in B} |O_{t_1}(x+\mathrm{d}x, y+\mathrm{d}y) - O_{t_2}(x,y)|. \tag{4.2}$$

In our experiments we used $a = 2$ for all sequences, as the results were insensitive to $a \geq 2$. Using S'_{t_1,t_2}, we define a similarity matrix for an N-image sequence as

$$R = \left[S'_{t_i,t_j}\right]_{N\times N}, i,j = 1,2,\cdots,N. \tag{4.3}$$

Fig. 4.2 shows an example of the similarity matrix R for watcher and talker, displayed

as images. The values of the matrix elements have been linearly scaled to the gray-scale intensity range [0,255]. Dark regions show more similarity. Note that the matrix is symmetric along the main diagonal. As can be seen from Fig. 4.2, the appearance of the similarity matrix R for watcher and talker are different. R for watcher has more interlacing of black and white regions indicating that the similarities between different images within the sequence vary significantly, i.e., the person is looking around and exhibiting FCFA behavior. On the contrary, R for talker looks more smooth which means that the similarities between images within the sequence are higher (S' is smaller). This happens when the head pose does not change much in the whole sequence indicating a focused attention behavior.

(a) watcher (b) talker

Figure 4.2: Similarity matrix R of a (a) watcher (exhibiting FCFA) and (b) talker (exhibiting focused attention).

4.2 Dimensionality Reduction and Fast Algorithm

Similarity matrix R calculated as in (4.1) and (4.2) using original images does show the difference between FCFA and focused attention behavior as can be seen in Fig. 4.2, however, it is time consuming to compute because of the high dimensionality of head images (the dimensionality of the head images is $n_1 n_2 = n_1 \times n_2$). A direct and easy solution to save computational time is to use principal component analysis (PCA) to reduce the dimensionality of the images. Here, we did not use the ISOMAP algorithm to reduce the dimensionality as was used in Chapter 3 because the video sequences we used in CPFA method is taken with whatever upward or downward motion of the head

which violated the assumption used in Chapter 3.

For any two n-dimensional vectors, $\mathbf{x} = (x_1, \cdots, x_n)^{\mathrm{T}}$ and $\mathbf{y} = (y_1, \cdots, y_n)^{\mathrm{T}}$, let $D_{\mathrm{E}}(\mathbf{x}, \mathbf{y})$ be the Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$ and $D_{\mathrm{Abs}}(\mathbf{x}, \mathbf{y})$ be the absolute distance between $\mathbf{x}$ and $\mathbf{y}$. A standard result in linear algebra shows that $D_{\mathrm{Abs}}(\mathbf{x}, \mathbf{y})$ is bounded as

$$D_{\mathrm{E}}(\mathbf{x}, \mathbf{y}) \leq D_{\mathrm{Abs}}(\mathbf{x}, \mathbf{y}) \leq \sqrt{n} D_{\mathrm{E}}(\mathbf{x}, \mathbf{y}). \tag{4.4}$$

Let the vectors $\mathbf{x}$ and $\mathbf{y}$ be transformed by PCA to the d-dimensional vectors $\mathbf{x}'$ and $\mathbf{y}'$, respectively. If the PCA dimensionality reduction preserves almost all of the energy ($D_E(\mathbf{x}, \mathbf{y}) \approx D_E(\mathbf{x}', \mathbf{y}')$), the difference between the absolute distance in the original space $D_{\mathrm{Abs_{Org}}} = D_{\mathrm{Abs}}(\mathbf{x}, \mathbf{y})$ and that in PCA subspace $D_{\mathrm{Abs_{PCA}}} = D_{\mathrm{Abs}}(\mathbf{x}', \mathbf{y}')$ is bounded by

$$(1 - \sqrt{d}) D_{\mathrm{E}} \leq (D_{\mathrm{Abs_{Org}}} - D_{\mathrm{Abs_{PCA}}}) \leq (\sqrt{n} - 1) D_{\mathrm{E}}. \tag{4.5}$$

The bound in (4.5) shows that when $\mathbf{x}$ is near (or similar) to $\mathbf{y}$, i.e. D_{E} is small, the difference between $D_{\mathrm{Abs_{Org}}}$ and $D_{\mathrm{Abs_{PCA}}}$ is narrowly bounded and from (4.4), because D_{E} is small, both $D_{\mathrm{Abs_{Org}}}$ and $D_{\mathrm{Abs_{PCA}}}$ are small too. When $\mathbf{x}$ is far away from (or dissimilar to) $\mathbf{y}$, i.e. D_{E} is large, from (4.4) both $D_{\mathrm{Abs_{Org}}}$ and $D_{\mathrm{Abs_{PCA}}}$ are large. Hence, $D_{\mathrm{Abs_{PCA}}}$ exhibit the same properties as $D_{\mathrm{Abs_{Org}}}$, and can be used to measure similarity.

We choose a d-dimensional PCA subspace for image representation. We have found that even for small representational error $d << n_1 n_2$ where the images are of dimension $n_1 \times n_2$. The projection matrix $\mathbf{P}$ from original image space to PCA subspace hence of is of dimension $d \times n_1 n_2$.

To account for the head center locating error, for the t-th head image, we shifted the head center by ± 1 pixel vertically or horizontally or both, which resulted in 9 possible

head images, written as vectors $\mathbf{H}_{t1}, \cdots, \mathbf{H}_{t9}$, each of which is $n_1 n_2$-dimensional. It is easy to see that this process is equivalent to the shifting used in 4.2, since when calculating the similarity between two images, here we shift both images by ± 1 pixel to search for the minimal similarity, while in 4.2 we shift one image by ± 2 pixel and keep the other fixed. Projecting each $\mathbf{H}_{ti}$ onto the predefined PCA subspace, we get the 9 vectors

$$\mathbf{h}_{ti} = \mathbf{P}\mathbf{H}_{ti}, \quad i = 1, \cdots, 9 \tag{4.6}$$

The similarity between image t_1 and image t_2 is then obtained by choosing the minimal pairwise absolute distances in the PCA subspace between the shifted head vectors for these two images, and is given as

$$S''_{t_1,t_2} = \min_{i,j} D_{\text{Abs}}(\mathbf{h}_{t_1 i}, \mathbf{h}_{t_2 j}), \;\; i, j = 1, \cdots, 9 \;\; . \tag{4.7}$$

The computation for similarity by searching for the minimal absolute distance will cost $O(d)$ using (4.7) in the PCA subspace instead of $O(n_1 n_2)$ using (4.2) in the original measurement space. This translates to significant savings for computing the similarity matrix R.

The efficient algorithm for computing the similarity matrix for the image sequence is described below:

1. Preprocessing and PCA Training

    - Given the training image sequences of length N, detect the location of the head in each image;
    - Nomalize the size of the head in each image to $n_1 \times n_2$, and set the bounding box to the fixed size $n_1 \times n_2$ and centered at the head center in each image;
    - Use the normalized head images of different persons' to find the PCA projection matrix $\mathbf{P}$.

2. Computing the Similarity Matrix

- Extract 9 shifted head subimage vectors $\{\mathbf{H}_{ti},\ i = 1, \cdots, 9\}$ from each image of the sequence, and compute their corresponding vectors $\{\mathbf{h}_{ti},\ i = 1, \cdots, 9\}$ in the PCA subspace according to (4.6).
- For the t-th ($t = 2, \cdots, N$) frame, calculate the absolute distance S'' as in (4.7) between itself and the previous $t-1$ images

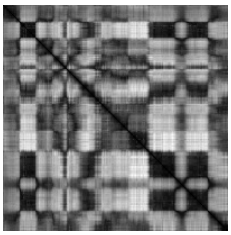
$$S''_{i,t},\ \ i = 1, \cdots, t-1; \tag{4.8}$$

- Form the similarity matrix R' by setting $S''_{t,i} = S''_{i,t}$, for $i > t$, and $S''_{j,j} = 0$, for $j = 1, \cdots, N$,

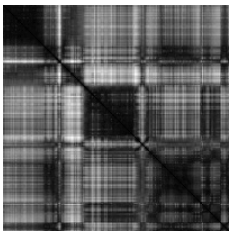
$$R' = \left[S''_{i,j}\right], i, j = 1, \cdots, N. \tag{4.9}$$

Fig. 4.3 shows images of the similarity matrix R' for watcher and talker calculated in the PCA subspace using the above algorithm. The values of the matrix elements have been linearly scaled to the gray-scale intensity range [0,255]. Note that similarity matrices R''s are similar to R's shown in Fig. 4.2 in texture except that they are darker than R's. The reason that R' is similar to R is that calculating R' in the PCA subspace preserves the similarities and dissimilarities between images as discussed above. The difference in average brightness can be attributed to the fact that the actual values in the 2 similarity matrices R and R' can be different leading to different scaling parameters for the $[0, 255]$ display range.

To calculate R' during online operation when the images are coming continuously, we need only to design a stack of length N based on the first-in-first-out (FIFO) rule. When a new image is obtained, we push it into the stack and remove the oldest one to form a new N-image sequence. To obtain R' for the new sequence, the only calculation is the S'' between the new image itself and its previous $N-1$ images.



(a) watcher



(b) talker

Figure 4.3: Plot of similarity matrix R' for watcher and talker.

4.3 Frequency Analysis

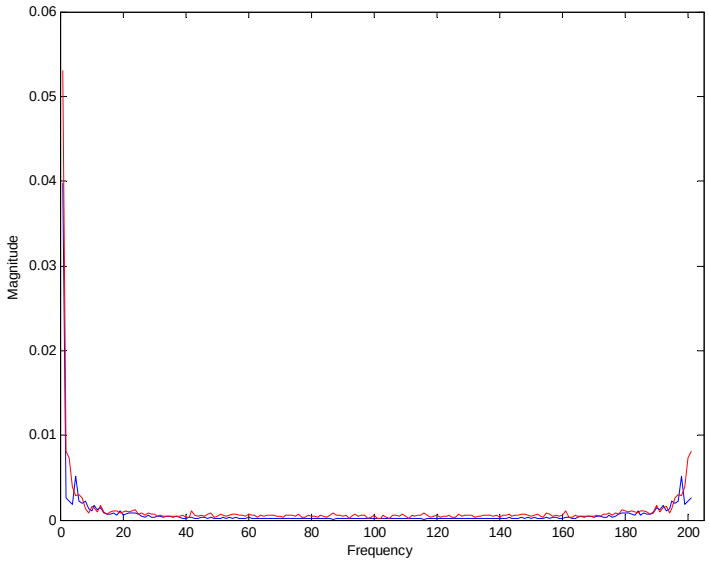
For analyzing cyclic motion, many methods could be used. We choose Fourier analysis for its simplicity and ease of use.

To find the characteristics of the behavior, one direct way is to apply 1-D Fourier Transform to all the rows of the similarity matrix R, and average the Fourier spectra of all the rows. Figure 4.4(a) shows the averaged Fourier spectra of watcher and talker, which appear to be similar. However, if we zoom into the low frequency area, as shown in Figure 4.4(b), we can see that the spectral values for talker are larger than those for watcher. This gives us a hint to find features in the low frequency area for classification. Since R' is a 2-D matrix, we use 2-D Discrete Fourier Transform [81] to find the Fourier spectrum matrix $F_{R'}$ of the similarity matrix.

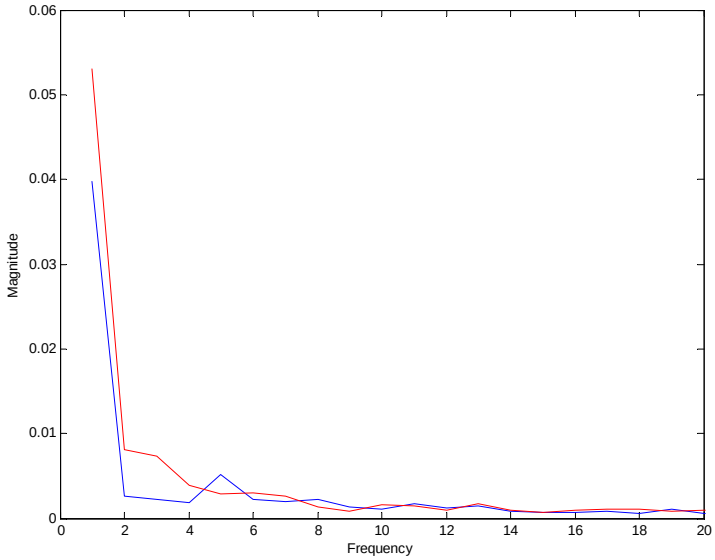
To make the value of the elements in Fourier spectrum matrix comparable for different persons, we normalized them by the total energy of the similarity matrix to obtain

$$F_{R'} = \frac{\mathcal{F}\{R'\}}{\sum_{i=1}^{N}\sum_{j=1}^{N}|R'(i,j)|^2}, \tag{4.10}$$

where N is the number of images in the sequence, and $\mathcal{F}\{\cdot\}$ denotes the 2-D Fourier Transform operator. Analogous to (4.10), for purposes of comparison we can also compute a matrix F_R based on the similarity matrix R computed as in (4.3) using the original head images.



(a)

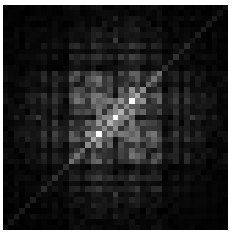


(b)

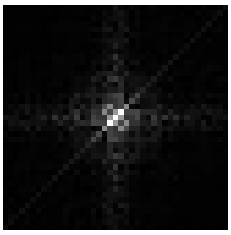
Figure 4.4: (a) Averaged 1-D Fourier spectrum of watcher (Blue) and talker (Red); (b)Zoom-in of (a) in the low frequency area.

Central areas of F_R and $F_{R'}$ matrices for watcher and talker are shown in Figs. 4.5 and 4.6. The values of the elements have been linearly scaled to [0,255]; as the DC

component here is much larger than that of any other frequency, we set it the value of the second largest element for display purposes; bright areas show high Fourier spectral values. Note that the symmetry property of the similarity matrices R and R', and the Fourier Transform makes F_R and $F_{R'}$ matrices symmetric diagonally and cross diagonally. From comparison of Fig. 4.5 and 4.6 it is apparent that the two spectra using R and R' are very similar. Hence, we use R' as it is computationally simpler to calculate.

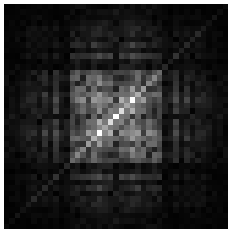


(a) watcher

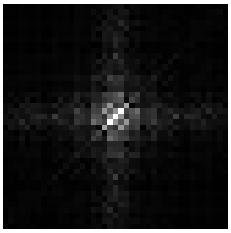


(b) talker

Figure 4.5: Central area of F_R matrix for (a) watcher and (b) talker.



(a) watcher



(b) talker

Figure 4.6: Central area of $F_{R'}$ matrix for (a) watch and (b) talker.

4.4 Feature Selection

Given the Fourier spectrum matrix $F_{R'}$ we choose as features those elements of $F_{R'}$ that show significant differences between the two classes. Thus, given an element e_j of the matrix $F_{R'}$, we define a coefficient δ_j to reflect the degree of difference between the two classes as:

$$\delta_j = \frac{|\text{mean}(e_j|\omega_1) - \text{mean}(e_j|\omega_2)|}{\text{std}(e_j|\omega_1) + \text{std}(e_j|\omega_2)} \tag{4.11}$$

where $\text{mean}(e_j|\omega_i)$, $\text{std}(e_j|\omega_i)$ are the mean and standard deviation of e_j given class ω_i, where $i = 1, 2$.

We calculated the δ_j values of 16 low frequency elements in $F_{R'}$, and the results are

shown in Fig. 4.7. The 4 elements which have significantly large values of δ_j are chosen to compose the feature vector. These 4 elements correspond to the Fourier spectrum at the frequencies $(0, 0)$, $(0, \frac{2\pi}{N})$ and $(\frac{2\pi}{N}, \pm\frac{2\pi}{N})$.

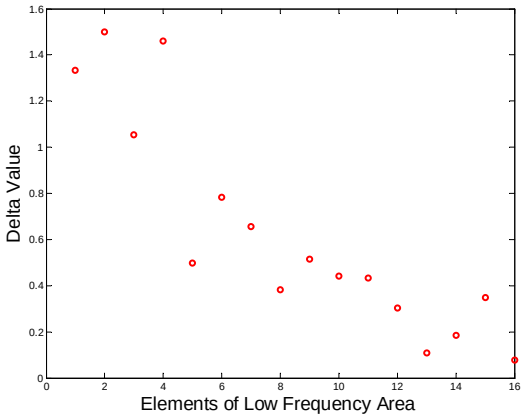


Figure 4.7: The δ_j values (Delta Value) of the 16 elements in the low frequency area.

4.5 K-NNR Classifier

As the distribution of the feature vector is unknown, we employ a nonparametric approach — *k-nearest-neighbor* (K-NNR) *rule* [82] for classification. We assign Class ω_1 for FCFA and Class ω_2 for focused attention and use $k = 3$ (odd to avoid ties). A Leave-One-Out Cross-validation (LOOCV) method is adopted to estimate the overall performance.

Figure 4.8 shows a block diagram of our algorithm. After we detect and normalize the head in each image of the sequence, we shift the bounding box to extract head subimage vectors for each image. By projecting on a pre-trained PCA transform matrix $\mathbf{P}$, corresponding vectors in the PCA subspace is obtained, where we use absolute distance to calculate the similarity between images and form the similarity matrix R'. Through frequency analysis on R', we get a normalized Fourier spectrum matrix $F_{R'}$. A feature vector is then formed by selecting the 4 element in the low frequency area of

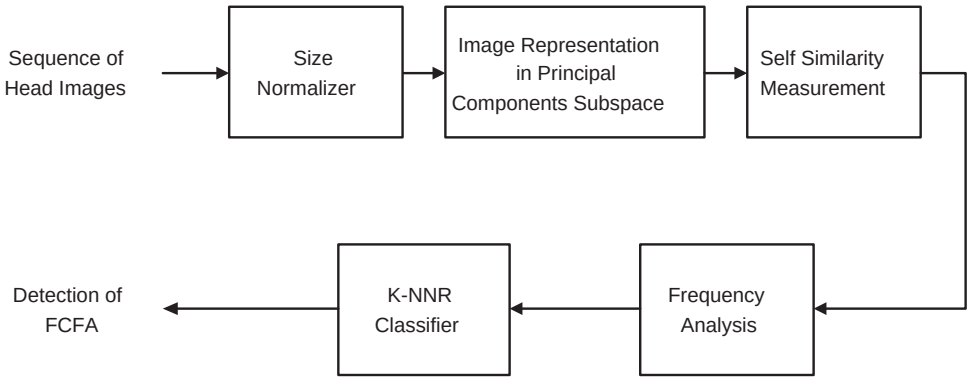


Figure 4.8: Overview of our CPFA algorithm.

$F_{R'}$. Using a K-NNR classifier, we detect FCFA behavior by the classification of FCFA from focused attention behavior.

Chapter 5

Experiments and Discussion

In this chapter, we give the experimental results of our HPE method and our CPFA method, and discuss their performance.

5.1 HPE Method

In this section, we present the results of our HPE method. In the first experiment, we use video sequences, where the persons are slowly rotating their heads for three complete revolutions continuously. We didn't set any limit for the rate of head rotaion. However, our expectation is to cover as many poses as possible, since we believe it will increase the accuracy of our person-independent mapping system, which can be deemed as a non-linear interpolating algorithm. To test the generalization ability of our person-independent mapping function to determine pose angle, we use a leave-one-out cross-validation (LOOCV) method. To test our algorithm to detect FCFA behavior, we performed a second experiment using new video data exhibiting simulated FCFA and focused attention. These results are also shown in this section.

5.1.1 Data Description and Preprocessing

The data we used is composed of two parts, 1) those used to learn the person-independent mapping; 2) data exhibiting FCFA and focused attention behavior for classification and testing performance of system. All image sequence data was obtained from a fixed video camera. To simplify the problem, we set the camera to be approximately level with the heads. During video sequence acquisition the persons were sitting on a chair which could be rotated. They kept their head level without any upward or downward tilt, as they were rotated in front of the camera during video acquisition.

Since the size of the head in each image throughout the sequence and between different sequences could be different, we normalized the head to a fixed size, $n_1 \times n_2 = 24 \times 16$. After head-size normalization, histogram equalization and Gaussian smoothing was applied to each image in the sequence to reduce the effects of varying illumination and noise.

For parameter learning, we used 7 persons' sequences (subsampled sequences shown in Fig. 5.1). The corresponding length of each sequence is shown in Table 5.1.

Table 5.1: Length of the 7 sequences used for parameter learning in HPE scheme.

Person	1	2	3	4	5	6	7	Total
Sequence Length	508	967	426	677	447	505	685	4215

For use in classification and detection of FCFA behavior, we obtained 4 more sequences, where two exhibited FCFA and two exhibited focused attention (subsampled sequences shown in Fig. 5.2). The corresponding length of the sequences are given in Table 5.2.

(1) (2)

(3) (4)

(5) (6)

(7)

Figure 5.1: Samples of the normalized, histogram equalized and Gaussian filtered head sequences of the 7 people used in learning.

5.1.2 Pose Estimation

We first individually embed every person's data and normalize them to find a unified embedding space as described in Chapter 3. Fig. 5.3 shows the unified embedding in

(a) (b)

(c) (d)

Figure 5.2: Samples of the normalized, histogram equalized and Gaussian filtered head sequences used in classification and detection of FCFA. ((a) and (b) exhibiting FCFA, (c) and (d) exhibiting focused attention).

Table 5.2: Length of the sequences used in classification and detection of FCFA.

Person	a	b	c	d
Sequence Length	2231	3074	1494	1322

the feature space for the persons in our experiment.

We use leave-one-out cross-validation (LOOCV) to test our person-independent mapping method, i.e., we take out in turn one sequence as the testing data and use all the remaining sequences for parameter learning. Fig. 5.4 shows the results of the person-independent mapping to estimate the head pose angle in each frame for each of the 7 sequences which are used in turn as the test data in the LOOCV method. The green lines correspond to "ground truth" head pose angle. This is obtained by calculating the projection of the test sequence into the unified 2-D embedded space. This gound truth can be compared to the pose angles estimated from the person-independent RBF

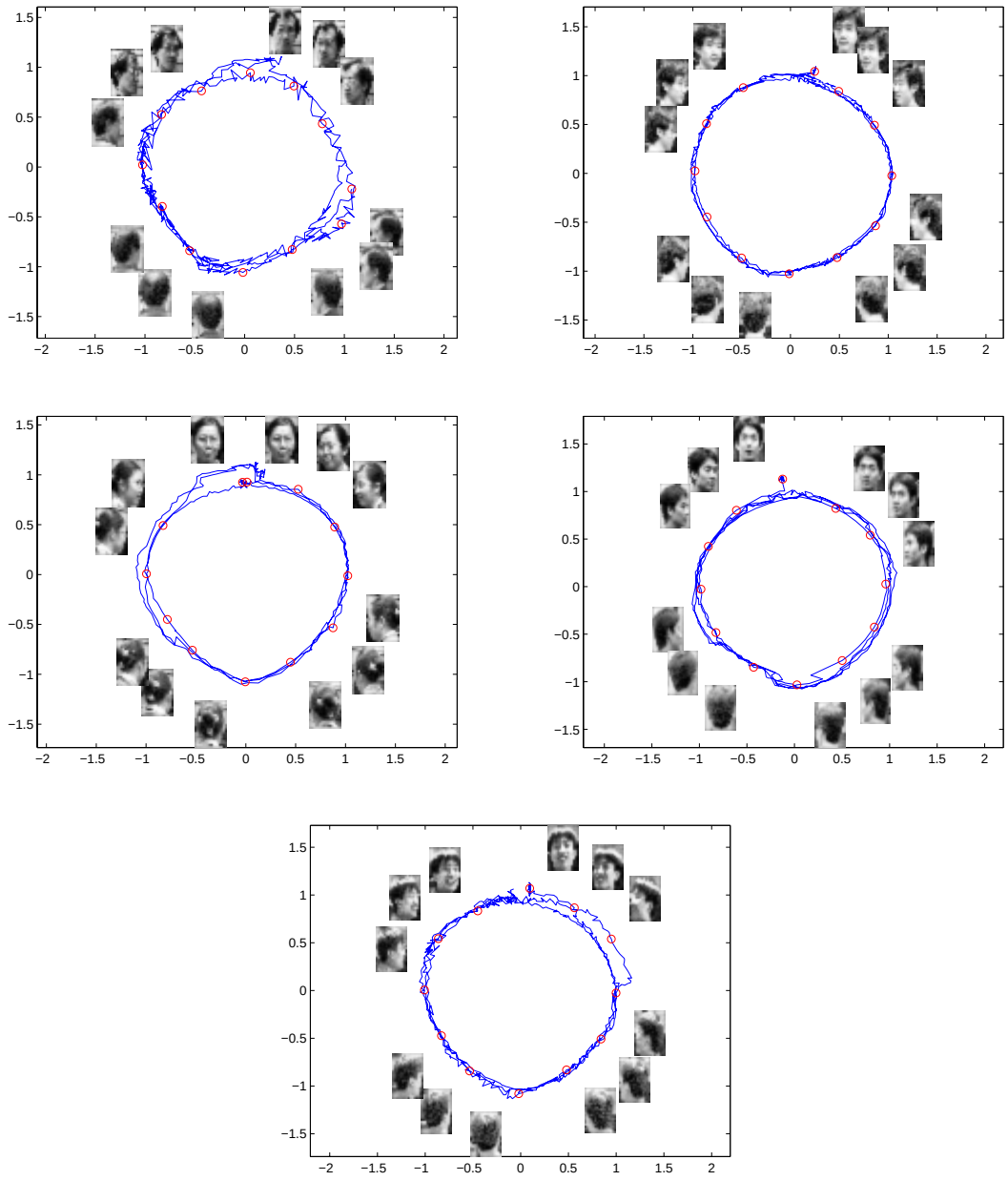


Figure 5.3: Feature space showing the unified embedding for 5 of the 7 persons (please see Fig. 3.5 for the other two).

interpolation system shown with red lines, and it can be seen that the latter are very good approximations to the ground truth. The values above the small head images are the pose angles of those images calculated from person-independent mapping.

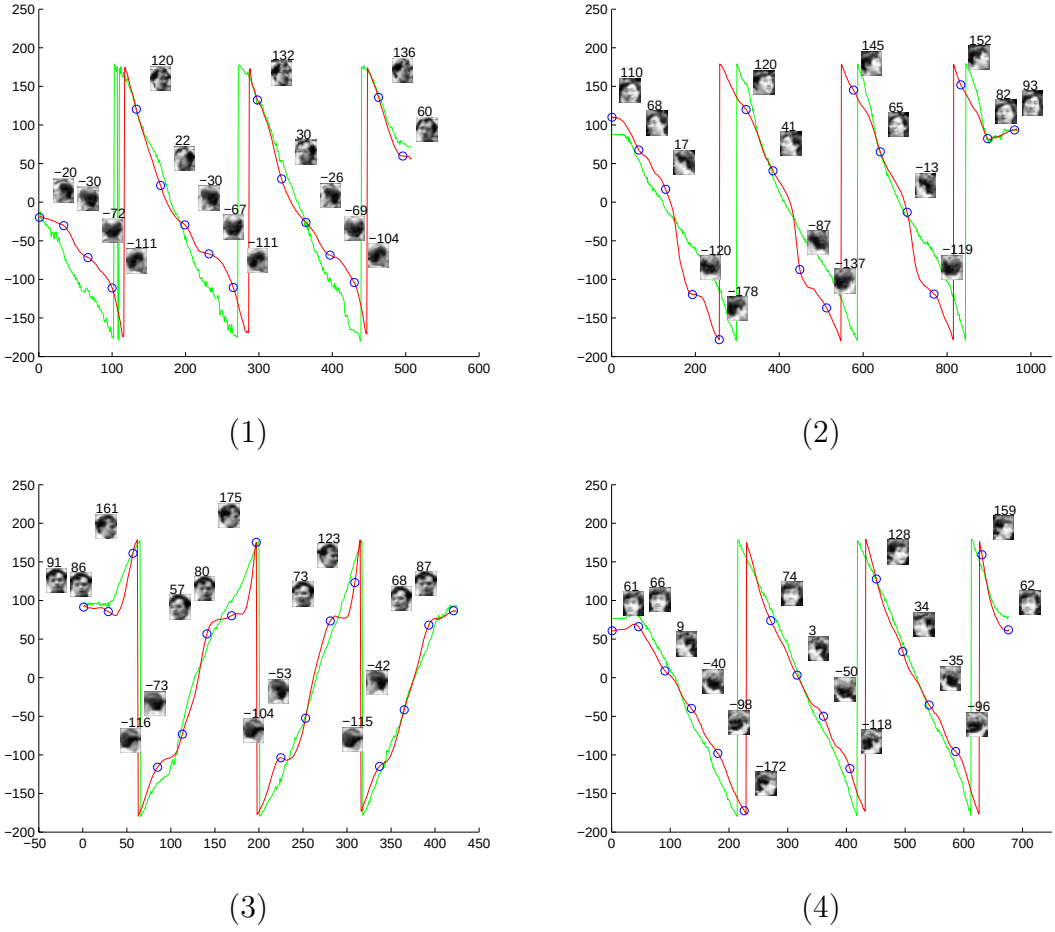
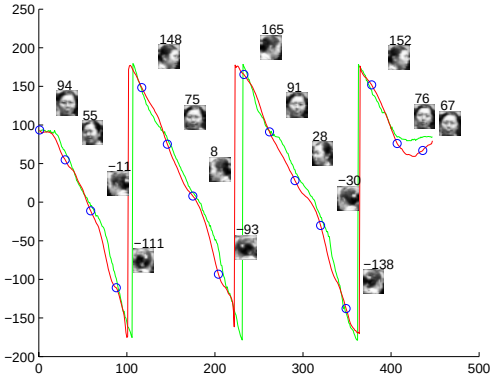


Figure 5.4: The LOOCV results of our person-independent mapping system to estimate head pose angle. Green lines correspond to "ground truth" pose angles, while red lines show the pose angles estimated by the person-independent mapping.

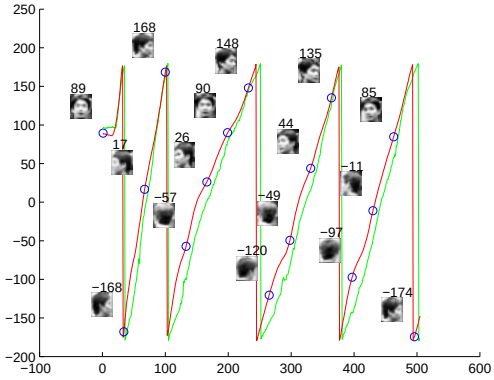
We found that our person-independent mapping system works well even if the face displays small facial expressions. This is the case for person (7) in Fig. 5.4(7), whose head image sequence is shown in Fig. 5.1(7), and the person appears to be smiling.

5.1.3 Validation on real FCFA data

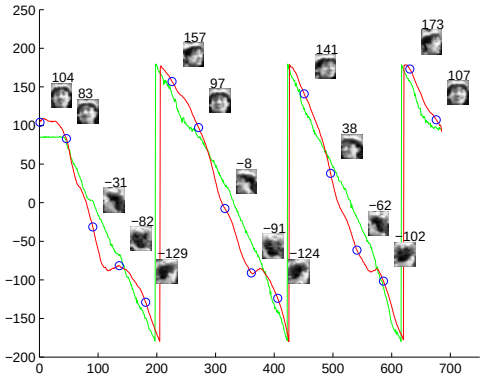
After testing the framework for person-independent head pose angle mapping system, we test its use for detecting FCFA behavior. For this purpose we acquire new data sequences, as sampled and shown in Fig. 5.2. These sequences are taken with the same



(5)



(6)



(7)

Figure 5.4 (continued): The LOOCV results of our person-independent mapping system to estimate head pose angle. Green lines correspond to "ground truth" pose angles, while red lines show the pose angles estimated by the person-independent mapping.

camera, but in a different environment than those used in Section 5.1.2. The sequences acquired here represent FCFA behavior (Fig. 5.2(a) and (b), where the persons are looking around) and focused attention behavior (Fig. 5.2(c) and (d), where the persons are roughly looking in two directions).

We process the whole sequence with the person-independent mapping system to estimate pose angle in each frame and then calculate the head pose entropy value E for each sequence as described in Section 3.3. To visualize the appearance of pose angles in sequences of FCFA and focused attention, we combine the estimated pose angle by the person-independent mapping system with the temporal information to

draw the trajectories as shown in Fig. 5.5 for FCFA sequences ((a) and (b)) and focused attention sequences ((c) and (d)). Here roughly circular trajectories in (a) and (b) depict the FCFA behavior of persons looking around quite well while for focused attention person are looking roughly in two directions, as can be seen in the trajectories of (c) and (d).

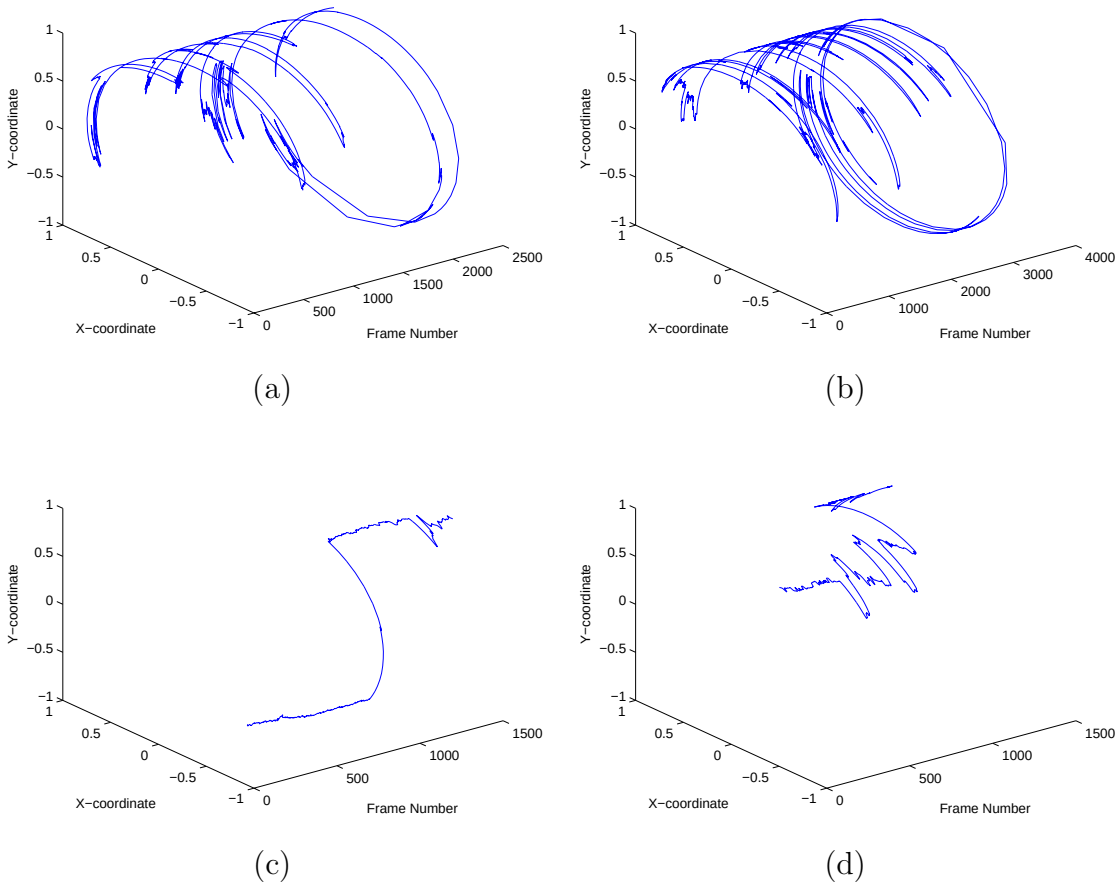


Figure 5.5: The trajectories of FCFA ((a) and (b)) and focused attention ((c) and (d)) behavior.

Table 5.3 shows the corresponding value of E for the sequences in Fig. 5.5 calculated using $Q = 36$ angular bins. It can be seen that the entropy values of FCFA behavior ((a) and (b)) are very distinct from those of focused attention ((c) and (d)). By setting a threshold of $E_0 = 2.5$, we can detect FCFA behavior perfectly in the 4 sequences.

Table 5.3: The entropy value of head pose corresponding to the sequences in Fig. 5.5.

	(a)	(b)	(c)	(d)
E	3.07	3.00	1.17	1.91

5.2 CPFA Method

In this section we present the results of our CPFA algorithm on FCFA and focused attention sequences which are different from those used in Section 5.1. In the first experiment, we use 11 sequences captured from a fixed camera, and use cross-validation to estimate the classification error of the CPFA method for detecting the two types of behaviors. To have a good estimate of the performance, we conducted a second experiment with 20 more sequences captured from different cameras and settings to validate the classifier built using all the data in the first experiment.

5.3 Data Description and Preprocessing

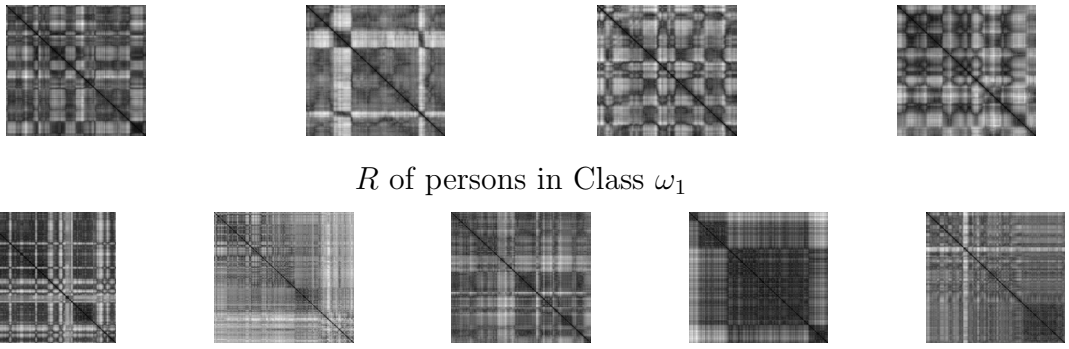
Video sequences used in the experiments are taken by a camera from the overhead corner of a hall with frame rate of 25 frames per second. These sequences are first cropped to a length of 40 seconds and then resampled by keeping one out of every 5 frames. Thus, for each person, we get an image sequence of 200 frames.

Since the fixed camera is far away from the object, the head scale within a sequence will not change. However for different sequences, the head sizes may be different. Thus, we first normalize every image of the sequence used into the size of $n_1 \times n_2 = 30 \times 20$. The original head size in the sequences ranges from 25×15 to 63×43.

To find the dimensionality d of the PCA subspace, we trained the data to preserve 98% of the total energy and resulted in a $d = 9$ dimensional space.

5.3.1 Classification and Validation

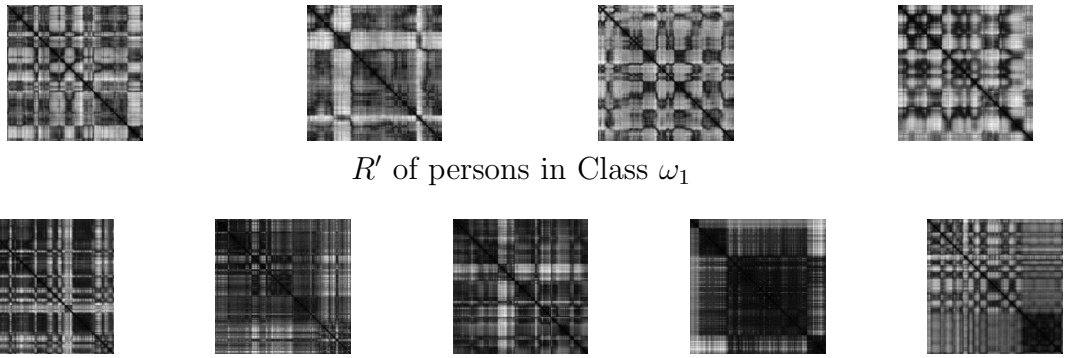
As described in Chapter 4, we assign Class ω_1 for FCFA and Class ω_2 for focused attention. The labeled training data here include five ω_1 sequences and six ω_2 sequences of different persons. The similarity matrix R and R' for each person are shown in Figure 5.6 and Figure 5.7.



R of persons in Class ω_1

R of persons in Class ω_2

Figure 5.6: Similarity matrix R (the original images are omitted here and the R's for watcher and talker are shown in Fig. 4.2).



R' of persons in Class ω_1

R' of persons in Class ω_2

Figure 5.7: Similarity matrix R' (the original images are omitted here and the R''s for watcher and talker are shown in Fig. 4.3).

The results of LOOCV using R showed that none of the ω_1 data in 5 cases was misclassified while one of the ω_2 data in 6 cases was misclassified. When examining

the cause of this misclassification (the similarity matrix is the leftmost of Class ω_2 in Figures 5.6 & 5.7), we found that the person was listening to others at first and then kept changing his attention to other directions (as shown in Figure 5.8). Thus, his data is, to some extent, similar to and overlaps with FCFA. As shown in Table 5.4, however, none is misclassified by R'.

Figure 5.8: Sampled images of misclassified data in the first experiment using R.

5.3.2 More Data Validation

To test whether the proposed method generalized well on other data sets, some more video sequences are validated on the classifier which is built with all of the data used in Section 5.3.1. The new sequences include 10 ω_1 sequences and 10 ω_2 sequences with different persons, different head sizes and different camera exposures taken by different cameras.

Using R, the results showed that 2 ω_1 sequences were misclassified and none of the ω_2 sequences is misclassified. Examining the misclassified data, we found that the two ω_1 data are taken under the same illumination and the same exposure which are the lowest among the whole data set. Their faces are dark and almost of the same color as that of hair. Thus, it is reasonable to expect that they would be misclassified.

Using R', however, only one sample was misclassified, yielding an improvement in classification accuracy. One possible reason for the better performance is that mapping the sequences into a subspace reduced the illumination effect while maintain the relative change between frames.

Table 5.4 summarizes the results of both experiments.

Table 5.4: Summary of experimental results of our CPFA method.

		using R		using R'	
		ω_1	ω_2	ω_1	ω_2
First Experiment	Class ω_1	4	1	5	0
	Class ω_2	0	6	0	6
	Accuracy	90.9%		100%	
Second Experiment	Class ω_1	10	0	10	0
	Class ω_2	2	8	1	9
	Accuracy	90%		95%	
Average Accuracy		90.3%		96.8%	

5.3.3 Computational Time

In the system, frames used for computation are 0.2s apart. The algorithm is implemented using sequences of $N = 200$ frames obtained by temporal subsampling of 40s of video on a 2.4GHz Pentium IV PC. The most time-consuming step is the calculation of the similarity matrix. Compared to this, the time used for FFT and K-NNR is trivial—63ms and 15ms respectively in Matlab. As Table 5.5 shows, running the algorithm to calculate the similarity matrix R' in Matlab needs 73.4s, which is about 2.5 times faster than calculating R which needs 186.3s. In a real-time system, upon the arrival of each image, we only need to compute the similarity between itself and the previous 199 subsampled images. The computation time is 0.75s in Matlab. It would be using less time if programmed in the C environment.

Table 5.5: Time used to calculate R & R' in Matlab.

R	R'
186.3s	73.4s

5.4 Discussion

Our HPE method works on images acquired from an uncalibrated single camera and can successfully estimate the head pose angle even when the person is totally or partially turned back to the camera. The method is robust to varying illumination, since the data we used was acquired under different illuminations, with or without light in different rooms and with different background (inhomogeneous). The unified embedding using ISOMAP combined with the nonlinear RBF mapping make our method person-independent regardless of whether the person is in our database. In addition, our system is also robust to small facial expression changes, since the training data we used to learn the non-linear mapping includes those where the person is smiling..

However, since our person-independent mapping system is based on an interpolative system, the results may degrade if the test images or sequences were not well represented in the original training space (which cause extrapolation). This can be explained by the fact that the RBF interpolation uses Gaussian kernels, where the outputs can be very small if the input data is far away from any of the centers. On the contrary, if the input data is well represented by the training data, the estimation results will be very good, such as for person (e) in Fig. 5.4.

Here, to simplify the problem, we use head sequences taken under the assumption that no upward or downward motion is included. This is to simplify the mapping by ISOMAP, where we need only a 2-D space to represent the dimensionality-reduced head sequences. If upward and downward motions are included, the problem will become complex where the dimensionality of the embedded space will be increased. However, we believe this problem can be solved by introducing some more complex algorithms, which sets up a future work for us.

As to our CPFA method, for FCFA, the person frequently changes his head pose (this can be achieved by rotating his head or rotating his body or both), which results in the similarity matrix R of the person demonstrating cyclicity. However, for focused

attention, the person seldom rotates his head, resulting in the similarity matrix R demonstrating little or no cyclicity. Thus, after 2-D Fourier Transform of the similarity matrix and normalization over the total energy, DC component and the magnitudes of the three lowest frequencies were found to be suitable features for classification.

Our CPFA algorithm is robust to low resolution and varying illumination. The lowest resolution of the head was 25×15 in the experiments. In addition, the similarity matrix R' is noise tolerant since PCA can denoise the raw data. Furthermore, our algorithm is robust to error in head location by searching for the minimal $S^{'}$ in a small area to reduce the location error.

Here, in both methods, we assume that the direction of visual attention is fully characterized by the head pose and do not consider eye gaze. We did not consider eye gaze detection as the head images we used in the experiment were relatively small and sometimes the eyes were not clear, making gaze detection very difficult. Besides, in many cases, in order to look at a big area, it is more convenient for people to change the head pose rather than eye gaze, which motivated the development of the proposed method.

Chapter 6

Conclusion

Attentive behavior detection is useful for human computer interaction. Knowing where a person is looking at can further improve the interactivity. It can be useful in remote learning systems to know if students are focusing on the lecture or inferring whether a product is attractive to people in the advertising documents; or for video surveillance, to know whether the attentive behavior of the person is abnormal. To infer this behavior, we have presented two different systems to detect FCFA.

In our HPE system, we use ISOMAP to embed each individual's high dimensional head image data into a low dimensional (2-D) space. By ellipse fitting, we normalize by reshaping, rotating, and mirror imaging if needed, the individual embedded space to find a unified embedded space. A RBF interpolation technique is used to find a person-independent mapping for new input head image data into the unified embedding space, i.e. our feature space. For head image sequences, we propose an adaptive local fitting algorithm to remove outliers and to smooth the output of RBF interpolation. The head pose estimate in each frame is then obtained by a simple coordinate-angle converter. To detect FCFA behavior from video sequences, the entropy of the head pose estimates over the entire sequence is used to classify the sequence as a FCFA or focused attention behavior. The experiment results show that our HPE method can very well estimate the head pose even when the head is turned back to the camera and by setting a

threshold of $E_0 = 2.5$ on the head pose angle entropy, we can successfully detect FCFA behavior.

For our CPFA method, by foreground segmentation and edge detection, we locate the head in each frame of the sequence. A similarity matrix is computed in a 9-dimensional principal components subspace as the head pose evolves over time. A 2-D frequency analysis is applied on the similarity matrix for feature extraction. Finally, K-NNR is proposed to differentiate FCFA from focused attention. The experiment results show our CPFA method achieved an average classification accuracy of 96.8% on 31 video sequences and the computational time for a 40s video sequence is 73.4s in Matlab.

Future work includes extending our HPE method to a system that can also work with different tilt angles of the head and large facial expressions such as laughs, which we believe can be done with a larger training data of more people with different tilt angles and different facial expressions. Furthermore, our CPFA method can also be extended to video summarization and segmentation.

Bibliography

[1] R. Cutler, L. Davis, "Robust Real-Time Periodic Motion Detection, Analysis, and Applications", In *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 22, No. 8, pp. 781-796, August 2000.

[2] M. Allmen, "Image Sequence Description Using Spatiotemporal Flow Curves: Toward Motion-Based Recognition," PhD thesis, Univ. of Wisconsin, Madison, 1991.

[3] C. Cohen, L. Conway, and D. Koditschek, "Dynamic System Representation, Generation, and Recognition of Besic Oscillatory Motion Gestures," *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, 1996.

[4] R. Cutler and L. Davis, "View-Based Detection and Analysis of Periodic Motion," *Proc. Int'l Conf. Pattern Recognition*, p. SA14, 1998.

[5] R. Cutler and L. Davis, "Real-Time Periodic Motion Detection, Analysis, and Applications," *Proc. Computer Vision and Pattern Recognition*, pp. 326-332, June 1999.

[6] H. Fujiyoshi an dA. Lipton, "Real-Time Human Motion Analysis by Image Skeletonization," *Proc. IEEE Workshop Applications of Computer Vision*, p.session 1A, 1998.

[7] B. Heisele and C. Wohler, "Motion-Based REcognition of Pedestrains," *Proc. Int'l Conf. Pattern Recognition*, 1998

[8] F. Liu, R. Picard, "Finding Periodicity in Space and Time", In *Proc. Int'l Conf. Computer Vision*, pp. 376-383, January, 1998.

[9] D. McReynolds and D. Lowe, "Rigidity Checking of 3D Point Correspondences Under Perspective Projection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 12, pp. 1,174-1,185, 1996.

[10] S. Niyogi and E. Adelson, "Analyzing and Recognizing Walking Figures in XYT," *Proc. Int'l Conf. Computer Vision and Pattern Recognition*, pp. 469-474, Dec. 1994.

[11] S. Niyogi and E. Adelson, "Analyzing Gait with Spatiotemporal Surfaces," *Proc. IEEE Workshop Motion of Non-Rigid and Articulated Objects*, pp. 64-69, 1994.

[12] R. Polana, R. Nelson, "Detection and Recognition of Periodic, Non-Rigid Motion", In *Int'l J. Computer Vision*, Vol. 23, No. 3, pp. 261-282, June 1997.

[13] S. Seitz, C. Dyer, "View-Invariant Analysis of Cyclic Motion", In *Int'l J. Computer Vision*, Vol. 25, No. 3, pp. 1-23, 1997.

[14] A. Selinger and L. Wixson, "Classifying Moving Objects as Rigid or Non-Rigid without Correspondences," *Proc. DARPA Image Understanding Workshop*, pp. 341-347, Nov. 1998.

[15] P. Tsai, M. Shah, K. Keiter, and T. Kasparis, "Cyclic Motion Detection for Motion Based Recognition," *Pattern Recognition*, vol. 27, no. 12, pp. 1,591-1,603, 1994.

[16] T. Martinetz and K. Schulten, "Topology Representing Networks," *Neural Networks*, vol. 7, no. 3, pp. 507-522, 1994.

[17] J. Tenenbaum, "Mapping a manifold of perceptual observations," In *Neural Information Precessing Systems*, vol. 10, MIT Press, 1997.

[18] J. Tenenbaum, V. de Silva, and J. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science*, 290(5500), pp. 2319-2323, December 2000.

[19] J. Tenenbaum, et al. "The Isomap Algorithm and Topological Stability," *Science*, vol. 295, Jan. 2002.

[20] S. Roweis, and L. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science*, 290(5500), pp. 2323-2326, Dec. 2000.

[21] T. Cox and M. Cox, "Multidimensional Scaling," *Number 59 in Monographs on Statistics and Applied Probability*, Chapman & Hall, 1994.

[22] J. Kruskal, and M. Wish, "Multidimensional Scaling,", *Sage*, 1978.

[23] J. Kruskal, "Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis," *Psychometrika*, vol. 29, pp. 1-27, 1964.

[24] http://www.isinspect.org.uk/reports/2004/0374_04_r.htm

[25] B. Blumberg, et. al., "Creature Smarts: the Art and Architecture of a Virtual Brain," in *proc. Game Developers Conf.*, pp. 147-166, 2000.

[26] J. Tsotsos et al., "Modeling Visual Attention via Selective Tuning", In *Artificial Intelligence*, Vol. 78, pp. 507-545, 1995.

[27] S. Baluja, and D. Pomerleau, "Dynamic Relevance: Vision-Based Focus of Attention using Artificial Neural Networks," *Artificial Intelligence*, 97, pp. 381-395, 1997.

[28] S. Baluja, and D. Pomerleau, "Expectation-Based Selective Attention for Visual Monitoring and Control of a Robot Vehicle," *Robotics and Autonomous Systems*, 22, pp. 329-344, 1997.

[29] P. Burt, "Attention Mechanisms for Vision in a Dynamic World," *Proc. Ninth Int'l Conf. on Pattern Recognition*, Beijing, China, pp. 977-987, 1988.

[30] J. Clark, and N. Ferrier, "Modal Control of an Attention Vision System," *Proc. IEEE Int'l Conf. Computer Vision*, Tarpon Springs, FL., pp. 514-523, 1988.

[31] J. Clark, "Spatial Attention and Latencies of Saccadic Eye Movements," *Vision Research*, 39(3), pp. 583-600, 1998.

[32] V. Concepcion, and H. Wechesler, "Detection and Localization of Objects in Time-Varying Imagery using Attention, Representation and Memory Pyramids," *Pattern Recognition*, 29(9), pp. 1543-1557, 1996.

[33] W. Cowan, "Evolving Conceptions of Memory Storage, Selective Attention and Their Mutual Constrains within the Human Information-Processing System," *Psychol. Bull.*, 104, pp. 163-191, 1988.

[34] F. Crick, and C. Koch, "Towards a Neurobiological Theory of Consciousness," *Seminars in the Neurosciences*, 2, pp. 263-275, 1990.

[35] J. Duncan, et al. "Integrated Mechanisms of Selective Attention," *Curr. Opin. Biol.*, 7, pp. 255-261, 1997.

[36] S. Exel, and L. Pessoa, "Attention Visual Recognition," *Int'l Conf. Pattern Recognition*, Brisbane, Australia, 1998.

[37] S. Grossberg, et al. "A Neural Theory of Attentive Visual Search: Interactions of Boundary, Surface, Spatial and Object Representations," *Psychological Review*, 10(3), pp. 470-489, 1994.

[38] S. Grossberg, "How Does the Cerebral Cortex Work? Learning, Attention, and Grouping by the Laminar Circuits of Visual Cortex," *Spatial Vision*, 12(2), pp. 13-185, 1999.

[39] W. Grimson, et al. "An Active Visual Attention System to Play "Where's Waldo"," *Proc. Conf. Computer Vision and Pattern Recognition*, Seattle, WA, 85-90, 1994.

[40] T. Grove, and R. Fisher, "Attention in Iconic Object Matching," *Proc. BMCV96*, Edinburgh, pp. 293-302, 1996.

[41] G. Humphreys, "SEarch via Recursive Rejection (SERR): A Connectionist Model of Visual Search," *Cognitive Psychology*, 25, pp. 43-110, 1993.

[42] Y. Kazanovich, and R. Borisyuk, "Dynamics of Neural Networks with a Central Element," *Neural Networks*, 12, pp. 441-454, 1999.

[43] C. Koch, and S. Ullman, "Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry," *Human Neurobiology*, 4, pp. 219-227, 1985.

[44] L. Itti, C. Koch, and E. Niebur, "A Model of Saliency-Based Visual Attention for Rapid Scene Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(11), pp. 1254-1259, 1998.

[45] V. Kryukov, "An Attention Model Based on the Principle of Dominanta," A. Holden and V. Kryukov (eds.) *Neurocomputers and Attention I: Neurobiology, Synchronization and Chaos*, Manchester: Manchester University Press, pp. 319, 1991.

[46] E. Niebur, et al. "An Oscillation Based Model for the Neuronal Basis of Attention," *Vision Research*, 33, pp. 2789-2802, 1993.

[47] E. Niebur, and C. Koch, "A Model for the Neuronal Implementation of Selective Visual Attention Based on Temporal Correlation among Neurons," *J. Neurosci.*, 1, pp. 141-158, 1994.

[48] B. Olshausen, et al. "A Neurobiological Model of Visual Attention and Invariant Pattern Recognition Based on Dynamic Routing of Infomation," *J. Neurosci.*, 13(11), pp. 4700-4719, 1993.

[49] E. Postma, et al. " SCAN: A Scalable Model of Attentional Selection," *Neural Networks*, 10 pp. 993-1015, 1997.

[50] A. Ratan, "The Role of Fixation and Visual Attention in Object Recognition," MIT AI-TR-1529, July, 1995.

[51] I. Rybak, et al. "A Model of Attention-Guided Visual Perception and Recognition," *Vision Reseach*, 38, pp. 2387-2400, 1998.

[52] G. Sela, and M. Levine, "Real-Time Attention for Robotic Vision," *Real-Time Imaging*, 3, pp. 173-194, 1997.

[53] B. Takacs, and H. Wechsler, "A Dynamic and Multiresolution Model of Visual Attention and Its Application to Facial Landmark Detection," *Computer Vision and Image Understanding*, 70(1), pp. 63-73, 1998.

[54] C. Westin, et al. "Attention Control for Robot Vision," *Proc. IEEE Computer Vision and Pattern Recognition*, San Francisco, CA, pp. 18-20, 1996.

[55] J. P. Gottlieb, et al. "The Representation of Visual Salience in Monkey Parietal Cortex," *Nature*, 391(6666), pp. 481-484, 1998.

[56] D. J. Robinson, and S. E. Peterson, "The Pulvinar and Visual Salience," *Trends in Neuroscience*, 15(4), pp. 127-132, 1992.

[57] W. Singer, and C. W. Gray, "Visual Feature Integration and the Temporal Correlation Hypothesis," *Annu. Rev. Neurosci.*, 18, pp. 555-86, 1995.

[58] M. Usher, and N. Donnelly, "Visual Synchrony Affects Binding and Segmentation in Perception," *Nature*, 394, pp. 179-182, 1998.

[59] R. Yang and Z. Zhang, "Model-based Head Pose Trackin With Stereovision," *FG 2002*, pp. 255-260, Washington DC, 2002.

[60] Y. Matsumoto, and A. Zelinsky, "An Algorithm for Real-time Stereo Vision Implementation of Head Pose and Gaze Direction Measurement," *FG 2000*, pp.499-505, 2000.

[61] S. Srinivasan and K. L. Boyer, "Head Pose Estimation Using View Based Eigenspaces," *ICPR 2002*, Quebec, 2002.

[62] Morency, Sundberg, Darrel, "Pose Estimation using 3D View-Based Eigenspaces," *IEEE Intl. Workshop on Analysis and Modeling of Faces and Gestures*, pp. 45-52, Nice, 2003.

[63] M. Harville, A. Rahimi, T. Darell, G. Gordon, J.Woodfill, "3D Pose Tracking with Linear Depth and Brightness Constraints," *ICCV'99*, Corfu, Greece, 1999.

[64] Morency, Rahimi, Checka and Darrell, "Fast stereobased headtracking for interactive environment," *FG 2002*, Washington DC, 2002.

[65] Rainer Stiefelhagen, Jie Yang, Alex Waibel, "Simultaneous Tracking of Head Poses in a Panoramic View," *ICPR 2000*, Barcelona, Spain, 2000.

[66] E. Seemann, K. Nickel, and R. Stiefelhagen, "Head Pose Estimation Using Stereo Vision for Human-Robot Interaction," *FG 2004*, Seoul, Korea, 2004.

[67] Q. Chen, H. Wu, T. Fukumoto, and M. Yachida, "3D head pose estimation without feature tracking," In *Proc. Int'l Conf. on Autom. face and Gesture Recog.*, pp. 88-93, 1998.

[68] T. Cootes, G. Edwards, and C. Taylor, "Active Appearance Models," In *Proc. European Conf. on Computer Vision*, pp. 484-498, 1998.

[69] S. Basu, I. Essa, A. Pentland, "Motion Regularization for Model-Based Head Tracking", In *Proc. IEEE Int'l Conf. Pattern Recognition*, Vol. 3, pp. 611-616, August 1996.

[70] J. Heinzmann, and A. Zelinsky, "3-D Facial Pose and Gaze Point Estimation Using a Robust Real-Time Tracking Paradigm," In *Proc. 3rd Int'l Conf. Automatic Face and Gesture Recognition*. Los Alamitos, CA, pp. 142-147, 1998.

[71] T. Jebara, and A. Pentland, "Parameterized Structure from Motion for 3D Adaptive Feedback Tracking of Faces," in *IEEE conf. Computer Vision and Pattern Recognition*, 1997.

[72] M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, Reliable Head Tracking Under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 4, April, 2000.

[73] M. Malciu and F. Preteux, "A Robust Model-Based Approach for 3D Head Tracking in Video Sequences," in *Proc. 4th Int'l Conf. Autom. Face and Gesture Recog.*, pp. 169-174, Grenoble, France, 2000.

[74] Z. Zivkovic, and F. van der Heijden, "A Stabilized Adaptive Appearance Changes Model for 3D Head Tracking," in *Proc. 2nd IEEE Int'l Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Realtime Systems*, pp. 175-181, Vancouver, Canada, 2001.

[75] Y. Wu, K, Toyama, "Wide-Range Person and Illumination-Insensitive Head Orientation Estimation", In *Proc. Fourth Int'l Conf. Automatic Face and Gesture Recognition*, pp. 183-188, March 2000.

[76] L. Brown and Y. Tian, "Comparative Study of Coarse Head Pose Estimation," in *Proc. Workshop on Motion and Video Computing*, pp. 125-130, 2002.

[77] R. Rae, H. Ritter, "Recognition of Human Head Orientation Based on Artificial Neural Networks", In *IEEE Trans. Neural Networks*, Vol. 9, No. 2, pp. 257-265, March 1998.

[78] L. Zhao, G. Pingali, I. Carlbom, "Real-Time Head Orientation Estimation Using Neural Networks", In *Proc. Int'l Conf. Image Processing*, September 2002.

[79] V. Krüger, S. Bruns, G. Sommer, "Efficient Head Pose Estimation With Gabor Wavelets", In *Proc. 11th British Machine Vision Conference*, Vol. 1, pp. 72-81, September 2000.

[80] R. Stiefelhagen, "Tracking Focus of Attention in Meetings", In *Proc. Fourth IEEE Int'l Conf. Multimodal Interfaces*, pp. 273-280, October, 2002.

[81] R. Gonzalez, R. Woods, "Digital Image Processing", *Addison-Wesley Publishing Company*, 1992.

[82] R. Duda, P. Hart, D. Stork, "Pattern Classification", 2nd Edn., *John Wiley & Sons, Inc.*, 2000.

[83] Z. Zeng, S. Ma, "Head Tracking by Active Particle Filtering", In *Proc. Fifth IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pp. 82-87, May 2002.

[84] M. Yang, D. Kriegman, and N. Ahuja, "Detecting Faces in Images: A Survey," in *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34-58, Jan 2002.

[85] Fitzgibbon, M. Pilu , R.Fisher "Direct least-square fitting of Ellipses" , *IEEE PAMI*, vol. 21, no. 5, pp. 476-480, June 1999.

[86] Hutcheson, M.C.,"Trimmed Resistant Weighted Scatterplot Smooth," *Master's Thesis*, Cornell University, Ithaca, NY, 1995.