ABSTRACT

Title of dissertation: FACE RECOGNITION AND FACIAL ATTRIBUTE ANALYSIS FROM UNCONSTRAINED VISUAL DATA

Huy Tho Ho, Doctor of Philosophy, 2014

Directed by: Professor Rama Chellappa
Department of Electrical and Computer Engineering

Analyzing human faces from visual data has been one of the most active research areas in the computer vision community. However, it is a very challenging problem in unconstrained environments due to variations in pose, illumination, expression, occlusion and blur between training and testing images. The task becomes even more difficult when only a limited number of images per subject is available for modeling these variations. In this dissertation, different techniques for performing classification of human faces as well as other facial attributes such as expression, age, gender, and head pose in uncontrolled settings are investigated.

In the first part of the dissertation, a method for reconstructing the virtual frontal view from a given non-frontal face image using Markov Random Fields (MRFs) and an efficient variant of the Belief Propagation (BP) algorithm is introduced. In the proposed approach, the input face image is divided into a grid of overlapping patches and a globally optimal set of local warps is estimated to synthesize the patches at the frontal view. A set of possible warps for each patch is

obtained by aligning it with images from a training database of frontal faces. The alignments are performed efficiently in the Fourier domain using an extension of the Lucas-Kanade (LK) algorithm that can handle illumination variations. The problem of finding the optimal warps is then formulated as a discrete labeling problem using an MRF. The reconstructed frontal face image can then be used with any face recognition technique. The two main advantages of our method are that it does not require manually selected facial landmarks as well as no head pose estimation is needed.

In the second part, the task of face recognition in unconstrained settings is formulated as a domain adaptation problem. The domain shift is accounted for by deriving a latent subspace or domain, which jointly characterizes the multifactor variations using appropriate image formation models for each factor. The latent domain is defined as a product of Grassmann manifolds based on the underlying geometry of the tensor space, and recognition is performed across domain shift using statistics consistent with the tensor geometry. More specifically, given a face image from the source or target domain, multiple images of that subject are first synthesized under different illuminations, blur conditions, and 2D perturbations to form a tensor representation of the face. The orthogonal matrices obtained from the decomposition of this tensor, where each matrix corresponds to a factor variation, are used to characterize the subject as a point on a product of Grassmann manifolds. For cases with only one image per subject in the source domain, the identity of target domain faces is estimated using the geodesic distance on product manifolds. When multiple images per subject are available, an extension of kernel discriminant

analysis is developed using a novel kernel based on the projection metric on product spaces. Furthermore, a probabilistic approach to the problem of classifying image sets on product manifolds is introduced.

Understanding attributes such as expression, age class, and gender from face images has many applications in multimedia processing including content personalization, human-computer interaction, and facial identification. To achieve good performance in these tasks, it is important to be able to extract pertinent visual structures from the input data. In the third part of the dissertation, a fully automatic approach for performing classification of facial attributes based on hierarchical feature learning using sparse coding is presented. The proposed approach is generative in the sense that it does not use label information in the process of feature learning. As a result, the same feature representation can be applied for different tasks such as expression, age, and gender classification. Final classification is performed by linear SVM trained with the corresponding labels for each task.

The last part of the dissertation presents an automatic algorithm for determining the head pose from a given face image. The face image is divided into a regular grid and represented by dense SIFT descriptors extracted from the grid points. Random Projection (RP) is then applied to reduce the dimension of the concatenated SIFT descriptor vector. Classification and regression using Support Vector Machine (SVM) are combined in order to obtain an accurate estimate of the head pose. The advantage of the proposed approach is that it does not require facial landmarks such as the eye and mouth corners, the nose tip to be extracted from the input face image as in many other methods.

FACE RECOGNITION AND FACIAL ATTRIBUTE ANALYSIS FROM UNCONSTRAINED VISUAL DATA

by

Huy Tho Ho

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2014

Advisory Committee:
Professor Rama Chellappa, Chair/Advisor
Professor Larry Davis
Professor Min Wu
Professor Amitabh Varshney
Professor Ramani Duraiswami

This dissertation is dedicated to my parents.

Acknowledgments

I would like to express my deepest gratitude to my advisor, Prof. Rama Chellappa, for his mentoring and guidance over the years. Without his constant support and guidance, this dissertation would not have been completed. I am also very thankful to Prof. Chellappa for providing me with the freedom to pursue my research interests as well as a wonderful research environment.

It was an honor to have Prof. Larry Davis, Prof. Min Wu, Prof. Amitabh Varshney, and Prof. Ramani Duraiswami on my dissertation committee. I am very thankful to them for serving on my committee and providing valuable suggestions to improve this dissertation. I am also grateful to all the professors whose classes I took during my graduate studies for their enlightening lectures that help provide me with a solid background for my PhD and future research.

I would like to express my gratitude to Dr. Behzad Shahraray, Dr. Raghuraman Gopalan, Dr. Zhu Liu, and David Gibbon at AT&T Labs for their guidance and support during my two internships.

I would like to thank my labmates and friends including Dr. Vishal Patel, Dr. Hien Nguyen, Dr. David Shaw, Chris Reale, Garrett Warnell, Priyanka Vageeswaran, Sumit Shekhar, Ashish Srivastava and others for making my graduate life memorable.

I would like to thank the staff in UMIACS and ECE. Special thanks are due to Janice Perrone, Melanie Prange, Maria Hoo and Arlene Schenk.

I thank my grandmother, my parents and my brother for their unconditional love and support. Finally, I thank my wife, Thanh, for taking care of me and always being with me during the ups and downs.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Face recognition and facial attribute analysis have been key research areas in computer vision and pattern recognition for more than two decades. Their applications can be found in multimedia, telecommunications, law enforcement, biometrics and surveillance. Although there have been some early successes in automatic face recognition and classification of facial attributes such as expression, age, gender, and head pose from visual data, these problems are still far from being completely solved, especially in uncontrolled environments. In fact, the performance of existing automatic facial analysis systems drops significantly when there are variations in pose, illumination, expression and blur conditions [10]. The tasks become even more challenging when only a limited number of images per subject is available for modeling these variations.

In this dissertation, different approaches to the problems of face recognition and facial attribute analysis in unconstrained settings are investigated. First, a method for synthesizing the virtual frontal view from a given non-frontal face image using Markov Random Fields (MRFs) and an efficient variant of Belief Propagation (BP) is proposed. It can be combined with any face recognition technique in order to handle the case where the probe face image is non-frontal. In the second part of

the dissertation, the task of face recognition in unconstrained settings is formulated as a domain adaptation problem where domain shifts are due to multiple factor variations such as illumination, blur and alignment between the probe and gallery images. Rather than ignoring the geometrical structures of the image space as in many traditional approaches, the proposed algorithm constructs the latent domain as a product of Grassmann manifolds based on the underlying geometry of the tensor space. The third part of the dissertation presents a hierarchical feature learning approach for performing classification of facial attributes including expression, age class, and gender. Rather than using hand-crafted features such as SIFT [11] or LBP [12], feature representations are obtained using dictionary learning, sparse coding, and spatial pooling over a hierarchical network on the training images. Obtaining the information about the head orientation has become a crucial pre-processing step in many pose-invariant face recognition algorithms [13]. In the last part of the dissertation, a technique for automatically estimating the head pose from an input face image is presented.

The above topics are briefly discussed in the remaining of this chapter.

1.1 Pose-Invariant Face Recognition using Markov Random Fields

In the first part of the dissertation, a patch-based method for synthesizing the virtual frontal view from a given non-frontal face image using MRFs and an efficient variant of the BP algorithm is investigated. By aligning each patch in the input image with images from a training database of frontal faces, a set of possible warps

is obtained for that patch. The alignments are then carried out efficiently using an illumination insensitive extension of the Lucas-Kanade (LK) algorithm [14] in the frequency domain. The objective of the algorithm is to find the globally optimal set of local warps that can be used to predict the image patches at the frontal view. This goal is achieved by considering the problem as a discrete labeling problem using an MRF. In our approach, the cost functions of the MRF are not just the simple sum of squared differences (SSD) between patches but are modified to reduce the effect of illumination variations. The optimal labels are obtained using a variant of the BP algorithm with *message scheduling* and *dynamic label pruning* [15]. The two main advantages of our approach over other state-of-the-art algorithms are that: (1) it does not require manually selected landmarks, and (2) no global geometric transformation is needed.

1.2 Model-Driven Domain Adaptation on Product Manifolds for Unconstrained Face Recognition

In the second part, a domain adaptive approach for face recognition using tensor geometry corresponding to models explaining facial variations, with as few as a single image per subject in the source domain, is discussed. In the proposed method, a latent domain where multifactor facial variations across the source and target domains can be captured together is constructed instead of finding linear transformations representing domain shifts as in [16, 17]. This latent domain is defined as a product of Grassmann manifolds based on the underlying geometry of

the tensor space. More specifically, multiple images of the same subject are synthesized from a given face image under different illuminations, blur conditions, and 2D perturbations to form a tensor representation of the face. The subject is then characterized as a point on a product of Grassmann manifolds by mapping the orthogonal matrices obtained from the decomposition of the tensor to the factor manifolds. Geodesic distance on product manifolds is used to perform face recognition for cases with only one image per subject available in the training. When multiple images per subject are available, an extension of kernel discriminant analysis is developed using a novel kernel based on the projection metric on product spaces. Furthermore, a probabilistic approach for performing image set classification using the Kullback-Leibler divergence as a distance measure in the projection space is also presented.

1.3 Hierarchical Feature Learning using Sparse Coding for Facial Semantic Analysis

In the third part of the dissertation, a fully automatic approach for performing classification of different attributes including expression, age class, and gender from face images using hierarchical feature learning is presented. The feature representations are obtained from the training data using dictionary learning, sparse coding, and spatial pooling. As label information is not used in the process of feature learning, the learned feature representation is generative and can be used for different classification tasks. Final classification is performed by linear SVM trained with the

corresponding labels for each task. As the features are learned using a network of multiple layers with spatial pooling at different neighborhood sizes, the proposed method is able to better capture the richness of visual data at multiple scales. Furthermore, the proposed method is fully automatic requiring no human intervention, and a single set of configuration parameters was used for all experiments.

1.4 Head Pose Estimation using Randomly Projected Dense SIFT Descriptors

Finally, an automatic method for estimating the head pose from a single 2D face image is presented. Dense SIFT descriptors [18] are extracted from image grid points in order to obtain a representation that is robust to noise and illumination variations. Random Projection (RP) is used to reduce the dimension of the concatenated descriptor vector for efficient processing. In order to better approximate the head pose, a combination of Support Vector Machine (SVM) [19] and Support Vector Regression (SVR) [20] is employed to infer a continuous mapping function from the image to the pose space. The advantage of the proposed approach is that it does not depend on the extraction of facial feature points such as the mouth and eye corners and the nose tip, which by itself is a challenging process.

1.5 Organization of the Dissertation

The dissertation is organized as follows. Chapter 2 discusses the patch-based method for frontal face reconstruction from non-frontal face images. The model-

driven domain adaptation approach for unconstrained face recognition on product manifolds is described in Chapter 3. In Chapter 4, the hierarchical feature learning method for facial attribute analysis is presented. The automatic head pose estimation method using dense SIFT descriptors and random projection is discussed in Chapter 5. Finally, directions for future work are given in Chapter 6.

Chapter 2: Pose-Invariant Face Recognition using Markov Random Fields

2.1 Introduction

Pose variations can be considered as one of the most important and challenging problems in face recognition. As the viewpoint varies, the 2D facial appearance will change because the human head has a complex non-planar geometry. Magnitudes of variations of innate characteristics, which distinguish one face from another, are often smaller than magnitudes of image variations caused by pose variations [21]. Popular frontal face recognition algorithms, such as Eigenfaces [22] or Fisherfaces [23, 24], usually have low recognition rates under pose changes as they do not take into account the 3D alignment issue when creating the feature vectors for matching.

In this chapter, a patch-based method for synthesizing the virtual frontal view from a given non-frontal face image using MRFs and an efficient variant of the BP algorithm is proposed. By aligning each patch in the input image with images from a training database of frontal faces, a set of possible warps is obtained for that patch. The alignments are then carried out efficiently using an illumination invariant extension of the Lucas-Kanade (LK) algorithm [14] in the frequency domain. The

objective of the algorithm is to find the globally optimal set of local warps that can be used to predict the image patches at the frontal view. This goal is achieved by considering the problem as a discrete labeling problem using an MRF. In the proposed approach, the cost functions of the MRF are not just the simple sum of squared differences (SSD) between patches but are modified to reduce the effect of illumination variations. The optimal labels are obtained using a variant of the BP algorithm with *message scheduling* and *dynamic label pruning* [15]. The two main advantages of our approach over other state-of-the-art algorithms are that: (1) it does not require manually selected landmarks, and (2) no global geometric transformation is needed. Experimental results on the FERET [1], CMU-PIE [2] and Multi-PIE [3] databases are presented to demonstrate the effectiveness of the proposed algorithm.

Organization of the chapter: Related works are discussed in Section 2.2. Section 2.3 describes the illumination-insensitive alignment method based on the LK algorithm. The reconstruction of the virtual frontal view using MRFs and BP is discussed in Section 2.4. Finally, in Section 2.5, we present experimental results in both frontal face reconstruction and pose-invariant face recognition.

2.2 Related Work

Existing methods for face recognition across pose can be roughly divided into two broad categories: (1) techniques that rely on 3D models and (2) 2D techniques. In the first type of approaches, the morphable model proposed by Blanz and Vet-

ter [25] fits a 3D model to an input face using the prior knowledge of human faces and image-based reconstruction. The main drawback of this algorithm is that it requires many manually selected landmarks for initialization. Furthermore, the optimization process is computationally expensive and often converges to local minima due to a large number of parameters that need to be determined. Another recently proposed method by Biswas and Chellappa [26] estimates the facial albedo and pose at the same time using a stochastic filtering framework and performs recognition on the reconstructed frontal faces. The disadvantage of this approach lies in the use of an iterative algorithm for updating the albedo and pose estimates leading to accumulation of errors from step to step. Given a non-frontal face image, the 3D pose normalization algorithm proposed by Asthana *et al.* [27] uses the pose-dependent correspondences between 2D landmark points and 3D model vertices in order to synthesize the frontal view. The main drawback of this method is the dependence on the fitting of landmarks using the Active Appearance Model (AAM) [28].

On the other hand, 2D techniques do not require the 3D prior information for performing pose-invariant face recognition. The AAM algorithm proposed by Cootes *et al.* [28] fits a statistical appearance model to the input image by learning the relationship between perturbations in the model parameters and the induced image errors. The main disadvantage of this approach is that each training image requires a large number of manually annotated landmarks. Gross *et al.* [29] proposed the eigen light-field (ELF) method that unifies all possible appearances of faces in different poses within a 4D space (two viewing directions and two pixel positions). However, this method discards shape variations due to different identity as it requires

a restricted alignment of the image to the light field space. Recently, Prince *et. al.* [30] use an affine mapping and pose information to generate the observation space from the identity space. In the approach proposed by Castillo and Jacobs [31], the cost of stereo matching was used in face recognition across pose without performing 3D reconstruction. Sarfraz and Hellwich [32] try to solve the problem by modeling the joint appearance of gallery and probe images across pose in a Bayesian framework.

Patch-based approaches for face recognition under varying poses have received significant attention from the research community. The main motivation in these approaches is that a 3D face is composed of many planar local surfaces and thus, an out-of-plane rotation, although non-linear under 2D imaging projection, can be approximated by linear transformations of 2D image patches. As a result, modeling a face as a collection of subregions/patches is more robust to pose variations than the holistic appearance. In the method proposed by Kanade and Yamada [33], each patch has a utility score based on pixel differences, and the recognition is performed using a Gaussian probabilistic model and a Bayesian classifier. Ashraf *et al.* [34] extended this approach by learning the patch correspondences based on 2D affine transforms. The problem with these approaches is that the transformations are optimized locally without taking into account the global consistency of the patches. In [35], linear regressions are performed on local patches in order to synthesize the virtual frontal view. Another approach proposed by [36] measures the similarities of local patches by correlations in a subspace constructed by Canonical Correlation Analysis (CCA). However, the common drawback of these two algorithms is that

the head pose of the input face image needs to be known a priori. Arashloo and Kittler [37] present a method for estimating the deformation parameters of local patches using Markov Random Fields (MRFs). The disadvantage of this approach is that it depends on estimating the global geometric transformation between the template and the target images. Although designed specifically for handling expression variations in face recognition, another related work is the method proposed by Liao and Chung [38], which formulates the face recognition problem as a deformable image registration problem using MRFs. However, this approach also depends on the extraction of salient regions from face images.

2.3 Illumination Insensitive Patch Alignment

2.3.1 Alignment of Local Patches using Weighted Lucas-Kanade

Assume that we have two images, the probe image I and the gallery image T, captured at two different viewpoints. The images are divided into M blocks (rectangle patches) and for each pair of corresponding patches, I_i and T_i, a local warp W_i is estimated to align them using the weighted Lucas-Kanade (LK) algorithm [39]. The warp W_i, parameterized by the vector $\mathbf{p}_i$, minimizes the following error function

$$E_i(\mathbf{p}_i) = ||I_i(\mathbf{p_i}) - T_i(\mathbf{0})||^2_{\mathbf{Q}} = [I_i(\mathbf{p}_i) - T_i(\mathbf{0})]^T \mathbf{Q} [I_i(\mathbf{p}_i) - T_i(\mathbf{0})] \qquad (2.1)$$

where $\mathbf{Q}$ is a symmetric, positive semi-definite weighting matrix. Note that $I_i(\mathbf{p_i})$ and $T_i(\mathbf{0})$ are both vectorized image patches. Equation (2.1) becomes the standard LK objective function [40] when $\mathbf{Q}$ is an identity matrix. If $W(\mathbf{p})$ is an affine warp

with parameters $\mathbf{p} = (p_1, p_2, p_3, p_4, p_5, p_6)^T$, it can be written as

$$W(\mathbf{p}) = \begin{pmatrix} 1+p_1 & p_3 & p_5 \\ p_2 & 1+p_4 & p_6 \end{pmatrix}.$$

The transformed image patch $I_i(\mathbf{p_i})$ is obtained by applying the warp to all the pixels in I_i.

Equation (2.1) is highly non-linear and thus, can be linearized by performing the first order Taylor expansion on $I_i(\mathbf{p}_i + \Delta\mathbf{p}_i)$

$$E_i(\mathbf{p}_i) \approx ||I_i(\mathbf{p}_i) + \mathbf{J}_i\Delta\mathbf{p}_i - T_i(\mathbf{0})||^2_{\mathbf{Q}} \tag{2.2}$$

where $\mathbf{J}_i = \left(\frac{\partial I_i(\mathbf{p}_i)}{\partial \mathbf{p}_i}\right)^T$ is the Jacobian of $I_i(\mathbf{p}_i)$. The value of $\Delta\mathbf{p}_i$ that minimizes (2.2) is given by

$$\Delta\mathbf{p}_i = \mathbf{H}_i^{-1}\mathbf{J}_i^T\mathbf{Q}[T_i(\mathbf{0}) - I_i(\mathbf{p}_i)] \tag{2.3}$$

where the pseudo-Hessian matrix is defined as

$$\mathbf{H}_i = \mathbf{J}_i^T\mathbf{Q}\mathbf{J}_i = \frac{\partial I_i(\mathbf{p}_i)}{\partial \mathbf{p}_i}\mathbf{Q}\frac{\partial I_i(\mathbf{p}_i)}{\partial \mathbf{p}_i}^T \quad . \tag{2.4}$$

An iterative solution to (2.1) can be obtained by iteratively solving for $\Delta\mathbf{p}_i$ and updating the warp parameters $\mathbf{p}_i = \mathbf{p}_i + \Delta\mathbf{p}_i$ until convergence.

2.3.2 Illumination Insensitive Alignment based on Gabor Features

It is known that the original LK algorithm is very sensitive to changes in illumination [41]. The main advantage of the weighted LK algorithm over the original method is that illumination variations can be handled by encoding the prior knowledge of the correlation and salience of image pixels into $\mathbf{Q}$. As a result, choosing

an appropriate weighting matrix $\mathbf{Q}$ is an important problem with the weighted LK algorithm. In a recently proposed method [14], it is shown that robustness against illumination changes as well as low computational complexity can be achieved by constructing $\mathbf{Q}$ from the Fourier transforms of a bank of Gabor filters [42].

A two dimensional Gabor filter $g_{\mu,\nu}(z)$ where $z = (x, y)$ is defined as the product of an elliptical Gaussian envelope and a complex plane wave [42]

$$g_{\mu,\nu}(z) = \frac{||k_{\mu,\nu}||^2}{\sigma^2} e^{-\frac{||k_{\mu,\nu}||^2 ||z||^2}{2\sigma^2}} \left[e^{ik_{\mu,\nu}.z} - e^{-\frac{\sigma^2}{2}} \right] \tag{2.5}$$

where ν and μ denote the scale and orientation of the Gabor filter, respectively. σ is the parameter determining the ratio of the Gaussian window width to the wavelength. The wave vector $k_{\mu,\nu}$ is defined as

$$k_{\mu,\nu} = k_\nu e^{i\phi_\mu} \tag{2.6}$$

where $k_\nu = \frac{k_{max}}{f^\nu}$ and $\phi_\mu = \frac{\pi\mu}{8}$. f is the spacing factor between kernels in the frequency domain and k_{max} is the maximum frequency. The term $e^{-\frac{\sigma^2}{2}}$ is subtracted in order to make the filter invariant to illumination changes. In this dissertation, a bank of 40 Gabor filters corresponding to five different scales, $\nu = 0, \ldots, 4$, and eight orientations, $\mu = 0, \ldots, 7$, was used in the experiments. The values of other parameters were set as follow: $\sigma = 2\pi$, $k_{max} = \frac{\pi}{2}$ and $f = \sqrt{2}$ [42].

Assume that $\mathbf{g}_k$ $(k = 1, \ldots, K)$ is the k-th impulse response of a bank of K Gabor filters, the alignment error can be written as the sum of squared differences (SSD) across all filter responses of the warped probe patch and the gallery patch

$$E_i(\mathbf{p}_i) = ||\{\mathbf{g}_k * I_i(\mathbf{p}_i)\}_{k=1}^K - \{\mathbf{g}_k * T_i(\mathbf{0})\}_{k=1}^K||^2 \tag{2.7}$$

where $\{.\}_{k=1}^{K}$ denotes the concatenation operation, i.e. $\{\mathbf{x}_k\}_{k=1}^{K} = [\mathbf{x}_1^T, \ldots, \mathbf{x}_K^T]^T$, and $*$ represents the 2D convolution operation. Using Parseval's relation [43], the error in (2.7) can be estimated in the Fourier domain as

$$E_i(\mathbf{p}_i) = ||\hat{I}_i(\mathbf{p}_i) - \hat{T}_i(\mathbf{0})||^2_{\mathbf{S}} \tag{2.8}$$

where $\mathbf{S} = \sum_{k=1}^{K}(\text{diag}(\hat{\mathbf{g}}_k))^T\text{diag}(\hat{\mathbf{g}}_k)$ and $\hat{I}_i$, $\hat{T}_i$, $\hat{\mathbf{g}}_k$ are the 2D Fourier transforms of I_i, T_i, $\mathbf{g}_k$, respectively. It is worth noting that $\mathbf{S}$ is a diagonal matrix and can be precomputed. As the 2D Fourier transform of a signal of length L is computed by pre-multiplying it by the $L \times L$ Fourier matrix $\mathbf{F}$, (2.8) is equivalent to

$$E_i(\mathbf{p}_i) = ||I_i(\mathbf{p}_i) - T_i(\mathbf{0})||^2_{\mathbf{F}^T\mathbf{S}\mathbf{F}} \quad . \tag{2.9}$$

From (2.3), the update $\Delta\mathbf{p}_i$ is obtained as

$$\Delta\mathbf{p}_i = \mathbf{H}_{flk}^{-1}(\mathbf{F}\mathbf{J}_i)^T\mathbf{S}\mathbf{F}[T_i(\mathbf{0}) - I_i(\mathbf{p}_i)] \tag{2.10}$$

where $\mathbf{H}_{flk} = (\mathbf{F}\mathbf{J}_i)^T\mathbf{S}(\mathbf{F}\mathbf{J}_i)$ is the pseudo-Hessian. In order to perform the update efficiently, the FFT algorithm [43] is applied to estimate the Fourier transforms of the columns of the Jacobian matrix $\mathbf{J}$ and the error image $T_i(\mathbf{0}) - I_i(\mathbf{p}_i)$ at each iteration.

The above formulation of the LK algorithm is known as the forward additive (FA) algorithm. In order to improve the computational efficiency, an extension to the forward additive LK called the inverse compositional (IC) algorithm was proposed in [44]. In this approach, the error function is formulated by linearizing $T_i(\Delta\mathbf{p})$ rather than $I_i(\mathbf{p}_i + \Delta\mathbf{p}_i)$

$$E \approx ||T_i(\mathbf{0}) + \mathbf{J}_{i(ic)}\Delta\mathbf{p}_i - I_i(\mathbf{p}_i)|| \tag{2.11}$$

where $\mathbf{J}_{i(ic)} = \left(\frac{\partial T_i(\mathbf{0})}{\partial \mathbf{p}_i}\right)^T$. The update $\Delta\mathbf{p}_i$ is can be solved as [14]

$$\Delta\mathbf{p}_i = \mathbf{B}[I_i(\mathbf{p}_i) - T_i(\mathbf{0})] \tag{2.12}$$

where $\mathbf{B} = \mathbf{H}^{-1}_{flk(ic)}(\mathbf{F}\mathbf{J}_{i(ic)})^T\mathbf{S}\mathbf{F}$. It is worth noting that the pseudo-Hessian $\mathbf{H}_{flk(ic)} = \mathbf{J}^T_{i(ic)}\mathbf{F}^T\mathbf{S}\mathbf{F}\mathbf{J}_{i(ic)}$ is computed only once for all iterations.

If N and n are the number of pixels in an image patch and the number of warp parameters, respectively, the computational complexity of the inverse compositional algorithm is $O(n^2 + nN)$ per iteration [14]. This is significantly better than the case of the forward additive approach where the computational complexity is $O(n^3 + n^2N + nN\log N)$ per iteration.

2.4 Frontal Face Reconstruction using Markov Random Fields

Given an input image I of a non-frontal face and M training face images $T^{(k)}$, $k = 1, \ldots, M$ captured at the frontal pose, all of them are divided into the same regular grid of N overlapping patches of size $w \times h$. A set of M possible local warps $\mathcal{P}_i = \{\mathbf{p}_i^{(k)} : k = 1, \ldots, M\}$ can be estimated for each patch I_i, by aligning it with the corresponding patches of the training images using the method presented in Section 2.3.2. By aligning the patches in the non-frontal views with the ones in the frontal views, we can obtain the information about how the local patches are transformed as a result of the 3D rotation of the face. The goal of our algorithm is to find a globally optimal set of warps for all the patches in the input image such that we can predict the input face at the frontal pose by transforming these patches using the obtained warps. This problem can be turned into a discrete labeling problem with

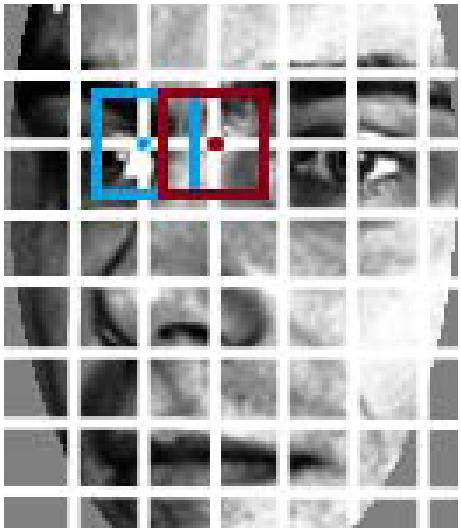


Figure 2.1: Two neighboring MRF nodes with overlapping patches.

a well defined objective function using a discrete MRF. Note that in our approach, the training database need not contain the frontal images of the person in the input image I.

2.4.1 Markov Random Fields

In the proposed algorithm, lattice points whose local patches are inside the image form a set of MRF nodes $\mathcal{V}$ (Figure 2.1). The set of warps $\mathcal{P}_i$ can be considered as the set of possible labels for node i. A 4-connected neighborhood system is then created by edges $\mathcal{E}$ of the MRF.

The *single node potential* $E_i(\mathbf{p}_i)$ penalizes the cost of assigning the warp $\mathbf{p}_i^{(k)}$ to node i. It can be defined using (2.9) as

$$E_i(\mathbf{p}_i) = ||I_i(\mathbf{p}_i) - T_i^{(k)}(\mathbf{0})||^2_{\mathbf{F}^T\mathbf{SF}} \tag{2.13}$$

where $\mathbf{p}_i \in \mathcal{P}_i$ and k is the index of the training image that corresponds to the warp $\mathbf{p}_i$. The *pairwise potential* $E_{ij}(\mathbf{p}_i, \mathbf{p}_j)$ is the cost of label discrepancy between two

neighboring nodes i and j. In other words, this smoothness term measures how well neighboring labels agree at the region of overlap. In order to reduce the effect of illumination changes, the local patches are normalized by subtracting the means and dividing by the standard deviations before estimating the sum of squared difference in the overlapping region. $E_{ij}(\mathbf{p}_i, \mathbf{p}_j)$ can be written as

$$E_{ij}(\mathbf{p}_i, \mathbf{p}_j) = \sum_{\mathbf{x} \in \text{node } i \cap \text{node } j} \left(\hat{I}_i(\mathbf{x}; \mathbf{p}_i) - \hat{I}_j(\mathbf{x}; \mathbf{p}_j) \right)^2 \tag{2.14}$$

where $\mathbf{p}_i \in \mathcal{P}_i$, $\mathbf{p}_j \in \mathcal{P}_j$ and $\hat{I}_i(\mathbf{x}; \mathbf{p}_i)$ denotes the intensity value at the location $\mathbf{x}$ in $\hat{I}_i(\mathbf{p}_i)$. The intensity at $\mathbf{x}$ of the normalized patch $\hat{I}_i(\mathbf{p}_i)$ is obtained as

$$\hat{I}_i(\mathbf{x}; \mathbf{p}_i) = \frac{I_i(\mathbf{x}; \mathbf{p}_i) - \mu_i}{\sigma_i} \tag{2.15}$$

where μ_i and σ_i are the mean and standard deviation of the intensities, respectively, of the local patch I_i without applying any warping function. As local deformations do not affect the intensities of image pixels, the values of μ_i and σ_i can be precomputed to improve the speed of the algorithm. The optimal labeling or the optimal set of warps $\{\hat{\mathbf{p}}_i\}_{i=1}^M$ can be found by minimizing the following energy function

$$E(\{\mathbf{p}_i\}_{i=1}^M) = \sum_{i \in \mathcal{V}} E_i(\mathbf{p}_i) + \lambda \sum_{(i,j) \in \mathcal{E}} E_{ij}(\mathbf{p}_i, \mathbf{p}_j) \tag{2.16}$$

where λ is a regularization parameter that controls the interaction between the single node potentials and pairwise potentials.

2.4.2 Priority Belief Propagation and Label Pruning

The minimization of (2.16) can be performed by using an optimization method for MRFs known as Belief Propagation (BP) [45]. It is an inference technique that

works by passing local messages along the nodes of a MRF. In the case of Markov networks without loops, BP is an exact inference method. Even in networks with loops, it often leads to good approximate results [46]. Using negative logarithmic probabilities, a message from node i to node j at time t is defined as

$$m_{ij}^{t}(\mathbf{p}_j) = \min_{\mathbf{p}_i \in \mathcal{P}_i} \{E_i(\mathbf{p}_i) + \lambda E_{ij}(\mathbf{p}_i, \mathbf{p}_j) + \sum_{k:k\neq j,(k,i)\in\mathcal{E}} m_{ki}^{t-1}(\mathbf{p}_i)\} \quad . \tag{2.17}$$

Assume that all messages converge after s iterations, the belief of node i for $\mathbf{p}_i \in \mathcal{P}_i$, $b_i(\mathbf{p}_i)$ is computed as

$$b_i(\mathbf{p}_i) = -E_i(\mathbf{p}_i) - \sum_{k:(k,i)\in\mathcal{E}} m_{ki}^{s}(\mathbf{p}_i) \quad . \tag{2.18}$$

The warp $\hat{\mathbf{p}}_i = \underset{\mathbf{p}_i \in \mathcal{P}_i}{\operatorname{argmax}}\, b_i(\mathbf{p}_i)$ is selected as the optimal label for node i.

It is known that the standard BP is slow and requires many iterations to converge [47]. In [15], two extensions to the standard BP were proposed in order to improve the speed and make the algorithm converge after a small number of iterations.

The first extension to the standard BP is the use of *dynamic label pruning*. If the number of active labels for a node is greater than L_{max}, a user specified constant, label pruning will be applied to the node. The labels of a visited node are traversed in the descending order of relative belief $b_i^{rel}(\mathbf{p}_i)$, where the relative belief is defined as $b_i^{rel}(\mathbf{p}_i) = b_i(\mathbf{p}_i) - b_i^{max}$ and b_i^{max} is the maximum belief of node i. Those labels $\mathbf{p}_i \in \mathcal{P}_i$ with $b_i^{rel}(\mathbf{p}_i) > b_{prune}$ are selected as active labels for node i. b_{prune} is the label pruning threshold belief. Furthermore, a label is declared as active only if it is not too similar to any of the already active labels in order to avoid choosing many similar labels and wasting a large part of the active label set. Two labels are

considered similar if their normalized cross correlation is greater than a threshold $T_{similar}$. Note that a minimum number of labels L_{min} is always kept for each node. The complexity of updating the messages is reduced from $O(|L|^2)$ to $O(|L_{max}|^2)$ by applying label pruning to BP [15]. In addition, the speed of BP can also be improved by precomputing the reduced matrices of pairwise potentials.

The second improvement is the use of *message scheduling* to determine the transmitting order for a node based on the confidence of that node about its labels. *The node most confident about its label should be the first one to transmit outgoing messages to its neighbors* [15].The priority of a node is defined as $priority(i) = \frac{1}{|Q_i|}$ where $|Q_i|$ is the cardinality of the set $Q_i = \{\mathbf{p}_i \in \mathcal{P}_i : b_i^{rel}(\mathbf{p}_i) \geq b_{conf}\}$. b_{conf} is the confidence threshold belief. By employing this message scheduling in BP, the node that has the most informative messages will transmit first in order to increase the confidence of its neighbors. This helps the algorithm to converge only after only a small, fixed number of iterations. Furthermore, message scheduling also makes the neighbors of the transmitting node more tolerant to label pruning.

2.5 Experimental Results

2.5.1 Frontal-View Classification using Dense SIFT Descriptors

In order to avoid degrading performance when applying the proposed pose compensation technique to face recognition, it is important to be able to automatically decide if the input face image is frontal or non-frontal. In our approach, the frontal-view classification is performed using a modified version of the algorithm

presented in Chapter 5. First, dense SIFT descriptors are extracted from image grid points in order to obtain a representation that is robust to noise and illumination variations. The dimension of the concatenated descriptor vector is reduced for efficient processing by using the Random Projection (RP). Finally, an SVM is employed to decide whether the face image is at the frontal pose or not.

The proposed frontal-view classification algorithm was trained using an SVM on 2D images generated from the 3D faces in the USF 3D database [25]. By rotating the 3D models and projecting them onto the image plane, we can synthesize the 2D face images at different viewing angles. Face images with less than $\pm 5^{\circ}$ in both the yaw and pitch angles are labeled as frontal. Figure 2.2 shows the 2D face images of a person in the database generated at different poses and the visualization of their corresponding dense SIFT descriptors.

The proposed frontal-view classification algorithm was tested on four different databases including the USF 3D database [25], FERET [1], CMU-PIE [2] and Multi-PIE [3]. For the USF 3D database, the synthesized face images were divided into five subsets. Four of them were used for training and the remaining subset was used for testing. It takes less than 4 seconds to perform the frontal-view classification for an input face image of size 130×150 on an Intel Xeon 2.13 GHz desktop.

Table 2.1 shows the classification rates for the four datasets. The results obtained using dense SIFT descriptors with PCA are also included for a comparison. It can be seen from the table that, although the classification rates are high for both approaches, the one using dense SIFT and RP achieves better results.

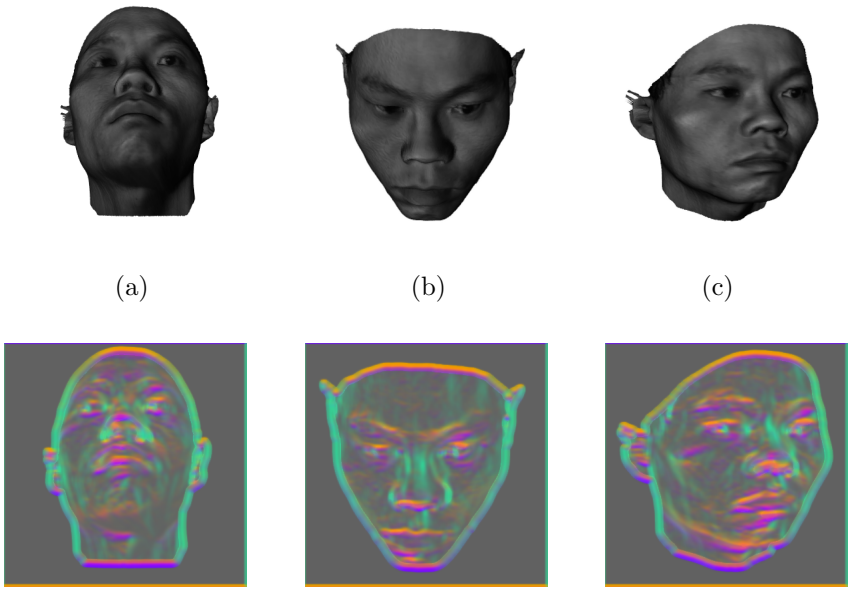


Figure 2.2: First row: the 3D face model of a person in the USF 3D database at different viewing angles. Second row: visualization of the corresponding dense SIFT descriptors.

2.5.2 Frontal Face Reconstruction

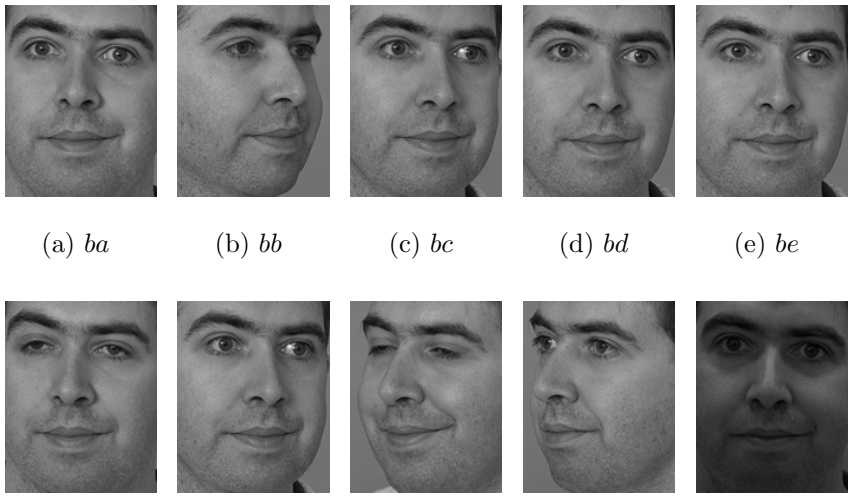
In this section, we present the results of reconstructing frontal views from non-frontal face images using the proposed approach. Given an input face image, it is roughly aligned to the frontal faces in the training database using the coordinates of the two eyes. The input face and eye locations are detected automatically using the Viola-Jones object detection framework [8]. Similar to [27], different cascade classifiers are trained to locate the faces and eyes for the three rough pose classes (left half-profile, frontal and right half-profile). Each classifier can also handle pitch angles ranging from $-30°$ to $30°$. Positive training samples were cropped from the

Table 2.1: Frontal-view classification rates for different datasets.

Method	USF 3D	FERET	CMU-PIE	Multi-PIE
Dense SIFT + PCA	96.6%	95.7%	94.7%	94.3%
Our approach (Dense SIFT + RP)	**98.3**%	**97.2**%	**96.9**%	**94.9**%

annotated face images of the first two hundred subjects of the Multi-PIE dataset [3] as well as from other datasets such as the USF 3D database [25], Pointing '04 [48], FacePix(30) [49] and LFW [50]. Negative samples were collected from a large number of random images on the Web. The input face image is translated, rotated and scaled so that the eyes map to canonical eye positions. Although this initial alignment process results in the difference in scale between the frontal and non-frontal faces, the local warps of image patches are able to compensate for this variation, given the pose of the non-frontal face is not too severe. Both the input and training images are smoothed by a 2D Gaussian filter in order to remove the noise as well as improve the accuracy of the estimation of image gradients in the alignment step.

The first dataset used in the experiments is the FERET dataset [1] that consists of images from two hundred subjects. Each subject in this database was captured at nine different view-points *ba*, *bb*, *bc*, *bd*, *be*, *bf*, *bg*, *bh*, *bi* which roughly correspond to nine viewing angles of 0°, 60°, 40°, 25°, 15°, −15°, −25°, −40°, −60°, respectively. The database also contains images denoted as *bk* which are frontal images corresponding to *ba*, but taken under different lighting. Figure 2.3 shows different face images of a subject in the FERET database with varying pose and illumination.



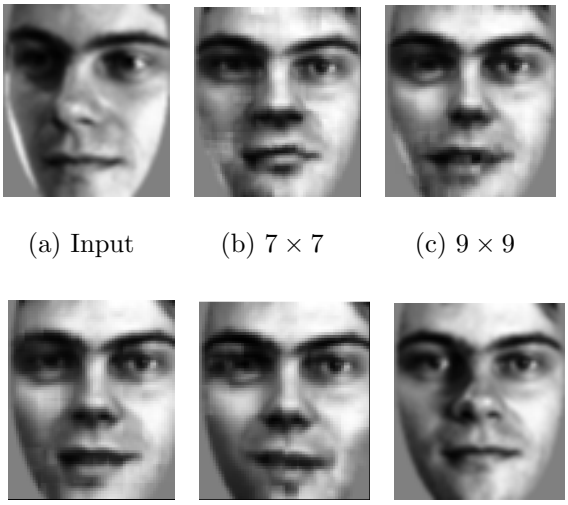
(a) *ba* (b) *bb* (c) *bc* (d) *bd* (e) *be*

(f) *bf* (g) *bg* (h) *bh* (i) *bi* (j) *bk*

Figure 2.3: Face images of a subject in the FERET database with varying viewpoints and illumination.

One of the most important parameters in our method is the patch size. It should not be either too large or too small. If the patches are too small, they do not contain sufficient information for estimating the alignment parameters, especially when there are large displacements. A good patch size must provide enough overlapping in order to align corresponding patches between different views. On the other hand, if the patch size is too big, alignment parameters may not be estimated accurately [51] and blocking effects also appear. Figure 2.4 shows the reconstructed frontal faces from a non-frontal face image using different patch sizes. It can be seen from the figure that if the patch size is too small or too large, there are many artifacts in the outputs. The patch size of 15×15 (Figure 2.4d) gave the best virtual frontal view when compared to the ground truth in Figure 2.4f.

In all the experiments reported in this chapter, the patch size was set at 15×15



(a) Input (b) 7×7 (c) 9×9

(d) 15×15 (e) 21×21 (f) Ground-truth

Figure 2.4: Reconstructed frontal faces with various patch sizes.

and the gap between two neighboring MRF nodes was selected at ten pixels in order to have a sufficient amount of overlap between the neighboring patches. $\lambda = 1$ was chosen to be the value of the regularization parameter. Two hundred frontal images denoted as *ba* from the FERET database were taken as the training set for guiding the alignment process. We observed that the number of iterations required for the priority BP algorithm with label pruning to converge is around five iterations. It takes less than two minutes to synthesize the frontal view for an input face image of size 130×150 on an Intel Xeon 2.13 GHz desktop.

In order to evaluate the performance of our approach in the case of varying illumination, another training set was formed from two hundred frontal face images of the FERET database taken under different lighting (those denoted as *bk*). The reconstructed frontal faces using two training sets *ba* and *bk* are shown in Figure

2.5. It can be seen from the figure that the difference in illumination between the input image and the training set does not affect the robustness of our algorithm. Both results obtained using *ba* and *bk* look very similar to each other and are close to the ground truths (Figures 2.5m and 2.5n).

The proposed algorithm was also tested on the CMU PIE database [2]. This database consists of face images taken from sixty eight subjects under thirteen different poses. The poses are denoted as $c05$ and $c29$ (the yawn angle about $\pm 22.5°$), $c37$ and $c11$ (the yawn angle about $\pm 45°$), and $c07$ and $c09$ (the pitch angle about $\pm 20°$). Figure 2.6 shows the synthesized frontal views of the same subject from the CMU PIE database at different poses. It can be seen from the figure that the proposed approach was able to reconstruct the frontal views very well regardless of the viewing angles.

In order to evaluate the range of poses that can be handled by the method, we synthesized the frontal views of 2D face images generated from the USF 3D models at various viewing angles. The proposed approach can handle up to $\pm 30°$ in the pitch angle and $\pm 45°$ in the yaw angle. Figure 2.7 shows the synthesized frontal views for face images of the same person at four different poses. It can be seen from Figure 2.7h that the algorithm failed to reconstruct the frontal face image at the extreme pose. This is because most of the information on one half of the face is occluded due to the viewing angle. Another reason is that the extreme pose results in large image transformations that can not be handled by local warps of image patches.

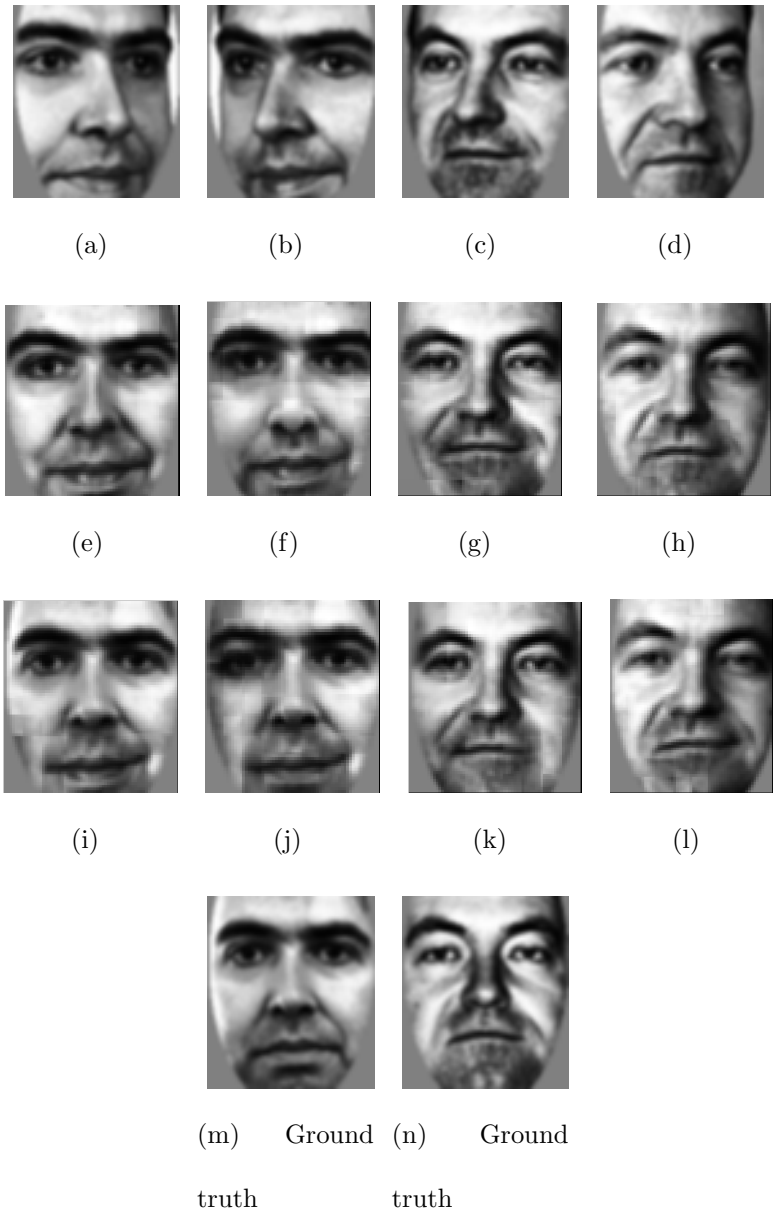


Figure 2.5: Reconstructed frontal faces using training sets under different lighting. First row: input images, second row: results obtained using *ba* training set, third row: results obtained using *bk* training set, last row: ground truths.

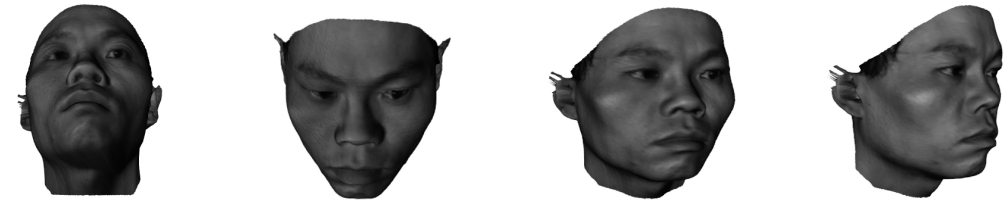
(a) $c05$ (b) $c07$ (c) $c29$ (d) $c37$

(e) $c05$ (f) $c07$ (g) $c29$ (h) $c37$

Figure 2.6: Some examples of reconstructed frontal faces of the same subject from the CMU PIE database. First row: input images, second row: reconstructed frontal views.

2.5.3 Pose Invariant Face Recognition

As presented in the above sections, it is more computationally efficient to classify whether a face image is frontal than to synthesize its frontal view (four seconds compared to two minutes). Thus, the frontal-view classifier is an important component of the proposed pose-invariant face recognition system. Before performing the recognition, the probe image was fed to the frontal-view classifier. If the image was classified as non-frontal, it was transformed to the frontal view using the proposed algorithm. As a result, it is possible to perform recognition by combining our algorithm and any frontal face recognition technique. As we do not require the reference set to include an example of the person in the test image, the same two



(a) Pitch: +30° (b) Pitch: −30° (c) Yaw: −45° (d) Yaw: −55°

(e) Pitch: +30° (f) Pitch: −30° (g) Yaw: −45° (h) Yaw: −55°

Figure 2.7: Reconstructed frontal faces for input images at different poses from the USF 3D database. First row: the 3D face model of a person in the USF 3D database at different viewing angles. Second row: 2D frontal images synthesized using the proposed method.

hundred *ba* frontal images from the FERET database were used as the training set for synthesizing frontal views in all the three face recognition experiments.

As in [27], if the face and both eyes cannot be detected using the cascade classifiers, a Failure to Acquire (FTA) has occurred. In this case, the frontal reconstruction is not carried out and the test image is not counted as a recognition error. The FTA rate is reported for each dataset in the recognition experiments below.

In our experiments, the Local Gabor Binary Pattern (LGBP) [52] was selected as the face recognizer due to its effectiveness. In this method, a feature vector is formed by concatenating the histograms of all the local Gabor magnitude pattern

Table 2.2: Recognition rates of different approaches on the FERET database [1]. The frontal faces *ba* were used as the gallery images.

Method	*bh* −40°	*bg* −25°	*bf* −15°	*be* +15°	*bd* +25°	*bc* +40°	**Avg.**
LGBP [52]	62.0%	91.0%	98.0%	96.0%	84.0%	51.0%	80.5%
LLR [35]	55.0%	89.5%	93.0%	89.0%	77.0%	53.0%	76.1%
PAN [53]	78.5%	91.5%	98.5%	97.0%	93.0%	81.5%	90.0%
3D Pose Norm. [27]	90.5%	98.0%	98.5%	97.5%	97.0%	91.9%	95.6%
Our approach	**91.0%**	**97.3%**	**98.0%**	**98.5%**	**96.5%**	**91.5%**	**95.5%**

maps over an input image. The histogram intersection is used as the similarity measurement in order to compare two feature vectors. More details about the application of the LGBP algorithm for face recognition can be found in [52].

FERET Database: First, the recognition performance of our method on the FERET database is reported. We also compare our approach with the Local Gabor Binary Pattern (LGBP) [52], the Locally Linear Regression (LLR) method [35], the Piecewise Affine warping No stretch (PAN) approach [53], and a recent method based on 3D pose normalization [27]. The frontal faces *ba* were used as the gallery images. Table 2.2 shows the recognition rates of different methods for two hundred subjects at seven poses ranging from $-40°$ to $+40°$. It can be seen that the proposed approach outperformed the methods proposed in [35, 52, 53]. The average rank-1 recognition rate of our algorithm was 95.5%, comparable to the result presented in [27] (95.6%). The FTA rate for the FERET dataset was 1.36%.

CMU-PIE Database: Next, we present the recognition results on the CMU

PIE database. We compare our results with the ones presented in [31, 35] and [29] for thirty four faces using the same set-up where the gallery pose is frontal ($c27$) and the probe poses are $c05$, $c07$, $c09$, $c11$, $c29$ and $c37$. It can be seen from Table 2.3a that the proposed approach outperformed [35] and [29]. However, it was not as good as the stereo matching method presented in [31] (98.5% compared to 99.5%) which requires four landmark points. The proposed algorithm is also compared with the methods in [52] and [27] using all sixty eight faces in the CMU-PIE database. Table 2.3b shows that our recognition rate (98.8%) is better than the ones obtained by [52] (82.4%) and comparable to [27] (99.0%). For the this dataset, the FTA was 0.84%.

Multi-PIE Database: We also performed face recognition experiments on one hundred and thirty seven subjects (*Subject ID* 201 to 346) with neutral expressions and frontal illumination from the Multi-PIE database [3]. One hundred and thirty seven frontal images from the earliest session (*Pose ID* 051) were used as the gallery images. The probe set included the remaining images of both frontal (from other sessions) and non-frontal views. The comparisons between our approach and the methods proposed in [52] and [27] on the Multi-PIE dataset are shown in Table 2.4. The average recognition rate achieved by our algorithm was better than the ones obtained by using the other two methods (89.4% compared to 64% and 87.7%). The FTA rate was 1.6%.

Table 2.3: Recognition rates of different approaches on the CMU-PIE database [2]. The frontal faces $c27$ were used as the gallery images.

(a) 34 Faces

Method	$c11$ $-45°$	$c29$ $-22.5°$	$c05$ $+22.5°$	$c37$ $+45°$	$c07$ up 22.5°	$c09$ down 22.5°	**Avg.**
ELF (Complex) [29]	78.0%	91.0%	93.0%	89.0%	95.0%	93.0%	89.8%
LLR [35]	89.7%	100.0%	98.5%	82.4%	98.5%	98.5%	94.0%
3ptSMD [31]	97.0%	100.0%	100.0%	100.0%	100.0%	100.0%	99.5%
Our approach	**97.0**%	**100.0**%	**100.0**%	**97.0**%	**97.1**%	**100.0**%	**98.5**%

(b) 68 Faces

Method	$c11$ $-45°$	$c29$ $-22.5°$	$c05$ $+22.5°$	$c37$ $+45°$	$c07$ up 22.5°	$c09$ down 22.5°	**Avg.**
LGBP [52]	71.6%	87.9%	86.4%	75.8%	78.8%	93.9%	82.4%
3D Pose Norm. [27]	98.5%	100.0%	100.0%	97.0%	98.5%	100.0%	99.0%
Our approach	**97.0**%	**100.0**%	**100.0**%	**97.0**%	**98.5**%	**100.0**%	**98.8**%

2.6 Conclusions

In this chapter, a method for synthesizing the virtual frontal view from a non-frontal face image was presented. By dividing the input image into overlapping patches, a globally optimal set of local warps was estimated to transform the patches to the frontal view. Each patch was aligned with images from a training database of frontal faces in order to obtain a set of possible warps for that node. It is worth noting that we do not require the training database to include the frontal

Table 2.4: Recognition rates of different approaches on one hundred and thirty seven subjects (*Subject ID* 201 to 346) with neutral expressions and frontal illumination from the Multi-PIE database [3]. The frontal images from the earliest session (*Pose ID* 051) were used as the gallery images.

Method	080_05 −45°	130_06 −30°	140_06 −15°	051_07 0°	051_08 +15°	041_08 +30°	190_08 +45°	**Avg.**
LGBP [52]	37.7%	62.5%	77.0%	92.6%	83.0%	59.2%	36.1%	64.0%
3D Pose Norm. [27]	74.1%	91.0%	95.7%	96.9%	95.7%	89.5%	74.8%	87.7%
Our approach	**86.3**%	**89.7**%	**91.7**%	**92.5**%	**91.0**%	**89.0**%	**85.7**%	**89.4**%

images of the person in the test image. By using an extension of the LK algorithm that accounts for substantial illumination variations, the alignment parameters were calculated efficiently in the Fourier domain. The set of optimal warps was obtained by formulating the optimization problem as a discrete labeling algorithm using a discrete MRF and an efficient variant of the BP algorithm. The energy function of the MRF was constructed to handle illumination variations between different image patches. Furthermore, based on the sparsity of local SIFT descriptors, an efficient algorithm was also designed to classify whether the pose of the input face image is frontal or non-frontal. Experimental results using the FERET, CMU PIE and Multi-PIE databases validate the effectiveness of the proposed approach.

Chapter 3: Model-Driven Domain Adaptation on Product Manifolds for Unconstrained Face Recognition

3.1 Introduction

Unconstrained face recognition is a very difficult problem due to appearance variations between the probe and gallery images caused by multiple factors such as blur, expression, illumination, pose and resolution. As a result, face classifiers trained with the assumption that the training and testing data are drawn from similar distributions usually have very poor performance, especially when applied to uncontrolled environments. For instance, face recognition algorithms trained on samples from a source domain containing sharp, well-illuminated face images do not often perform well when used on a target domain containing blurred, poorly-illuminated face images [54]. The performance of these algorithms further degrades when only a limited number of images per subject is available due to the cost and other challenges in data acquisition.

While there have been several studies addressing pre-specified facial variations across source and target domains [10], such as the nine points of light study for illumination [55], analyzing domain shifts caused by multiple, unknown factors has

not received much attention. Domain adaptation is a recent paradigm for addressing such transformations in a broader setting, where given labeled data from the source domain and few (resp. no) labeled data from target domain probe images, semi-supervised (resp. unsupervised) approaches have been devised to account for variations in data across domains [16, 56, 57]. Most of these techniques address domain shifts in a statistical sense as models causing variations in data are not known. This limits their application to the particular problem of face recognition where there is a rich literature on models for pose, lighting, blur, expression and aging. As a result, it is important to understand domain shifts with respect to the underlying constraints pertaining to models that generate the observed data. Such an analysis would necessitate the study of geometrical properties of the image space induced by these models.

Many traditional approaches, however, often either ignore the geometric structures of the space or naively treat the space as Euclidean [58]. While non-linear manifold learning algorithms such as ISOMAP [59] or Locally Linear Embedding (LLE) [60] offer alternatives, they require large amounts of training data to estimate the underlying non-linear manifold structure of the data. Such a requirement on data may not always be satisfied in many real-world applications. One possible solution for handling facial variations due to multiple factors is by employing a mathematical framework called multilinear algebra - the algebra of higher-order tensors. As matrices represent linear operators over a vector space, their generalization, tensors, define multilinear operators over a set of vector spaces [61]. While there have been studies using multilinear algebraic framework for face recognition [61, 62],

such approaches ignore the curved geometry of the image space and resort to an Euclidean treatment. Attempts to incorporate non-linear geometrical structures into the tensor computing framework have been reported in [63–65], but they again need large training data.

We present a domain adaptive solution for face recognition using the tensor geometry corresponding to models explaining facial variations, with as few as a single image per subject in the source domain. Instead of finding linear transformations representing the shift across domains as in [16, 17], we propose a model-driven approach to construct a latent domain where multifactor facial variations across the source and target domains can be captured together. One main advantage of such an approach is even if data within the source domain and/or the target domain is heterogeneous, for instance when the domain shift is due to blur and both source and target data contain a mix of sharp and blurred faces, the process of accounting for domain shift remains unaltered unlike other techniques that expect the domains to be more or less homogeneous [16, 17, 57]. Furthermore, the proposed method overcomes the data requirement constraint for modeling domain variations by synthesizing multiple face images under different illumination, blur and 2D alignment from a *single* input image on the source or target domain, and uses them to formulate a multidimensional tensor unlike other methods like [63] that places more stringent data-requirement constraints. The tensor obtained from the set of synthesized images can then be represented on a product manifold by performing Higher-Order Singular Value Decomposition (HOSVD) and mapping each orthogonal factored matrix to a point on a Grassmann manifold. The order of the tensors is the num-

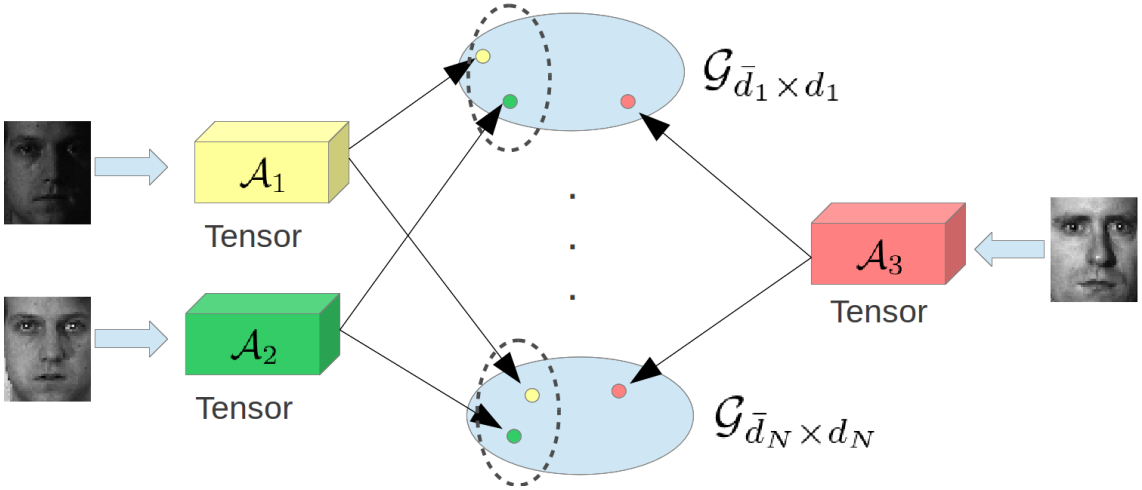


Figure 3.1: An illustration of the approach. Face images from different domains are mapped to a latent domain using the multifactor analysis framework. First, a tensor $\mathcal{A}_i$ is obtained from each face image by synthesizing it under multifactor variations. The tensors are then mapped to a product manifold, the collection of $\mathcal{G}_{\bar{d}_j \times d_j}$'s $(j = 1, \ldots, N)$, that acts as a latent domain. Subsequent computations are performed in the latent domain using geometric and statistical tools with which the identity of target domain faces are inferred. *(This figure is best viewed in color.)*

ber of factors used in the synthesis process. We then recognize the target domain face labels by performing computations pertaining to the tensor geometry for cases where the source domain either contains only one image per subject, or has multiple images per subject. We also address the problem of image set matching which is relevant to video-based face recognition where multiple frames in a video provide evidence related to the facial identity. An illustration of the proposed approach is shown in Figure 3.1.

Contributions:

- We propose a model-driven domain adaptation approach for face recognition with multiple factor variations, using multilinear algebraic principles. Unlike many other methods that require a large training set, the proposed algorithm uses as little as one face image per subject to characterize the underlying geometry of the latent domain as a product of Grassmannian manifolds.

- We then introduce a novel kernel derived from the projection metric on product spaces. When there are sufficient samples available for each subject in the source domain, this projection kernel can be employed to extend any kernelized learning algorithms to product manifolds, which enables us to account for facial variations such as 3D pose and expression that are not explicitly modeled.

- We also present a probabilistic approach for performing image set classification. The classification algorithm is then implemented in the projection space using the Kullback-Leibler divergence as a distance measure.

Organization of the chapter: Section 3.2 discusses related works. The formulation of the proposed approach is given in Section 3.3, along with an introduction to related mathematical details. Details about computations on product manifolds for performing face recognition are presented in Section 3.4. Section 3.5 focuses on the synthesis of face images under multiple factor variations. Experimental results for constrained and unconstrained face recognition on still-images as well as video datasets are provided in Section 3.6. Section 3.7 concludes the chapter.

3.2 Related Work

This section summarizes some previous works on domain adaptation as well as tensor and manifold learning that are relevant to the proposed method. More comprehensive surveys on general face recognition as well as the use of matrix manifolds in computer vision are available from [10] and [58], respectively.

With face recognition making a gradual transition from constrained acquisition scenarios that were prevalent until early 2000's, to the more recent unconstrained real-world settings, we are faced with the challenging problem of accounting for multiple facial variations across the source domain training data and the target domain testing data. Domain adaptation is one promising methodology for addressing such issues. While first investigated by the natural language processing community [66], adaptation in the context of visual object recognition has been receiving attention over the last three years. For instance, Saenko *et al.* [16] proposed a semi-supervised approach that leverages partially labeled data from the target domain to learn a domain shifting transformation on the labeled source domain data using metric learning. Kulis *et al.* [17] extended this work to handle asymmetric transformation across the source and target domains. Hoffman *et al.* [67] addressed multi-domain adaptation by using a hierarchical clustering type approach to select domains that are most informative to perform recognition. Unsupervised adaptation, where there is no availability of labels from the target domain, was addressed by Gopalan *et. al.* [57] through an incremental approach based on Grassmann manifold interpretation that could handle both single and multi-domain adaptation. Gong *et al.* [68]

extended this approach by proposing an elegant solution to learn incremental information along the manifold by formulating a geodesic flow kernel. Independent of [68], a similar extension was developed by Zheng *et al.* [69]. Subsequently, Shi and Sha [70] proposed an information-theoretical approach for joint learning of domain shift features and classifiers, and Jhuo *et al.* [71] proposed a low-rank, sparsity-driven regularization approach that is robust to noise or outliers. Recent approaches such as [72–74] attempted to find domain shifts by using dictionary learning and sparse coding. While the above-mentioned techniques address the problem of domain adaptation by learning an appropriate domain shifting transformation, another class of techniques advocate a classifier-based approach that directly seeks to learn a target domain classifier from the classifiers trained on source domain(s) [75–77].

These techniques perform adaptation in a statistical sense by minimizing data-dependent mismatch in domain properties. Most facial variations, however, result from changes in image formation mechanisms, and hence it is important to analyze domain shifts for face recognition by taking these imaging models, which often give rise to the notion of manifolds, into account. There have been some attempts in this direction. In order to directly model non-linear image manifolds, many approaches formed a set of synthesized face images from a single face [78–82]. However, in these methods, the synthesized images were simply generated by 2D perturbations [79, 80] or extracting patches from the original image [81, 82]. As a result, the manifolds constructed by these approaches may not capture the variations introduced by multiple factors such as illumination, pose or blur. Although the approach in [81] tried to reduce the effect of illumination by performing photometric normalization on the

image patches, this factor was not modeled explicitly on the image manifold.

To capture the variations created by multiple factors, the multilinear algebraic framework was introduced into the field of computer vision by Vasilescu and Terzopoulos [61]. In that paper, they proposed an extension of Principal Component Analysis (PCA) [83] called Multilinear PCA (MPCA) or Tensorfaces in order to handle multiple factor variations in face recognition. A kernel extension of the MPCA framework was developed in [84]. However, this approach ignored the curved geometry of the image space as it estimated the distance metric in the Euclidean space.

In order to incorporate the *geometrical* structures of the image space into the multilinear algebraic framework, Lui *et. al.* [63] characterized actions as tensors and mapped them to points on a product manifold for action classification from videos. In [85], Park and Savvides combined MPCA with ISOMAP [59] to preserve the local neighborhood structures. The drawback of this approach is that it required a dense sampling of the training dataset to construct the manifold. To avoid this drawback, the same authors proposed to use a Grassmannian instead of ISOMAP as the manifold representation [64]. However, as only a single Grassmann manifold was employed to model the non-linear structures, this may not capture the complex variations created by multiple factors. Another work by Park and Savvides [65] decomposed the manifold in the data space into factor-dependent sub-manifolds. However, this approach, together with [63] and [64], required multiple images for the manifold learning which may not be practical in the case of limited training samples. As a more systematic alternative, our method formulates a product manifold as a

latent domain by analytically characterizing *multiple* factor variations with as few as *one* face image.

3.3 Problem Formulation

Given an input face image $\mathbf{I}$ of a subject, we analytically characterize domain shifts due to changes in illumination, blur and 2D perturbations of face images in different domains. First, the face is illuminated using the albedo estimated by applying the method of [6] and the universal configuration of lighting directions presented in [55]. The span of these relighted images approximates the subspace of illumination variation for this subject. Each relighted image is then blurred by convolving with a complete set of orthonormal basis functions in order to obtain a blur-invariant representation [86]. The relighted and blurred images are further perturbed by applying 2D similarity transformations in order to characterize the registration manifold. The set of synthesized images obtained after the last step are represented by a 4th-order tensor $\mathcal{A} \in \mathcal{R}^{d_1 \times d_2 \times d_3 \times d_4}$, where d_1 is the number of pixels in the face, d_2 is the number of light sources used for relighting, d_3 is the number of orthonormal basis vectors used to get the blur-invariant representation, and d_4 is the number of 2D similarity transformations. As a result of applying HOSVD on the tensor, we obtain a set of orthogonal matrices, where each matrix represents a variation factor and can be handled as a linear term. The tensor is then mapped to a point on a product of Grassmann manifolds using these orthogonal matrices. This product manifold acts as a latent domain for comparing projected data points from

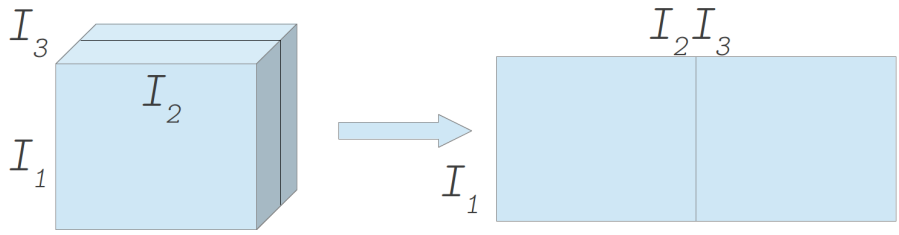


Figure 3.2: Mode-1 flattening of a 3rd-order tensor.

different domains. For instance, if only one sample is available per subject in the source domain, recognition of target domain faces is performed using the geodesic distance on the product manifold. In the case when there are multiple source domain samples per subject, a novel kernel on product spaces is proposed. This kernel can be used with any kernelized learning techniques in order to capture other variations such as 3D pose or expression that are not explicitly modeled.

Next, we will provide a brief review of the background mathematics used in the chapter regarding tensors and how to represent tensors on product manifolds.

3.3.1 Tensors and Tensor Decomposition

Tensors are the natural generalization of matrices to multidimensional spaces. Let $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times \ldots \times d_N}$ be an N-order tensor, an element of $\mathcal{A}$ is denoted as $\mathcal{A}_{i_1 \ldots i_n \ldots i_N}$. The mode-$n$ *flattening* (or *unfolding*) of $\mathcal{A}$ maps the tensor to a 2D matrix $\mathbf{A}_{(n)} \in \mathbb{R}^{d_n \times \bar{d}_n}$ where $\bar{d}_n = d_1 \times \ldots \times d_{n-1} \times d_{n+1} \times \ldots \times d_N$. Each column vector of $\mathbf{A}_{(n)}$ is obtained by varying the n-th index i_n of $\mathcal{A}$ while keeping the other indices fixed. An example of mode-1 flattening of a 3rd-order tensor is shown in Figure 3.2.

Another important operation on tensors that is worth mentioning is the mode-

n product of a tensor $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times \ldots \times d_n \times \ldots \times d_N}$ and a 2D matrix $\mathbf{M} \in \mathbb{R}^{l_n \times d_n}$. The product, denoted by $\mathcal{A} \times_n \mathbf{M}$, returns a tensor $\mathcal{B} \in \mathbb{R}^{d_1 \times d_2 \times \ldots \times d_{n-1} \times l_n \times d_{n+1} \times \ldots \times d_N}$ which can be computed in terms of flattened matrices as

$$\mathbf{B}_{(n)} = \mathbf{M}\mathbf{A}_{(n)}. \tag{3.1}$$

Similar to Singular Value Decomposition (SVD) for matrices, a tensor can be factorized using an extension of SVD, called HOSVD [63], as

$$\mathcal{A} = \mathcal{Z} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times \ldots \times_N \mathbf{U}_N, \tag{3.2}$$

where $\mathcal{Z} \in \mathbb{R}^{d_1 \times d_2 \times \ldots \times d_N}$ is the core tensor, $\mathbf{U}_n \in \mathbb{R}^{d_n \times d_n}$, $1 \leq n \leq N$, are the mode-n orthogonal matrices spanning the column space of $\mathbf{A}_{(n)}$. $\mathbf{U}_n$ can be obtained by performing SVD on $\mathbf{A}_{(n)}$

$$\mathbf{A}_{(n)} = \mathbf{U}_n \Sigma_n \mathbf{V}_n^\top, \tag{3.3}$$

where $\Sigma_n \in \mathbb{R}^{d_n \times \bar{d}_n}$ is a rectangular diagonal matrix of singular values of $\mathcal{A}_{(n)}$, and $\mathbf{V}_n \in \mathbb{R}^{\bar{d}_n \times \bar{d}_n}$ is an orthogonal matrix spanning the row space of $\mathcal{A}_{(n)}$.

The core tensor $\mathcal{Z}$ captures the interaction between the mode matrices $\mathbf{U}_1, \ldots, \mathbf{U}_N$. It is analogous to the diagonal singular value matrix in conventional SVD. However, it is worth noting that $\mathcal{Z}$ does not have the diagonal structure [61].

3.3.2 Grassmann Manifolds

Given an n-dimensional real vector space $\mathcal{V}$, the *Grassmann manifold* (or simply *Grassmannian*) $\mathcal{G}_d(\mathcal{V})$ (with $0 \leq d \leq n$) is a set of all d-dimensional linear subspaces of $\mathcal{V}$ [87]. In the special case where $\mathcal{V} = \mathbb{R}^n$, the Grassmannian $\mathcal{G}_d(\mathbb{R}^n)$ is

denoted as $\mathcal{G}_{n,d}$. Each point on $\mathcal{G}_{n,d}$ represents a subspace spanned by the column space of an $n \times d$ orthogonal matrix. Thus, all orthogonal matrices $\mathbf{Y} \in \mathbb{R}^{n \times d}$ spanning the same linear subspace are considered equivalent, i.e.

$$\lfloor \mathbf{Y} \rfloor = \{\mathbf{YR} | \mathbf{R} \in O(d)\}, \tag{3.4}$$

where $O(d) = \{\mathbf{R} \in \mathbb{R}^{d \times d} | \mathbf{R}^\top \mathbf{R} = \mathbf{R}\mathbf{R}^\top = \mathbf{I}_d\}$ is the orthogonal group.

We use the projection metric [87] as the measure of the geodesic distance between two points on a Grassmann manifold. Let $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)^\top$ be the principal angles between the two linear subspaces $\mathcal{Y}_1$ and $\mathcal{Y}_2$, the geodesic distance based on the projection metric is computed as

$$d_c(\mathcal{Y}_1, \mathcal{Y}_2) = ||sin(\boldsymbol{\theta})||_2, \tag{3.5}$$

where $sin(\boldsymbol{\theta})$ is the vector of sines of the principal angles.

If $\mathbf{Y}_1, \mathbf{Y}_2 \in \mathbb{R}^{n \times d}$ are the orthogonal basis of $\mathcal{Y}_1$ and $\mathcal{Y}_2$, respectively, the principal angles between the subspaces can be numerically computed by performing SVD on $\mathbf{Y}_1^\top \mathbf{Y}_2$ [88]. The singular values of this SVD are the cosines of the principal angles.

The projection metric can be understood as the Euclidean distance in $\mathbb{R}^{n \times n}$ by defining an embedding $\Psi_P(\mathcal{G}_{n,d})$ as

$$\Psi_P : \mathcal{G}_{n,d} \to \mathbb{R}^{n \times n}, \quad span(\mathbf{Y}) \mapsto \mathbf{Y}\mathbf{Y}^\top. \tag{3.6}$$

Thus, the corresponding inner product or projection kernel of the space can be obtained as

$$k_P(\mathbf{Y}_1, \mathbf{Y}_2) = tr\left[(\mathbf{Y}_1\mathbf{Y}_1^\top)(\mathbf{Y}_2\mathbf{Y}_2^\top)\right] = ||\mathbf{Y}_1^\top \mathbf{Y}_2||_F^2, \tag{3.7}$$

where tr is the matrix trace operator and $||.||_F$ is the Frobenius norm. As $k_P(\mathbf{Y}_1, \mathbf{Y}_2) = k_P(\mathbf{Y}_1\mathbf{R}_1, \mathbf{Y}_1\mathbf{R}_2)$ for any $\mathbf{R}_1, \mathbf{R}_2 \in O(d)$, this kernel is well defined. The proof that $k_P(\mathbf{Y}_1, \mathbf{Y}_2)$ is positive definite is given in [89].

3.3.3 Representing Tensors on Product Manifolds

As a result of performing SVD on the flattening matrix $\mathbf{A}_{(n)}$, we obtain two orthogonal matrices $\mathbf{U}_n$ and $\mathbf{V}_n$. The reason for not choosing $\mathbf{U}_n$ to represent the geometry of the tensor is that each $\mathbf{U}_n$ is a point on a special orthogonal group $SO(d_n)$. The geodesic distance on $SO(d_n)$ cannot be obtained as a closed form. Furthermore, if points on $SO(d_n)$ are mapped to a Grassmann manifold, the geodesic distance would always be zero [63].

The matrix $\mathbf{V}_n$ in (3.3) spans the row space of $\mathcal{A}_{(n)}$. As it is usually the case that $d_n < \bar{d}_n$, where $\bar{d}_n$ is defined in Section 3.3.1, $\mathbf{V}_n$ can be substituted by an $\bar{d}_n \times d_n$ orthogonal matrix $\tilde{\mathbf{V}}_n$ by selecting the columns of $\mathbf{V}_n$ corresponding to the non-zeros singular values. Hence, the tensor $\mathcal{A}$ can be represented geometrically as a Cartesian product of the mappings of each $\tilde{\mathbf{V}}_n$ to a point on the Grassmann (factor) manifold $\mathcal{G}_{\bar{d}_n, d_n}$. Furthermore, it is known that the Cartesian product $\mathcal{M} = \mathcal{G}_{\bar{d}_1, d_1} \times \ldots \times \mathcal{G}_{\bar{d}_N, d_N}$ is also a smooth manifold with the manifold topology equivalent to the product topology [90]. Thus, the tensor $\mathcal{A}$ can be represented as a point on this product manifold.

3.4 Computations on Product Manifolds

To account for domain shifts in face recognition, we characterize the tensor by synthesizing facial variations due to illumination, blur and 2D alignment from a single face image. While we defer the details on the synthesis process to the next section, here we focus on performing computations on the latent domain, the product of Grassmannians, where tensors corresponding to source domain face images are modeled to infer the identity of tensors derived from target domain faces. More specifically, we first present details on estimating the geodesic distance on product manifolds, which can accommodate cases where the source domain has only one face image per subject. We then derive a positive definite kernel for product manifolds based on an extension of the projection metric to product spaces. With multiple images per subject in the source domain, this kernel can be used in any kernelized learning algorithm to account for domain shifts due to other factors, such as 3D pose and expression, that are not explicitly synthesized in Section 3.5. As an illustration, by extending the kernel linear discriminant analysis (KLDA) on Grassmann manifolds [89] to product spaces using the proposed kernel, we can find projection directions maximizing inter-class variations (such as due to identities) while minimizing intra-class variations (such as due to pose, expression, occlusion, etc.). Finally, we present a probabilistic approach for performing classification of image sets on product spaces, with applications to video-based face recognition.

3.4.1 Geodesics and Projection Kernels on Product Manifolds

The geodesic in the product manifold $\mathcal{M} = \mathcal{G}_{\bar{d}_1,d_1} \times \ldots \times \mathcal{G}_{\bar{d}_N,d_N}$ is the Cartesian product of the geodesics in $\mathcal{G}_{\bar{d}_1,d_1}, \ldots, \mathcal{G}_{\bar{d}_N,d_N}$ [91]. As a result, the geodesic distance based on the projection metric on the product manifold can be estimated as

$$d_c^{\mathcal{M}}(\mathcal{A}^{(1)}, \mathcal{A}^{(2)}) = ||sin(\boldsymbol{\Theta})||_2, \tag{3.8}$$

where $\mathcal{A}^{(1)}$ and $\mathcal{A}^{(2)}$ are N-order tensors, and $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1^\top, \ldots, \boldsymbol{\theta}_N^\top)^\top$ with $\boldsymbol{\theta}_n$ is the vector of principal angles computed on the factor manifold $\mathcal{G}_{\bar{d}_n,d_n}$.

In the case where there are only limited training samples, even with just one sample per subject, the above distance can be used to perform nearest-neighbor classification on the latent domain. As the geodesic distance between two points is the shortest distance on a curved space, it provides a meaningful similarity measure that takes into account the underlying geometry of the latent domain. Next, a positive definite kernel on product manifolds is introduced. When there are sufficient training samples, this kernel function can be used with any kernelized learning technique to statistically account for variations such as 3D pose and expression on the latent domain.

The extension of the embedding in (3.6) to the product of Grassmann manifolds $\mathcal{M}$ can be written as:

$$\begin{aligned}\Psi_P^{\mathcal{M}} : \mathcal{G}_{\bar{d}_1,d_1} \times \ldots \times \mathcal{G}_{\bar{d}_N,d_N} &\to \mathbb{R}^{\bar{d}_1 \times \bar{d}_1} \times \ldots \times \mathbb{R}^{\bar{d}_N \times \bar{d}_N}, \\ (span(\mathbf{Y}_1), \ldots, span(\mathbf{Y}_N)) &\mapsto \left(\mathbf{Y}_1\mathbf{Y}_1^\top, \ldots, \mathbf{Y}_N\mathbf{Y}_N^\top\right)\end{aligned} \tag{3.9}$$

Thus, the projection kernel function on the product manifold can be defined as the

inner product of this product space:

$$k_P^{\mathcal{M}}(\mathcal{C}^{(1)}, \mathcal{C}^{(2)}) = tr\left[\sum_{i=1}^{N}\left(\mathbf{Y}_i^{(1)}\mathbf{Y}_i^{(1)\top})(\mathbf{Y}_i^{(2)}\mathbf{Y}_i^{(2)\top}\right)\right], \tag{3.10}$$

where $\mathcal{C}^{(m)} = \left\{\mathbf{Y}_1^{(m)}, \ldots, \mathbf{Y}_N^{(m)}\right\}$, $\mathbf{Y}_i^{(m)} \in \mathbb{R}^{\bar{d}_i \times d_i}$ with $i = 1, \ldots, N$, and $\left(span(\mathbf{Y}_1^{(m)}), \ldots, span(\mathbf{Y}_N^{(m)})\right) \in \mathcal{M}$ with $m = 1, 2$.

This leads to the following proposition for the projection kernel on product manifolds:

Proposition 3.4.1. *The projection kernel $k_P^{\mathcal{M}}(\mathcal{C}^{(1)}, \mathcal{C}^{(2)})$, defined in* (3.10)*, is a positive definite kernel.*

Proof. For all $\mathbf{Y}_i^{(m)} \in \mathbb{R}^{\bar{d}_i \times d_i}$ with $i = 1, \ldots, N$ and $m = 1, 2$, we have

$$\begin{aligned} k_P^{\mathcal{M}}(\mathcal{C}^{(1)}, \mathcal{C}^{(2)}) &= tr\left[\sum_{i=1}^{N}\left(\mathbf{Y}_i^{(1)}\mathbf{Y}_i^{(1)\top}\right)\left(\mathbf{Y}_i^{(2)}\mathbf{Y}_i^{(2)\top}\right)\right] \\ &= \sum_{i=1}^{N} tr\left[\left(\mathbf{Y}_i^{(1)}\mathbf{Y}_i^{(1)\top}\right)\left(\mathbf{Y}_i^{(2)}\mathbf{Y}_i^{(2)\top}\right)\right] \\ &= \sum_{i=1}^{N} k_P\left(\mathbf{Y}_i^{(1)}, \mathbf{Y}_i^{(2)}\right) \end{aligned}$$

Thus, $k_P^{\mathcal{M}}(\mathcal{C}^{(1)}, \mathcal{C}^{(2)})$ is the sum of the positive definite kernels defined in (3.7) on each factor manifold. Thus, it is a well-defined and positive definite kernel. □

The proposed projection kernel between two tensors, $\mathcal{A}^{(1)}$ and $\mathcal{A}^{(2)}$, can be computed by setting $\mathbf{Y}_i^{(m)} = \tilde{\mathbf{V}}_i^{(m)}$, for $i = 1, \ldots, N$ and $m = 1, 2$, where $\tilde{\mathbf{V}}_i^{(m)}$ are defined as in Section 3.3.1.

Equipped with the above notation of the projection kernel on product manifolds, we can adapt any kernelized algorithms to perform the learning on the latent domain. In this work, we chose the KLDA algorithm on Grassmann manifolds [89]

since its utility for face recognition has been demonstrated before. When there are sufficient training data available, the projection directions computed by using KLDA on product spaces help better separate samples from different people as they maximize inter-class variations due to identities, while minimize intra-class variations due to factors such as pose, expression, occlusion, etc. The extension is straight forward as we only need to replace the kernel function $k_P(\mathbf{Y}_1, \mathbf{Y}_2)$ with $k_P^{\mathcal{M}}(\mathcal{C}^{(1)}, \mathcal{C}^{(2)})$. The detailed implementation of KLDA on Grassmannians can be found in [89].

3.4.2 Image Set Classification on Product Manifolds

In this section, we present a probabilistic approach to perform domain adaptation for the problem of image set classification. Such a setting occurs naturally in video-based face recognition where several frames in a video sequence are representative of the facial identity. Given a set of images of a subject, it can be characterized as a set of points on a latent domain by projecting the points to a product manifold. For a classification problem with C different subjects, these points can be further mapped to vectors on a $(C-1)$-dimensional space obtained by performing KLDA on the latent domain using the projection kernel proposed in Section 3.4.1.

Assume that the distribution of points in the set $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_M | \mathbf{x}_i \in \mathbb{R}^{(C-1)}, i = 1, \ldots, M\}$ can be approximated by a multivariate Gaussian distribution $\boldsymbol{\pi} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the maximum likelihood estimates of $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ can be written as

$$\hat{\boldsymbol{\mu}}_{ML} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{x}_i \quad \text{and} \quad \hat{\boldsymbol{\Sigma}}_{ML} = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T.$$

Given two sets of points $\mathcal{S}^{(1)}$ and $\mathcal{S}^{(2)}$ represented by the distributions $\boldsymbol{\pi}_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\boldsymbol{\pi}_2 \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, respectively, a distance measure between the sets can be estimated by using the Kullback-Leibler (KL) divergence, which can be obtained in closed form as [92, 93]:

$$d_{KL}(\boldsymbol{\pi}_1||\boldsymbol{\pi}_2) = \frac{1}{2}\bigg(tr(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1) - ln\left(\frac{det(\boldsymbol{\Sigma_1})}{det(\boldsymbol{\Sigma_2})}\right) + (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}_2^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) - (C-1)\bigg). \tag{3.11}$$

where $det(\boldsymbol{\Sigma})$ denotes the determinant of $\boldsymbol{\Sigma}$. It is worth noting that the Kullback-Leibler divergence is a positive but non-symmetric measure. As a result, we estimate the KL divergences of the distribution of a probe set from the distributions of all the gallery sets, and select the gallery set that produces the minimum distance as the best match.

3.5 Multifactor Synthesis

Domain shifts caused by variations in factors such as illumination, blur, pose or expression can result in images of the same person having significantly different appearance in different domains. Furthermore, domain shifts can also be caused by localization errors of face detection algorithms when finding the facial bounding boxes and thus, reduce the accuracy of many existing face recognition algorithms. In this section, we discuss how to synthesize faces of the same subject with varying lighting and blur conditions from a single input image. We also present the details of how to characterize the registration manifold [80] using 2D perturbed images in order to account for the in-plane alignment issue. The synthesis process helps

to characterizes domain shifts caused by factors such as illumination, blur and 2D alignment without the need for a large training dataset. Domain shifts due to other factors such as 3D pose and expression, that are not explicitly synthesized, are handled by the kernel learning technique on product manifolds presented in the previous section.

3.5.1 Illumination

By restricting to convex objects with the Lambertian reflectance model, the diffused component of the surface reflection is given by

$$I_{i,j} = \rho_{i,j} \max(\mathbf{n}_{i,j} \cdot \mathbf{s}, 0), \tag{3.12}$$

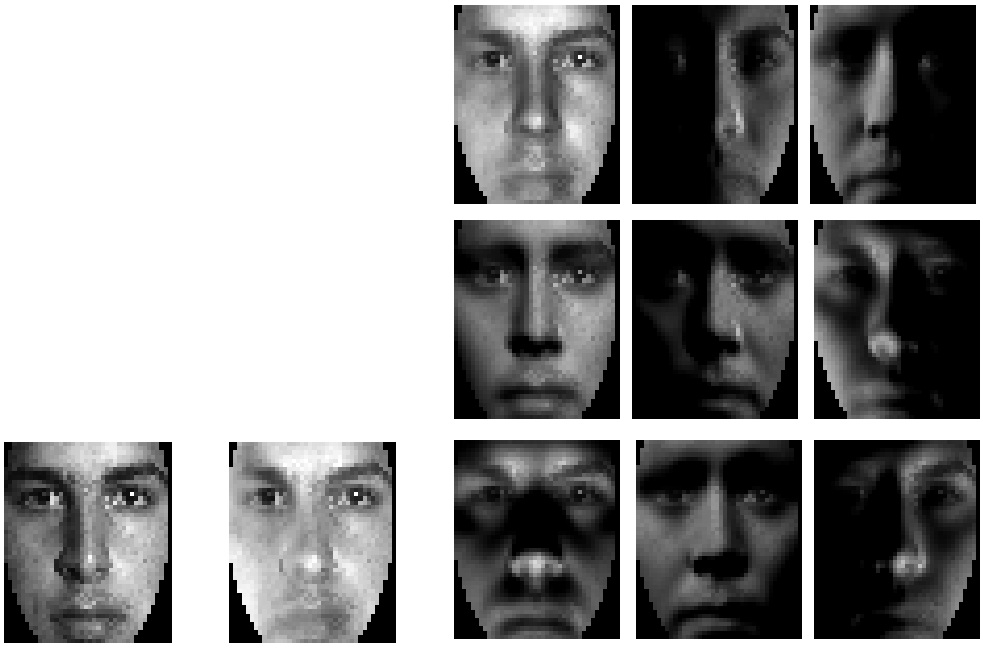
where $I_{i,j}$ is the pixel intensity at position (i, j), $\rho_{i,j}$ and $\mathbf{n}_{i,j}$ are the albedo and surface normal at the corresponding surface point, and $\mathbf{s}$ is the light source direction [6]. From (3.12), the initial estimate of the albedo $\rho_{i,j}^{(0)}$ can be obtained as

$$\rho_{i,j}^{(0)} = \frac{I_{i,j}}{\mathbf{n}_{i,j}^{(0)} \cdot \mathbf{s}^{(\mathbf{0})}}, \tag{3.13}$$

where $\mathbf{n}_{i,j}^{(0)}$ and $\mathbf{s}^{(\mathbf{0})}$ are the initial values of the surface normal and illuminant direction. The values of $\mathbf{n}_{i,j}^{(0)}$ are obtained from an average 3D face in the USF 3D database [25]. The initial lighting direction $\mathbf{s}^{(\mathbf{0})}$ is estimated using the approach presented in [94].

The initial estimate of the albedo $\rho_{i,j}^{(0)}$ can be related to the true albedo $\rho_{i,j}$ as:

$$\begin{aligned} \rho_{i,j}^{(0)} &= \rho_{i,j} \frac{\mathbf{n}_{i,j} \cdot \mathbf{s}}{\mathbf{n}_{i,j}^{(0)} \cdot \mathbf{s}^{(\mathbf{0})}} = \rho_{i,j} + \frac{\mathbf{n}_{i,j} \cdot \mathbf{s} - \mathbf{n}_{i,j}^{(0)} \cdot \mathbf{s}^{(\mathbf{0})}}{\mathbf{n}_{i,j}^{(0)} \cdot \mathbf{s}^{(\mathbf{0})}} \rho_{i,j} \\ &= \rho_{i,j} + w_{i,j}, \end{aligned} \tag{3.14}$$



(a) (b) (c)

Figure 3.3: From left to right : (a) input face image, (b) the albedo estimated using [6], and (c) images of the same person illuminated by using nine different light sources .

where $w_{i,j} = \frac{\mathbf{n}_{i,j} \cdot \mathbf{s} - \mathbf{n}_{i,j}^{(0)} \cdot \mathbf{s}^{(0)}}{\mathbf{n}_{i,j}^{(0)} \cdot \mathbf{s}^{(0)}} \rho_{i,j}$ is the signal-dependent additive noise. $\mathbf{n}_{i,j}$ is the true surface normal at (i, j) and $\mathbf{s}_{i,j}$ is the true lighting direction.

By considering (3.14) as a signal estimation problem where $\rho_{i,j}$ is the original signal and $\rho_{i,j}^{(0)}$ is the noisy observation, the albedo image can be solved by using the Linear Minimum Mean Square Error (LMMSE) method as in [6]. Figures 3.3a and 3.3b shows a face image and its albedo estimated using [6], respectively.

It has been shown that the set of all images of a convex, Lambertian object under different lighting conditions can be approximated by a nine-dimensional linear subspace [95]. This linear subspace can be characterized by illuminating the object

using nine pre-specified light sources given in [55]

$$\begin{aligned} \phi &= \{0, 68, 74, 80, 85, 85, 85, 85, 51\}^\circ \\ \theta &= \{0, -90, 108, 52, -42, -137, 146, -4, 67\}^\circ. \end{aligned}$$

where ϕ and θ denote the azimuth and elevation angles, respectively. Figure 3.3c shows nine images of the same person illuminated by lights from the above configureation. The face image of this person at an arbitrary illumination condition can be written as a linear combination of these nine basis images

$$\mathbf{I} = \sum_{i=1}^{9} \alpha_i \mathbf{I}_i. \tag{3.15}$$

As a result, given a single face image, we can estimate the albedo and relight the face at the nine different light sources using the approach in [6] in order to approximate the subspace of illumination variations of this person.

3.5.2 Blur

The blurring process can be modeled by the image formulation equation as [86]

$$\tilde{\mathbf{I}} = \mathbf{I} * \mathbf{k} + \boldsymbol{\eta}, \tag{3.16}$$

where $*$ denotes the 2D convolution between a clean image $\mathbf{I}_{(n_1 \times n_2)}$ and an unknown blur Point Spread Function (PSF) $\mathbf{k}_{(b_1 \times b_2)}$. $\tilde{\mathbf{I}}_{(n_1 \times n_2)}$ is the blurred image and $\boldsymbol{\eta}_{(n_1 \times n_2)}$ represents the noise introduced by the system (i.e. quantization or other sensor induced errors).

It can be seen that, given $\{\boldsymbol{\phi}_i\}_{i=1}^{K}$ as a complete set of orthonormal basis functions for $\mathbb{R}^{b_1 \times b_2}$ with $K = b_1 \times b_2$, any square-integrable, shift-invariant kernel

$\mathbf{k}_{(b_1 \times b_2)}$ can be written as

$$\mathbf{k} = \sum_{i=1}^{K} \alpha_i \boldsymbol{\phi}_i, \tag{3.17}$$

where $\{\alpha_i\}_{i=1}^{N}$ are the combining coefficients. Without noise, (3.16) can be rewritten as

$$\tilde{\mathbf{I}} = \mathbf{I} * \sum_{i=1}^{K} \alpha_i \boldsymbol{\phi}_i = \sum_{i=1}^{K} \alpha_i (\mathbf{I} * \boldsymbol{\phi}_i). \tag{3.18}$$

Let $\mathbf{D}(\mathbf{I}) = [(\mathbf{I} * \boldsymbol{\phi}_1)^v \quad (\mathbf{I} * \boldsymbol{\phi}_2)^v \ldots (\mathbf{I} * \boldsymbol{\phi}_K)^v]$ be a dictionary of size $d \times K$, where $d = n_1 \times n_2$ with $d > K$, and $(.)^v$ denotes the vectorization operation. The column span of $\mathbf{D}(\mathbf{I})$, i.e. $span(\mathbf{D}(\mathbf{I})) = \{\mathbf{I} * \mathbf{k} | \mathbf{k} \in \mathbb{R}^{b_1 \times b_2}\}$, is a subspace containing the set of convolutions of $\mathbf{I}$ with arbitrary kernels of maximum size $b_1 \times b_2$. Under certain assumptions, the $span(\mathbf{D}(\mathbf{I}))$ allows us to obtain a representation of the image $\mathbf{I}$ that is invariant to blurring with an arbitrary $\mathbf{k}$.

Proposition 3.5.1. *Under three assumptions: (i) there is no noise in the system ($\boldsymbol{\eta} = 0$), (ii) the maximum size of the blur kernel $b_1 \times b_2 = K$ is known, and (iii) the $K \times K$ Block-Toeplitz-Toeplitz-Block (BTTB) matrix corresponding to the unknown blur PSF, under zero boundary conditions for convolution, is full rank, $span(\mathbf{D}(\mathbf{I}))$ is a blur-invariant of $\mathbf{I}$. In other words, $span(\mathbf{D}(\mathbf{I})) = span(\mathbf{D}(\tilde{\mathbf{I}}))$, where $\tilde{\mathbf{I}}$ is a blurred version of $\mathbf{I}$.*

Proof. See the proof of Proposition 2.1 in [86]. □

The main advantage of this representation is that there are no constraints on the shape of the blur kernels that can be handled, as long as the blur kernels satisfy the above assumptions. In the chapter, all the face images are resized to

40×48 and the maximum kernel size is set at $b_1 \times b_2 = 7 \times 7$. As a result, for each relighted face image, a set of $K = 49$ basis vectors is obtained by convolving the image with $\{\boldsymbol{\phi}_i\}_{i=1}^{49}$. In our experiments, the set of basis vectors $\{\boldsymbol{\phi}_i\}_{i=1}^{49}$ is selected as the columns of a 49×49 identity matrix.

3.5.3 2D Registration

In practice, it may be unrealistic to expect that a face detection system can locate faces with many appearance variations with high precision. Thus, the extracted bounding boxes of faces with varying illumination or blur conditions may not align perfectly. In order to account for the alignment errors during the face localization process, a set of perturbed images using 2D similarity transformations is obtained for each face image in order to characterize the registration subspace [80].

A similarity transformation mapping an image coordinate (i, j) to the new coordinate (u, v) can be written in the homogeneous form as

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & t_x \\ \sin(\theta) & \cos(\theta) & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} i \\ j \\ 1 \end{pmatrix} \tag{3.19}$$

where θ and s are the rotation and isometric scaling parameters, respectively. t_x and t_y are the translation parameters. In our experiments, we set the values of θ as $\{-4, -2, 0, 2, 4\}^\circ$, s as $\{0.9, 0.95, 1, 1.05, 1.1\}$, and t_x and t_y as $\{-3, 0, 3\}$ as they provide a reasonable coverage for possible alignment errors between the probe and gallery images. Bilinear interpolation is employed to sample the transformed images. As a result, a total of two hundreds and twenty five perturbed images are

synthesized for each relighted and blurred face image.

3.6 Experiments

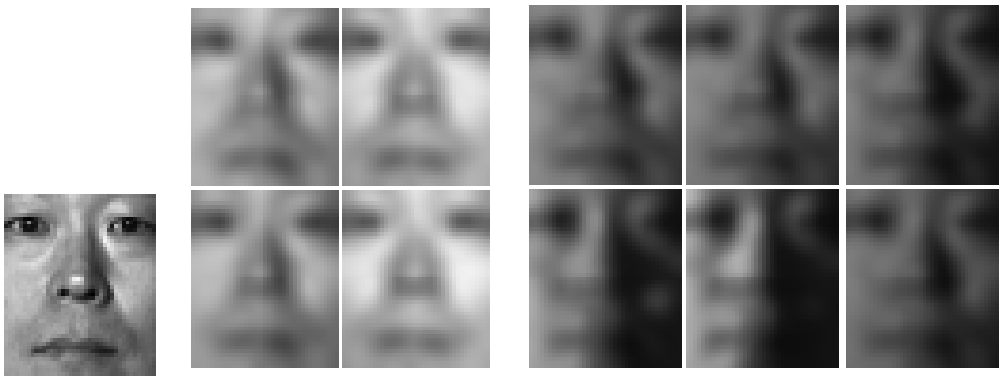
We first test our approach for face identification where the goal is to estimate the subject label of a probe image, and then for face verification, where given a pair of probe images, the goal is determine if they correspond to the same subject or not. For face identification we consider four public datasets namely, the CMU-PIE [2] and AR [7] datasets that contain still faces captured under constrained settings, the UMD remote face dataset [86] comprising of unconstrained still faces, and the Honda/UCSD video dataset [96]. For face verification we use the recent, unconstrained Labeled Faces in the Wild (LFW) dataset [50]. Most of these datasets contain facial variations that are not explicitly synthesized by our method. We compare our approach with several other techniques that were evaluated on these datasets. It is also worth noting that the existing works on domain adaptation such as [16, 17, 57] may not be applicable to these experimental settings. One of the reasons is that they impose data requirement constraints that are not often satisfied as there may be as little as one image available per individual per the source and target domains. Another reason is that the requirement for the source and target domains to be more or less homogeneous may not hold in unconstrained face recognition as domain shifts can be caused by multiple factor variations such as illumination, blur, expression and alignment. In all these experiments, we use the algorithmic parameters that were discussed in Section 3.5. Given a cropped face

image of size 40×48, it takes about 4 seconds to generate the synthesized images and perform the tensor decomposition on an Intel Core i7 computer. It takes less than 0.05 second to estimate the geodesic distance between two tensors on a product manifold using the same machine.

3.6.1 CMU-PIE Dataset

First, experimental results on the *illumination subset* of the CMU-PIE dataset [2] are presented. Facial bounding boxes are obtained using the Viola-Jones object detection algorithm [8] without performing any pre-processing alignment step. We apply the same experiment settings as in [54]. The source domain which contains the frontal images (c_{27}) with good illumination (f_{21}) of all 68 subjects is used as our gallery. The target domain containing the remaining frontal images with 10 different illumination conditions is used as probe. The probe set is further divided into two subsets: a) Good Illumination (GI) consisting of f_{09}, f_{11}, f_{12}, and f_{20}, and b) Bad Illumination (BI) consisting of $f_{13}, f_{14}, f_{15}, f_{16}, f_{17}$ and f_{22}. The probe faces are blurred by convolving with Gaussian kernels of $\sigma \in (0.5, 1.0, 1.5, 2, 2.5, 3)$ and size $(2\sigma + 1) \times (2\sigma + 1)$ for each σ. Figure 3.4 shows some examples face images from the CMU-PIE database used in the experiments.

We compare the proposed method with the algorithms discussed in [86,97,98], and [54]. The Local Phase Quantization (LPQ) method in [97] utilized phase information computed locally for every image position in order to perform blur insensitive face classification. On the other hand, Nishiyama *et al.* [98] proposed a



(a) (b) (c)

Figure 3.4: Example images of a subject from the CMU-PIE used in the experiments: (a) clear and well-illuminated gallery image, (b) Good Illumination (GI) probe images, and (c) Bad Illumination (BI) probe images. A 7×7 Gaussian kernel with $\sigma = 3$ is used to blur the probe images.

method called FAcial DEblur INference (FADEIN) that attempted to infer a PSF representing the process of blur on faces. Gopalan *et al.* [86] performed blur robust face recognition by comparing subspaces created from a clean image and its blurred version on the Grassmann manifold. The Illumination-Robust Recognition of Blurred Faces (rIRBF) algorithm [54] handled blur and illumination variations in face recognition by comparing bi-convex sets formulated from face images at different blur and illumination conditions. Recognition rates of different approaches across domain shifts caused by illumination and (synthetic) blur variations on the CMU-PIE dataset are shown in Table 3.1. $\sigma = 0$ means that the recognition rates are obtained with only illumination variations and without blurring the probe faces. As there is only a single gallery image per subject, the nearest-neighbor classifica-

tion based on the geodesic distance on the latent domain is used in our approach. It can be seen from the figure that our method achieves consistently higher recognition rates compared to other algorithms in all combinations of illumination and blur. When the size of the blur kernel increases, the performance of all algorithms decreases. However, even at the worst scenario (kernel size of 7×7 at $\sigma = 3$, bad illumination), the proposed method still achieves the highest recognition rate at 92.4%, which is 11% higher than the next best result obtained by [54]. In the case of bad illumination and blur, the assumptions on the Lambertian model and quantization noise used in obtaining the synthesized blurred images may be violated, and thus lead to the reduction in the performance of our algorithm.

We also compare our algorithm with the results obtained by applying the Euclidean nearest-neighbor (NN) classification based on ℓ_2 norm directly on synthesized (blurred, relighted and transformed) images from the training set. It is clear from Table 3.1 that the performance of the method based on direct NN on synthesized images degrades by a large margin when the blur kernel size increases and is significantly lower than the recognition rates obtained by our algorithm. This can be explained by noting that the direct NN method only searches for the closest discrete point in the image space rather than modeling domain shifts due to multiple factor variations as in our approach.

Recognition results using the proposed approach without synthesizing images at different illuminations are also included. It can be seen from Table 3.1a that when the lighting component is held out, the performance of the proposed method remains approximately the same with good illuminated faces. However, in the case of

bad illumination in Table 3.1b, the recognition rates reduces significantly, especially when the size of the blur kernel is large. This shows the importance of modeling illumination variations in our approach when the lighting condition is bad.

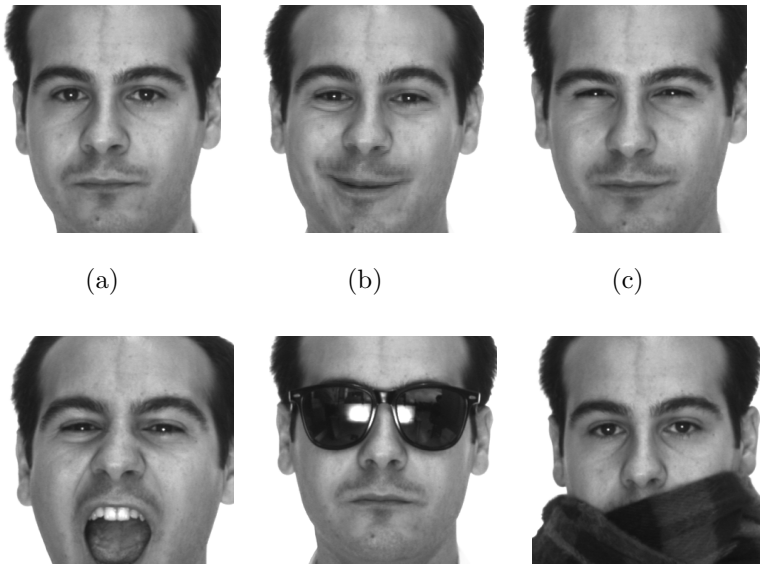
Table 3.1: Recognition rates (in %) of different approaches across illumination and (synthetic) blur variations on the CMU-PIE dataset. σ is the standard deviation of the Gaussian kernel used for blurring.

(a) Good Illumination (f_{09}, f_{11}, f_{12} and f_{20})

σ	0	0.5	1	1.5	2	2.5	3
LPQ [97]	99.63	99.63	99.63	99.63	97.05	79.42	46.32
FADEIN [98] + LPQ	98.53	95.6	93.6	91.2	89.8	88.60	87.13
Grassmannian [86]	99.63	99.63	99.63	99.63	99.63	96.32	93.38
rIBRF [54]	99.7	99.7	99.63	99.63	99.63	99.63	97.45
NN_2 with synthesized images	95.59	93.75	91.91	91.91	77.2	58.82	52.21
Our approach (without illumination)	100	100	100	99.63	99.63	99.26	98.53
Our approach	100	100	100	100	100	100	99.26

(b) Bad Illumination ($f_{13}, f_{14}, f_{15}, f_{16}, f_{17}$ and f_{22})

σ	0	0.5	1	1.5	2	2.5	3
LPQ [97]	99.1	97.79	96.08	88.97	73.04	58.08	27.7
FADEIN [98] + LPQ	91.5	87.7	81.8	69.11	62.74	56.37	44.61
Grassmannian [86]	85.71	84.66	84.24	79.2	71.01	67.23	60.92
rIBRF [54]	95.1	92.7	92.7	91.6	88.2	84.78	81.36
NN_2 with synthesized images	92.89	86.27	81.37	68.87	66.42	54.65	35.05
Our approach (without illumination)	98.77	98.77	98.77	96.08	96.08	92.65	88.48
Our approach	100	100	100	99.26	98.77	96.64	92.4



(a) (b) (c)

(d) (e) (f)

Figure 3.5: Six facial images of a subject from the first session in the AR dataset [7]. The faces are detected and cropped using the OpenCV implementation of the Viola-Jones object detection algorithm [8].

3.6.2 AR Dataset

In this section, experimental results for face images in the AR face dataset [7] are presented, which contain expression variations and real occlusions. It is worth mentioning that the proposed approach is not explicitly designed to handle domain shifts due to occlusion and expression. Hence, this offers a test case to analyze the robustness of our method to variations that are not synthesized.

The AR face database contains frontal images of more than 100 individuals taken over two sessions separated by two weeks time. Following the experimental setups in [4, 99], a total of 12 images per person are used in the experiments. Face

detection is also performed using the Viola-Jones object detection algorithm. However, unlike [4, 99], we do not perform any facial alignment step and instead, use the bounding boxes returned by the face detection algorithm directly in the recognition. The detected faces from six images of an individual in the first session of the AR dataset are shown in Figure 3.5. They are labeled a through f, and the corresponding images in the second session are labeled a' through f'. In addition to occlusions, there are also expression variations between the images.

Table 3.2 compares the recognition rates for different approaches on the AR dataset with a variety of training and testing sets. These are very challenging experiments as in many cases, there are approximately 50% occlusions in both the training and testing images. As a large part of the face images is occluded, the albedo cannot be reliably estimated and thus, we do not perform the synthesis for illumination. Furthermore, because the number of training samples is limited, the nearest-neighbor classification based on the geodesic distance is employed in our approach. Recognition rates using the simple Euclidean nearest-neighbor classification based on the ℓ_2 and ℓ_1 norms, NN_2 and NN_1, are also included. We also compare our algorithm with two methods, Partial Within-Class Match (PWCM) [99] and Partial Support Vector Machines (PSVM) [4], that are specifically designed to handle occlusion in face recognition. PWCM performs classification by reconstructing a test sample as a linear combination of the training samples from each class. The reconstruction is solely based on the visible data in the face images. On the other hand, PSVM extends SVM to handle occlusion by deriving a criterion that can handle the case of missing entries in the feature vectors.

Table 3.2: Recognition rates (%) with real occlusion on the AR dataset for a variety of training and testing sets. Results for other methods were obtained from [4]

Training Set	Testing Set	PSVM [4]	PWCM [99]	NN_2	NN_1	**Our approach**
$[e, f]$	$[a]$	96.0	89.0	45.0	79.0	91.0
$[e, f]$	$[a']$	79.4	71.0	31.0	50.0	76.0
$[e, f]$	$[b, c, d]$	80.0	72.0	31.7	59.7	66.3
$[e, f]$	$[b', c', d']$	58.7	47.3	20.3	32.7	52.7
$[e, f]$	$[e', f']$	57.0	55.0	25.5	29.0	66.5
$[e, f, e', f']$	$[b, c, d, b', c', d']$	86.6	76.2	31.3	56.5	79.8
$[e, f, e', f']$	$[a, a']$	96.4	95.0	48.5	83.0	95.0

It can be seen from the table that our method significantly outperforms both the Euclidean nearest-neighbor classification algorithms. Although our results are not as good as the ones obtained by PSVM, it is encouraging to see that the proposed algorithm is better than PWCM in most cases. Especially in the case where there are occlusions in both the training ($[e, f]$) and testing sets ($[e', f']$), the proposed algorithm outperforms both PSVM and PWCM by a large margin (66.5% compared to 57.0% and 55.0%, respectively). These experiments show that our method is robust to domain shifts caused by variation factors such as occlusion and expression even if they are not explicitly modeled.

3.6.3 UMD Remote Face Dataset

We then present recognition results on the UMD remote face dataset using the same data partitioning reported in [86]. This is an unconstrained dataset used

Figure 3.6: Example images of six subjects from the UMD remote face dataset. First row: source domain containing clean face images. Second row: target domain containing moderately blurred face images. Third row: target domain containing severely blurred face images.

for surveillance consisting of cropped faces of 17 subjects. In addition to moderate-to-severe blur, face images in the dataset also contain moderate variations in other factors such as pose, illumination, expression and occlusion. In this experiment, the source domain contains face images with variations such as illumination, occlusion and pose but without much blur. The target domains contain faces with moderate and severe amount of blur as well as other variations. Figure 3.6 shows some example images of the UMD remote face dataset with respect to different domains.

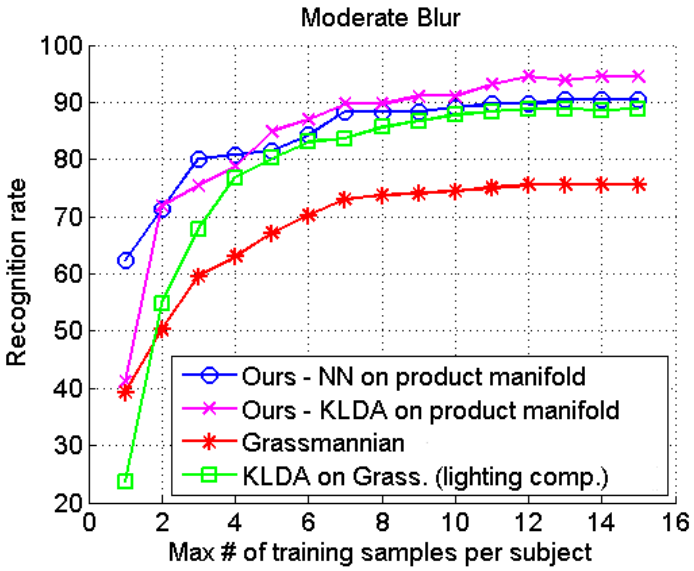
The comparisons between our approach and the method discussed in [86] are shown in Figure 3.7. It can be seen that our approach achieves better results for both

moderate and severe blur conditions regardless of the number of training samples. When only a single training image per subject is available, our approach based on the nearest-neighbor classification using geodesic distance on the latent domain still obtains significantly higher recognition rates compared to [86], for both moderate and severe blur cases. This is due to accounting for domain shift due to not only blur variations as in [86], but also for illumination and 2D alignment in our approach. The performance of the proposed KLDA algorithm on the latent domain increased significantly when more data are used in the training. The main reason is that it is able to learn the structure of the image space better by capturing domain shift due to other factors such as 3D pose and expression that are not explicitly modeled.

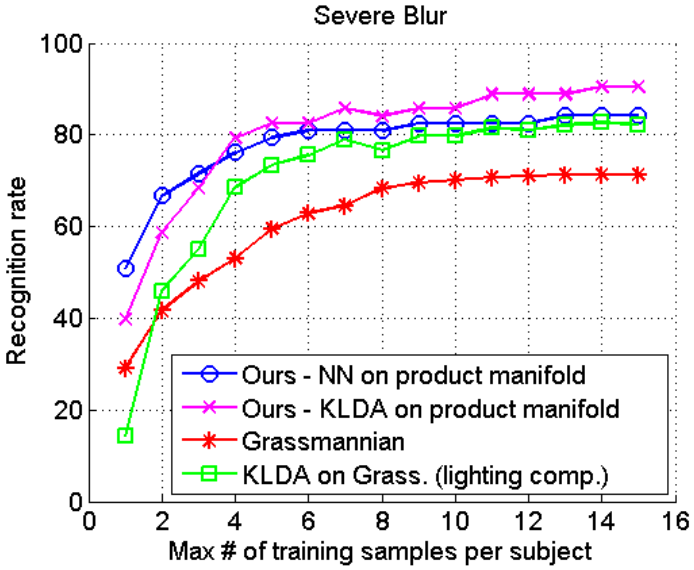
3.6.4 Honda/UCSD Video Dataset

Experiments on face recognition from videos were also conducted on the Honda/UCSD dataset [96]. This dataset contains 59 videos sequences of 20 different subjects. The number of frames in each video sequence varies from 12 to 645. Variations in illumination, pose, occlusion and expression appear across different sequences of each subject. The faces are also detected and cropped from the video frames using the Viola-Jones algorithm.

The proposed approach is compared with different algorithms such as [5, 100–103]. Kim *et al.* [100] presented a discriminative learning method based on canonical correlations (DCC) and applied it to image set classification. Another discriminative learning technique proposed Wang and Chen, called Manifold Discriminant Analysis



(a)



(b)

Figure 3.7: Recognition rates for moderate and severe blurred probe images on the UMD remote face dataset. (*This figure is best viewed in color*).

(MDC) [101], modeled each image set as a manifold and tried to find an embedded space to better separate manifolds from different classes. On the other hand, Cevikalp and Triggs [102] developed methods called AHISD (Affine Hull based Image Set Distance) and CHISD (Convex Hull based Image Set Distance) that characterized each image set by a convex geometric region (the affine or convex hull). The Sparse Approximated Nearest Points (SANP) algorithm [103] introduced a between-set distance defined as the nearest distance between sparse approximated points in the two sets. The last method used in the comparison is the Dictionary-based Face Recognition from Video (DFRV) algorithm [5] that extracted joint appearance and behavioral features from facial videos using dictionary learning.

We follow the experiment procedure in [103]: 20 sequences were used for training and the remaining 39 sequences for testing. Table 3.3 shows the recognition results obtained by using the algorithm presented in Section 3.4.2. The set length is the maximum number of cropped face images per video sequence. If the number of images in a sequence is less than the set length, all the images are used for classification. It can be seen from the table that our algorithm consistently outperforms all other methods in the comparison. This shows that when multiple video frames are available, the proposed KLDA on product manifolds is able to find an embedded space that separated face images from different individuals well.

Table 3.3: Recognition rates (%) on the Honda/UCSD dataset for different values of the maximum set length. The results for other methods were obtained from [5].

Set Length	DCC [100]	MDA [101]	AHISD [102]	CHISD [102]	SANP [103]	DFRV [5]	**Our approach**
50 frames	76.92	74.36	87.18	82.05	84.62	89.74	**97.44**
100 frames	84.62	94.87	84.62	84.62	92.31	**97.44**	**97.44**
Full Length	94.87	97.44	89.74	92.31	**100**	97.44	**100**
Average	85.47	88.89	87.18	86.33	92.31	94.87	**98.29**

3.6.5 Face Verification

In order to apply the proposed approach to face verification, given a pair of face images $\mathbf{I}_1$ and $\mathbf{I}_2$, two 4-th order tensors $\mathcal{A}^{(1)}$ and $\mathcal{A}^{(2)}$ are formulated from the synthesized images at different illumination, blur and 2D transformations as presented in Section 3.5. The vector $\mathbf{\Theta} = (\boldsymbol{\theta}_1^\top, \ldots, \boldsymbol{\theta}_4^\top)^\top$, with $\boldsymbol{\theta}_j$ is the vector of principal angles on the factor manifold $\mathcal{G}_{\bar{d}_j, d_j}$ $(j = 1, \ldots, 4)$, is computed from the pair of tensors and used as the feature in the training and testing based on Support Vector Machines (SVM) [19]. In this experiment, SVM with RBF kernel was used for classifying whether the pair of images was from the same individual or not. The optimal values of the parameters for training SVM are determined by performing cross validation on the training set.

We use the Labeled Faces in the Wild (LFW) [50] dataset for our experiments. It is a challenging dataset containing more than 13000 unconstrained face images from 5749 individuals. Face images in the dataset have large variations in pose,

(a) Same (b) Different

Figure 3.8: Examples of same and different pairs of face images from the LFW dataset.

illumination, age, expression, etc. Figure 3.8 shows examples of same and different pairs of face images from this dataset.

We follow the "image restricted" protocol [50]: only binary "same" or "different" labels are available for pairs of image, the identities of the images are unknown. The performance is measured by using 10-fold cross validation. In order to evaluate the robustness of the proposed algorithm to face cropping quality, we report our results on both unaligned and aligned cropped faces of the dataset [104].

Table 3.4 compares the face verification rates of different approaches on the LFW dataset on this image restricted protocol. The method proposed by Nowak and Jurie [105] used randomized binary trees to quantize the differences between local descriptors sampled from "same" and "different" image pairs. On the other hand, Wolf *et. al.* [106] developed a new patch-based descriptor based on Local Binary Patterns (LBP) [107]. Another method used in the comparison was proposed by Pinto *et. al.* [108] that combined V1-like models and multiple kernel learning

(MKL) for face verification. All of these above methods were evaluated on LFW images aligned using an unsupervised technique called funneling [109]. We also include the results obtained using a recent pose-invariant algorithm based on adaptive probabilistic elastic matching (APEM) [110] on both aligned and unaligned images. It can be seen from the table that the proposed approach outperforms other methods used in the comparison, except the APEM on aligned images. This is understandable as our algorithm does not explicitly synthesize pose and expression variations. Furthermore, a data-driven method such as KLDA on product manifolds cannot be applied as only pairs of face images are available without any identity information. However, the result is encouraging as we are able to outperform the APEM method on unaligned images, even though that method combines multiple features such as LBP [107] and SIFT [11] and is designed to handle pose variations. The verification rate obtained using the proposed approach on aligned images is only slightly better than when using unaligned images. This shows that our algorithm is not as sensitive to 2D face alignment as in other approaches, since this factor is explicitly accounted for using 2D perturbations. The Receiver Operating Characteristic (ROC) curves of different approaches are shown in Figure 3.9 in order to better evaluate their performances. The ROC curve of the proposed algorithm is obtained by thresholding the probability estimates computed using kernel SVM.

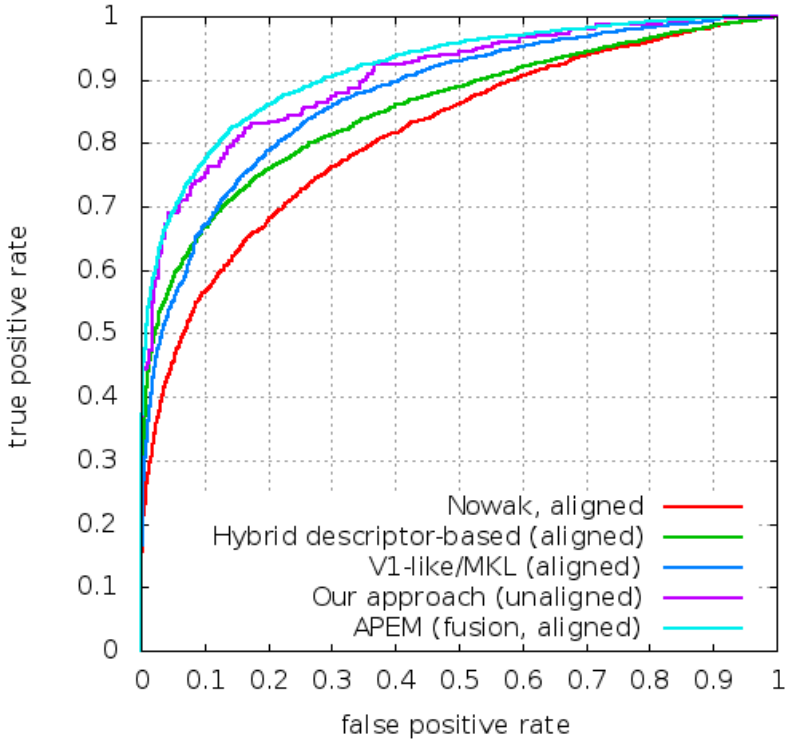


Figure 3.9: ROC curves of different approaches on the LFW dataset.

Table 3.4: Performance comparison for different methods on the most restricted LFW. Both mean classification rates and standard errors of the mean are reported.

Method	Accuracy ± Error (%)
Nowak (unaligned) [105]	72.45 ± 0.40
Nowak (aligned) [105]	73.93 ± 0.49
Hybrid descriptor-based (aligned) [106]	78.47 ± 0.51
V1-like/MKL (aligned) [108]	79.35 ± 0.55
APEM (fusion, unaligned) [110]	81.70 ± 1.78
APEM (fusion, aligned) [110]	84.08 ± 1.20
Our approach (unaligned)	82.67 ± 1.14
Our approach (aligned)	82.94 ± 0.83

3.7 Conclusions

We have shown that the underlying geometry of a set of face images of a person under multiple factor variations plays an important role in the recognition of face images from different domains. We showed that such a geometry can be studied by representing this set of images as a tensor and mapping the tensor to a point on a product manifold. The product manifold served as a latent domain where domain shifts due to multifactor variations such as illumination, blur and 2D alignment were jointly modeled. For cases where only a single gallery image per subject was available, geodesic distance was used to perform nearest-neighbor classification on the latent domain. Furthermore, a novel positive definite kernel based on an extension of the projection metric to the product space was proposed. When there were

sufficient samples available from the source domain, this projection kernel could be employed in any kernelized learning techniques to account for domain shifts due to other facial variations such as 3D pose and expression that were not explicitly modeled. Finally, a probabilistic method for classifying image sets on the latent domain using the KL divergence was also introduced. Competitive experimental results on different datasets showed the effectiveness of the approach in handling domain shifts caused by multifactor variations.

Chapter 4: Hierarchical Feature Learning using Sparse Coding for Facial Attribute Analysis

4.1 Introduction

Many applications in multimedia processing rely on the ability to extract information such as expression, age class, and gender from face images in order to improve user experience as well as to personalize content for each individual. However, this is a very challenging task due to the complex geometry of human faces as well as the appearance variations caused by factors such as illumination, pose, expression, occlusion, and resolution, especially when the face images are captured in unconstrained environments. In order to handle these variations, it is important to be able to effectively capture prominent visual structures from the input data at different scales and orientations.

The tasks of recognizing facial attributes including expression, age class, or gender can be considered as multi-class classification problems. A common pipeline for solving such problems consists of two stages: feature extraction and classification. In many current approaches, hand-crafted features such as LBP [12], Gabor [42], or SIFT [11] are used to obtain a representation of the input image. However, these

features usually lack explicit semantic meanings [111] and also do not take advantage of variations in training data. Furthermore, the process of designing these features requires expert knowledge and is often tedious as well as time consuming. An emerging trend with promising results in other computer vision and multimedia applications such as facial expression recognition [111], object classification [112, 113], and scene understanding [114], is that of learning feature representation directly from the training data. These approaches have been demonstrated to effectively capture complex relationships between different statistical patterns in data.

Motivated by the works in [112, 113], we propose a fully automatic approach for performing classification of different attributes including expression, age class, and gender from face images using hierarchical feature learning. The feature representations are obtained from the training data using dictionary learning, sparse coding, and spatial pooling. An overview of the proposed system is shown in Figure 4.1. First, facial bounding boxes and landmarks are detected from the input images using the methods of [9, 115]. In the next step, Procrustes analysis [28] is employed to align the detected landmarks to a reference mean shape in order to account for variations in 2D translations, rotations and scales. Hierarchical feature learning is performed separately in a local window at each landmark location. As a result, the number of encoders obtained from the feature learning process is the same as the number of extracted landmarks in each face. Given a face image, these encoders are used to obtain the local feature representations at the corresponding landmarks. The local features at all the landmarks are then concatenated into a single feature vector representing the whole face. The advantage of learning feature

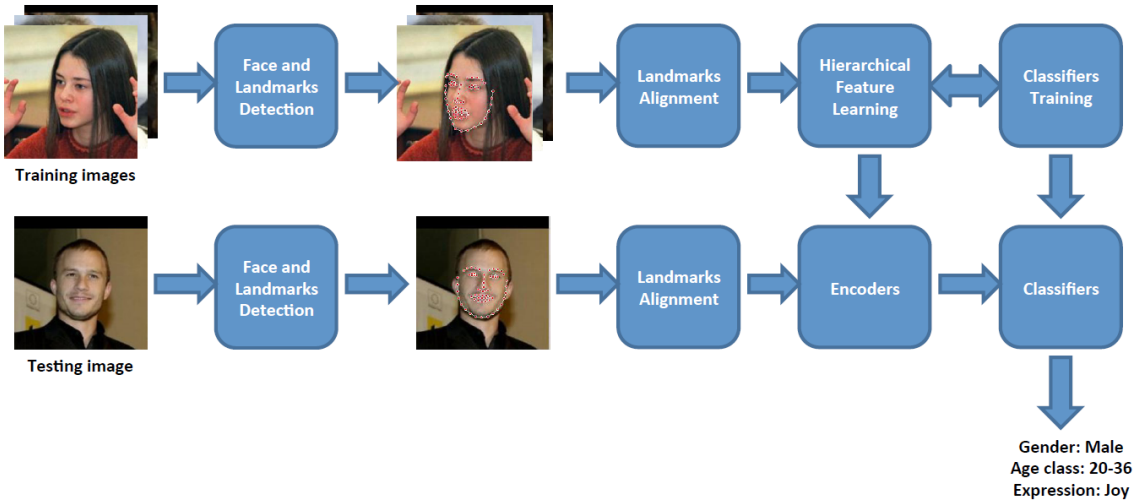


Figure 4.1: Analysis pipeline from input image to generative feature extraction and classification output *(best viewed in color).*

representations locally at each landmark location is that it provides the correspondences between local facial regions in the presence of 3D pose variations. However, in the case that the input images have very low resolution, landmarks cannot be reliably extracted and feature learning is carried out for the whole face with regularly sampled patches. Finally, the feature vectors of all training images are used to learn a set of classifiers - one for each facial attribute. In our approach, classification is done using the linear SVM [19] due to its effectiveness in handling high dimensional data.

The main contribution of this work is a multistage architecture for performing facial attribute analysis using sparse coding. To the best of our knowledge, this is one of the first works that employs sparse coding and deep neural networks for analyzing multiple facial attributes such as expression, age class, and gender at the same time.

The first advantage of the proposed method is that the features are learned using a network of multiple layers allowing us to better capture the richness of visual data. Another advantage is that the feature learning process is generative and can be employed with different label sets in order to solve different classification problems using the same set of features. Furthermore, the proposed system is fully automatic requiring no human intervention, and a single set of configuration parameters was used for all experiments.

Organization of the chapter: The remainder of the chapter is organized as follows. Related work is discussed in Section 4.2. Section 4.3 presents the details of the proposed approach. Experimental results are provided in Section 4.4. Section 4.5 concludes the chapter with a brief summary and discussion.

4.2 Related Work

Understanding attributes such as expression, age and gender from face images has been an active research topic for several years. This section summarizes relevant previous works on facial expression, age, and gender classification as well as feature learning for the proposed algorithm.

In most appearance-based facial analysis algorithms, there are two main steps: extracting features for facial representation and performing classification, usually by employing algorithms such as AdaBoost [116] or SVM [19]. In the first step, the representation can be obtained from either the whole face or local regions using different features such as Haar-like features [117,118], Gabor [119,120], local binary

patterns (LBP) [121, 122], or SIFT [123]. Although these algorithms have obtained reasonable performance in some scenarios, the use of hand-crafted features may not sufficiently capture the complex structures of human faces, especially in unconstrained settings where there are variations in pose, illumination, and resolution. In order to improve the performance of gender and age classification of face images taken in real- world settings, Shan [124, 125] proposes to learn discriminative LBP bins using AdaBoost. However, the main drawback of these methods is that their performance is still bounded by the discriminative power of LBP features.

Another family of approaches for analyzing facial attributes is based on modeling shape variations using facial landmarks. Turaga *et al.* [126] show that the space of landmarks can be interpreted as a Grassmann manifold and the problem of age estimation can be posed as a problem of manifold function estimation. Taheri *et al.* [127] uses a Riemannian interpretation of deformations that facial expressions cause on parts of the face to derive models for expressions on the affine shape-space. The common drawback of these algorithms is that they heavily rely on the accuracy of the landmark estimation process to model shape variations. As a result, these methods may encounter difficulties in uncontrolled environments when the landmark extraction is not reliable. Although landmark extraction is also used in our approach, it is more robust to low accuracy landmark locations. Feature representations are extracted on local windows centered at the landmarks and spatial pooling helps to handle the effect of small transformations in each local window. Furthermore, in the case where landmarks are not reliably extracted, for instance when the input face has low resolution, feature learning is instead performed over

the whole face.

The process of designing features for visual data is very challenging as well as time consuming. As a result, learning features from the raw data has attracted a lot of interest from the research community. An early work by Manjunath and Chellappa [128] extracts salient features at different spatial scales in the input image using a multistage system. Deep belief networks (DBN) [129] and its variants, such as deep autoencoders [130] and convolutional DBNs [131], learn high-level feature representations from unlabeled data and have been demonstrated to be effective for different classification problems. Inspired by the success of deep learning, the approaches in [113, 132] use multi-layer sparse coding networks to build feature hierarchies layer by layer. A recent work by Liu *et al.* [111] proposes a deep architecture, AU-aware Deep Networks(AUDN), for facial expression recognition by decomposing the appearance variations into a batch of local facial Action Units (AUs).

4.3 Our Approach

4.3.1 Face and Landmark Detection

Given an input face image, the face is first detected using the part-based face detector in [115]. The advantage of using this face detector over tradition methods such as the Viola- Jones face detector [8] is that it performs well for face images captured in unconstrained environments, possibly with large pose variations. Furthermore, [115] also detects landmarks from the input face. However, in this chapter we employ the facial landmark detection approach based on Constrained

Local Models (CLM) [9, 133] as it provides better localizations of the landmarks compared to [115]. In the CLM framework, the fitting of landmarks is posed as the search for the point distribution model (PDM) parameters, $\mathbf{p} = \{s, \mathbf{R}, \mathbf{t}, \mathbf{q}\}$, that jointly minimizes the misalignment error over all landmarks. The PDM parameters contain a global scaling s, a rotation $\mathbf{R}$, a translation $\mathbf{t}$, and a set of non-rigid parameters $\mathbf{q}$. The misalignment error can be written as [133]:

$$\mathcal{Q}(\mathbf{p}) = \mathcal{R}(\mathbf{p}) + \sum_{i=1}^{n} \mathcal{D}_i(\mathbf{x}_i; \mathcal{I}) \tag{4.1}$$

where $\mathcal{R}$ is a regularization term that penalizes complex deformations, $\mathcal{D}_i$ denotes the measure of misalignment for the i^{th} landmark $\mathbf{x}_i$ in the image $\mathcal{I}$. In the approach proposed by Asthana *et al.* [9], each $\mathcal{D}_i$ is a discriminant linear detector (i.e. patch expert) trained to detect part i^{th}. An illustration of the basic idea behind CLM fitting is shown in Figure 4.2, with more details in [9]. Figure 4.3 shows some examples of landmarks detected from unconstrained face images in the Labeled Faces in the Wild (LFW) [50] dataset using the approach in [9].

After the face and landmarks are detected from the input image, Procrustes analysis [28] is performed in order to align the input face with a reference mean shape. This alignment step accounts for variations in translation, in-plane rotation, and scale. As a result, it helps bring faces to roughly the same location in the image. An example of aligning a face image using the detected landmarks and Procrustes analysis is shown in Figure 4.4.

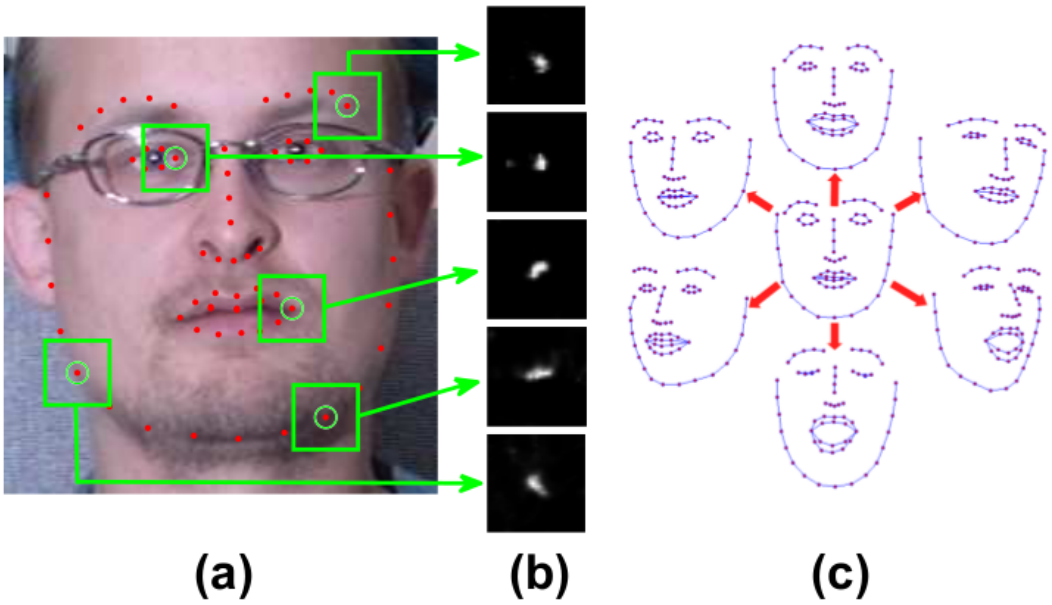


Figure 4.2: Overview of the CLM Framework: (a) Sample image Patches. (b) Computed response maps from exhaustive local search for landmarks. (c) Instances from the 3D Shape Model.

4.3.2 Dictionary Learning

Let $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n] \in \mathbb{R}^{d \times N}$ be a set of d-dimensional training samples, where each $\mathbf{y}_i$ $(i = 1, \ldots, N)$ is a vectorized image patch in our case, the task of learning a dictionary $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_K] \in \mathbb{R}^{d \times K}$ together with the sparse codes $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{K \times N}$ is typically posed as the following optimization problem:

$$\begin{aligned} \mathbf{D}^*, \ \mathbf{X}^* \ &= \ \underset{\mathbf{D},\mathbf{X}}{\operatorname{argmin}} \ \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \\ \text{s.t.} \quad & \|\mathbf{x}_i\|_p \leq \lambda, \forall i \in [1, N] \\ & \|\mathbf{d}_j\|_2 = 1, \forall j \in [1, K] \end{aligned} \tag{4.2}$$

Figure 4.3: Examples of detected landmarks from unconstrained face images in the LFW dataset using [9].

where $\|\mathbf{Y}\|_F$ denotes the Frobenius norm defined as $\|\mathbf{Y}\|_F = \sqrt{\sum_{i,j} |Y_{i,j}|^2}$, λ is a positive constant, and the constraint $\|\mathbf{x}_i\|_p \leq \lambda$ promotes sparsity in the coefficient vectors. The constraints $\|\mathbf{d}_j\|_2 = 1, j = 1, \ldots, K$, keep the columns of the dictionary (or dictionary atoms) from becoming arbitrarily large that may result in very small sparse codes.

Many algorithms have been proposed in the literature for solving the optimization problem in (4.2). In the case where the ℓ_0 norm is enforced, i.e. $\|\mathbf{x}_i\|_0 \leq \lambda$ where λ is the number of non-zeros in the coefficient vector, the K-SVD algorithm [134]

(a) Input image (b) Aligned image

Figure 4.4: An example of aligning a face image using the detected landmarks and Procrustes analysis.

can be used to train a dictionary. K-SVD is an iterative algorithm that operates by alternatively computing $\mathbf{D}$ and $\mathbf{X}$. The sparsity can also be promoted by enforcing the ℓ_1 norm on $\mathbf{X}$, i.e. $\|\mathbf{x}_i\|_1 \leq \lambda$. In this case, the online dictionary learning algorithm in [135] can be applied to solve the above problem. This algorithm approximates the optimal solution iteratively by efficiently minimizing at each step a quadratic surrogate function of the empirical cost over the set of constraints. More details about these methods can be found in [134] and [135]. In our method, K-SVD is used to learn the dictionary due to its efficiency compared to the online learning algorithm in [135].

4.3.3 Sparse Coding

After the dictionary $\mathbf{D}$ is learned, given a vectorized image patch $\mathbf{y}$, its sparse code $\mathbf{x}$ can be computed by minimizing the following objective function:

$$\|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 \text{ s.t. } \|\mathbf{x}\|_p \leq \lambda \tag{4.3}$$

where $\|\mathbf{x}\|_p$ can either be the ℓ_0 or ℓ_1 norm of $\mathbf{x}$. If the ℓ_0 norm of $\mathbf{x}$ is enforced, Orthogonal Matching Pursuit (OMP) [136] can be applied to compute the sparse code. OMP is a greedy algorithm that iteratively selects an element of the sparse code to be made non-zero to minimize the residual reconstruction error. In the case of enforcing the ℓ_1 norm of $\mathbf{x}$, the objection function in (4.3) can be formulated as a LASSO (Least Absolute Shrinkage and Selection Operator) problem:

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \beta\|\mathbf{x}\|_1 \tag{4.4}$$

where β is a positive regularization constant. A numerical method called LARS (Least Angle Regression Stagewise) [137] can be used to solve (4.4) for all possible values of β at once. Similar to OMP, LARS is also an iterative algorithm but it guarantees that the solution path is the global optimizer of (4.4). In the proposed approach, OMP is employed to compute the sparse codes of image patches in order to be consistent with the dictionary learning method discussed in Section 4.3.2. Furthermore, by pre-computing the inner products between image patches and dictionary atoms, a batch version of the OMP algorithm [138] can be employed to provide significant speed-up.

Rather than using the original sparse codes in the later stage of the framework,

a non-linear operation is applied on the sparse codes in order to obtain a new set of features. Instead of employing a sigmoid function as in traditional neural networks, the non-linear rectification is performed by separating the negative sparse coefficients from the positive ones. This operation is termed *POSNEG* and has been shown to play an important role in improving the final system performance [112, 139]. Given a sparse code $\mathbf{x} \in \mathbb{R}^K$, the rectified sparse code $\mathbf{u} \in \mathbb{R}^{2K}$ is obtained using POSNEG by setting

$$u_j = \max(0, x_j) \tag{4.5}$$

$$u_{j+K} = \max(0, -x_j) \tag{4.6}$$

where u_j and x_j are the elements at the index j of the vectors $\mathbf{u}$ and $\mathbf{x}$, respectively.

4.3.4 Hierarchical Feature Learning

When landmarks can be reliably extracted from the input face image, the process of learning feature representations is performed independently for each local window centered at each landmark location. The final feature vector for each face is the concatenation of the features learned at local windows centered at all landmark locations. However, when we cannot detect landmarks from the input face image, for instance when the image has low resolution, feature learning is carried out for the whole face. For simplicity, the following discussion of hierarchical feature learning considers the latter case.

Layer 1: First, dense sampling is performed on each training face image to obtain a set of small overlapping patches (e.g. 6×6). In order to reduce the effect

of illumination, each patch is contrast- normalized by subtracting the mean and divided by the standard deviation of its intensity values. As the number of patches obtained from the dense sampling can be very large, we randomly sample a subset of patches to use for learning the dictionary. For face images or local windows of size of at least 25×25, we found that a good trade-off between accuracy and computational efficiency can be obtained by setting the number of randomly sampled patches per face image or local window to 100. After the dictionary is learned, the sparse codes for all sampled patches are computed using the batch OMP algorithm. The POSNEG non-linear rectification is then performed on the sparse codes in order to obtain a new set of features. Spatial max-pooling is used to aggregate the sparse codes over each spatial cell:

$$\mathbf{z}_j = \underset{j \in \mathcal{N}_i}{\text{elem\_max}} \, \mathbf{u}_{ij} \tag{4.7}$$

where $\mathbf{u}_{ij}$ are rectified sparse codes in each spatial cell $\mathcal{N}_i$, and elem_max is the element-wise maximum operator. In our approach, max-pooling is carried out over spatial cells of size 4×4. The reason for selecting max-pooling over other methods of pooling such as average- pooling is because it is particularly well-suited for the separation of sparse features [140]. Figure 4.5 visualizes the process of local pooling over each spatial cell.

Layer 2: Similar computations are performed in this layer except that the input is not image intensities but instead, the pooled sparse codes from the previous layer. The pooled features obtained from the previous layer are further aggregated by concatenation over each 2×2 neighborhood and contrast- normalized. The

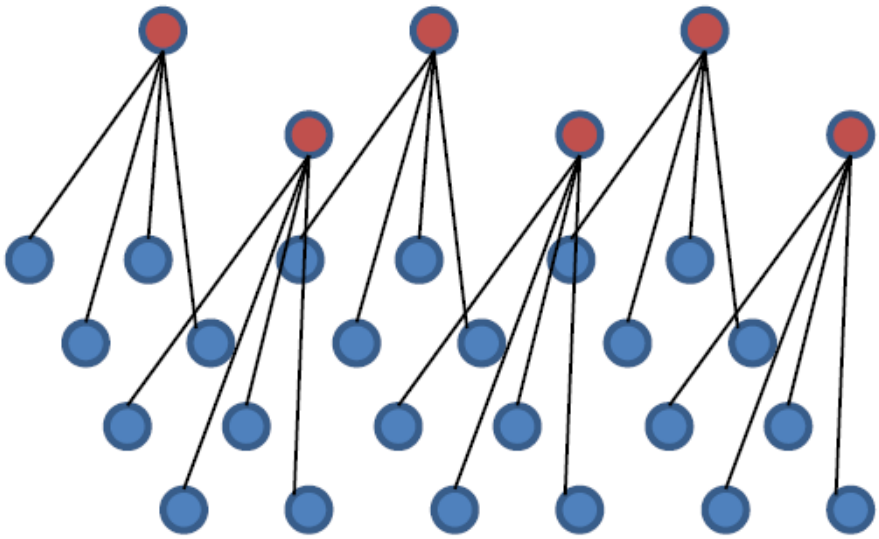


Figure 4.5: Local pooling over spatial cells. For ease of viewing, the size of each spatial cell is set to 2×2. Blue circles are original feature vectors and each red circle is a pooled vector over a spatial cell *(best viewed in color)*.

concatenated feature vector helps handle local variations in illumination as well as foreground-background contrast. This contrast-normalization is different from the one in the first layer and is performed as follow:

$$\hat{\mathbf{f}} = \frac{\mathbf{f}}{\sqrt{\|\mathbf{f}\|^2 + \epsilon}} \tag{4.8}$$

where $\mathbf{f}$ is the concatenated feature, and ϵ is a small positive constant. The value of ϵ is set to 10^{-7} as it is found to work well in our experiments. However, as long as ϵ is small enough, it does not affect the performance of the system significantly. At the end of this layer, spatial pyramid max-pooling is used to aggregate the sparse codes. In our approach, a three-level spatial pyramid is employed to compute the final feature vector for each face image. A visualization of this spatial pyramid is shown in Figure 4.6. By performing max-pooling over the whole image (or local

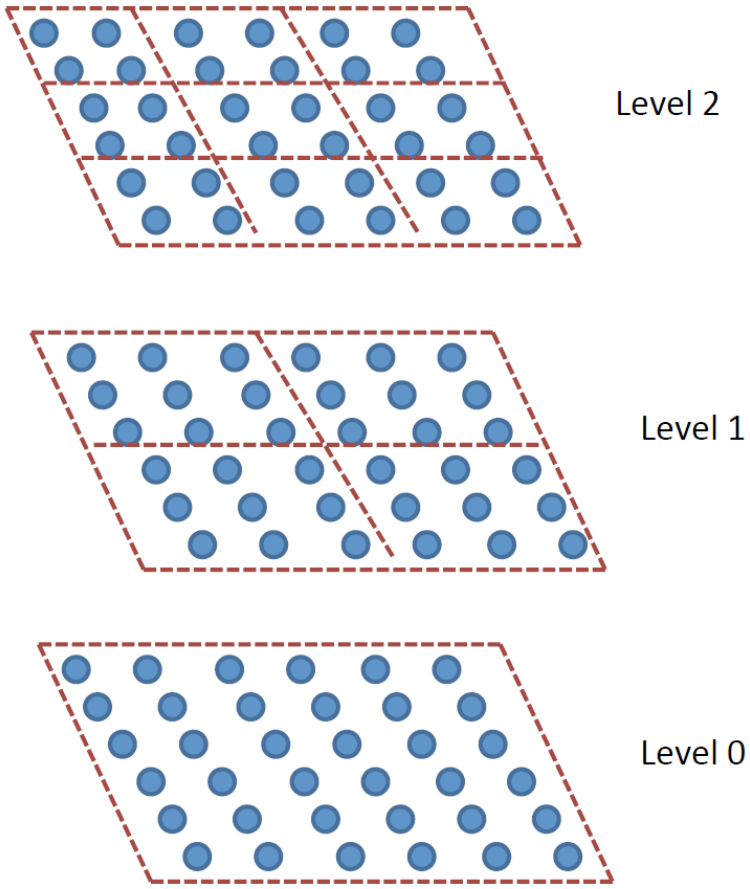


Figure 4.6: A three-level spatial pyramid used in the proposed approach.

window), a single feature vector is obtained at level 0 of the spatial pyramid. At level 1, four feature vectors are obtained from dividing the image into four quadrants and performing max-pooling on each quadrant. Similarly, the image is divided into 9 quadrants in level 2, yielding nine feature vectors. The final vector is obtained by concatenating 14 feature vectors from the spatial pyramid.

4.3.5 Implementation

The proposed framework is written in C++ and optimized to run on computing clusters using a hybrid MPI-OpenMP implementation. Jobs are divided to

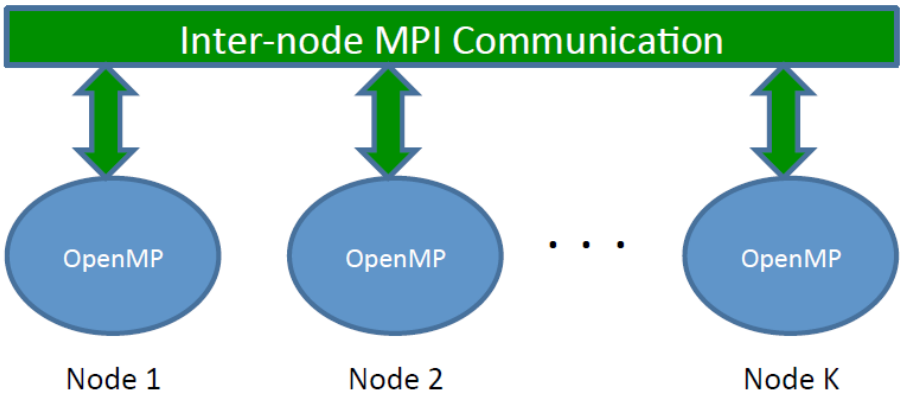


Figure 4.7: Visualization of a hybrid MPI-OpenMP implementation on a cluster of K nodes.

nodes in a cluster using Boost.MPI[1], a C++ interface to the standard Message Passing Interface (MPI). OpenMP is employed for parallelization inside each node in order to reduce latency from data movement between nodes. A visualization of this implementation is shown in Figure 4.7.

In all experiments reported in Section 4.4, the training and testing are performed on a cluster of six nodes with 24 2.2Ghz processors on each node. It takes around 30 minutes to learn the features using a two-layer network for 20000 images of size 61×49 (i.e. for the Images of Groups dataset in Section 4.4.2.1).

4.4 Experiments

Experimental results on different constrained and unconstrained datasets are reported in this section. For the Extended Cohn-Kanade (CK+) dataset [141] and Labeled Faces in the Wild [50] dataset, we report the results using feature learning

[1] `http://www.boost.org`

for both with and without landmarks as they can be reliably detected from the face images in these datasets. For the Kaggle facial expression challenge dataset [142] and Images of Groups dataset [143] dataset, we do not perform landmark detection and feature learning is carried out for the whole face image as many of these images have low resolution.

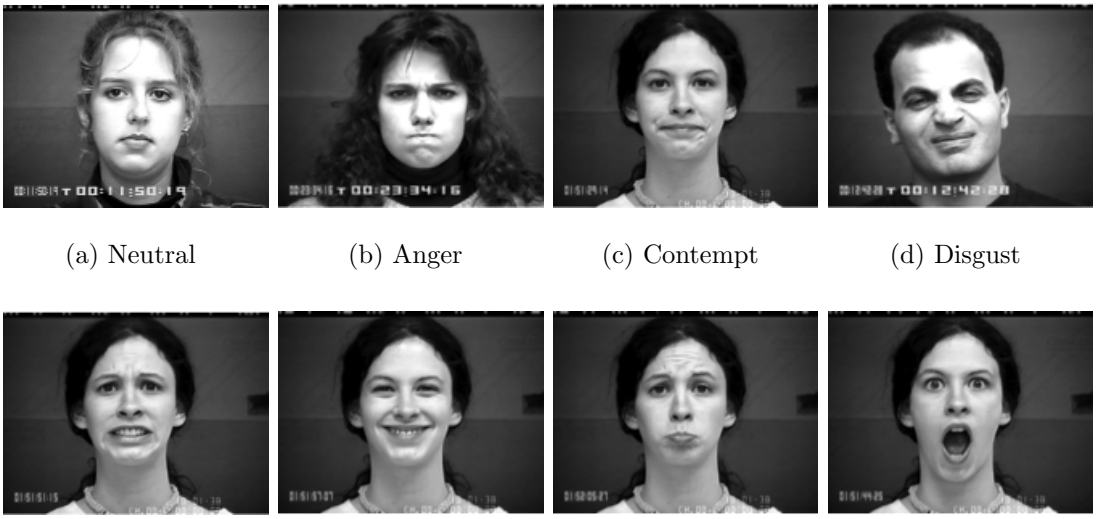
Parameter settings: In the case that landmarks are used, local windows of size 25×25 centered at each landmark location are extracted. The patch size is set equal to 6×6. The number of the dictionary atoms is set to 500 and 1000 for the first and second layer, respectively. For both layers, the value of λ is set to 4 for training and 40 for testing, respectively. This is similar to the finding in [139] as smaller sparsity constant is needed to make the learning more stable. Furthermore, we need a higher sparsity constant (i.e. denser feature vectors) in order to better capture the structure of the test samples. Max-pooling is performed over spatial cells of size 4×4 in the first layer. The feature vectors obtained from the first layer are aggregated over 2×2 spatial cells before feeding to the second layer. Final feature vectors are obtained by performing max-pooling over a three-level spatial pyramid, partitioned into 1×1, 2×2, and 3×3. Linear SVMs with regularization parameters of 100 and 1 are employed for classification with and without landmarks, respectively. This is due to the difference in the dimensions of the final feature vectors in each case. The values of these parameters are obtained by using the findings in [113, 134, 139] to create a small subset of values, and perform cross-validation on the training data to obtain the optimal settings.

4.4.1 Expression Classification

4.4.1.1 Extended Cohn-Kanade (CK+) Dataset

In this section, we present the expression recognition results for the CK+ dataset [141]. This dataset contains 593 image sequences of 123 subjects taken under controlled conditions. However, there are only 327 sequences with seven valid expression labels: Anger (An), Contempt (Co), Disgust (Di), Fear (Fe), Happiness (Ha), Sadness (Sa), and Surprise (Su). Figure 4.8 shows examples of these expressions. For each sequence, the first frame (Neutral) and three peak frames with the most expressions are used and results are reported over 10-fold cross-validation. As the resolution of images in this dataset is relatively high (640×940), we can reliably extract 66 landmarks from each face and the final feature vector is the concatenation of the features obtained at the local window centered at each landmark location.

Table 4.1 shows the expression recognition results for different methods on the dataset. We compare our approach with the results obtained by using hand-crafted features such as LBP, SIFT, and HOG. It can be seen from the table that the proposed method significantly outperforms these approaches. It is worth mentioning that linear SVMs is used in our method whereas SVMs with non-linear RBF kernels are used with the mentioned approaches. The proposed method is also compared against CSPL [144] that learns common and specific patches for discriminating facial expressions. Even though CPSL only handles six expression categories from 96 subjects, it still does not perform as well as our approach in



(a) Neutral (b) Anger (c) Contempt (d) Disgust

(e) Fear (f) Happy (g) Sad (h) Surprise

Figure 4.8: Examples of different facial expressions in the CK+ dataset.

this dataset. Furthermore, recognition results using over-complete representations (OR) [111] and AU-aware Receptive Fields (AURF) [111] are also included in the comparison. AURF achieves the accuracy of 92.22, which was the state-of-the-art performance in the CK+ dataset. The main drawback of AURF is that it uses the class information in the process of feature learning and thus, the learned representations cannot be applied to recognize other facial attributes. Finally, the performance obtained using our algorithm to learn features for the whole face without landmark extraction is also reported. It can be seen from the table that learning features at landmark locations helps improve the absolute recognition accuracy by nearly 2% on this dataset.

In order to better assess the detailed performance of our algorithm, the confusion matrix is shown in Table 4.2. It can be seen from the confusion matrix that the recognition accuracies of Contempt, Fear, and Sad are not as good as that of

Table 4.1: Expression recognition accuracy on the CK+ dataset.

Method	Accuracy
LBP (SVM with RBF kernel) [111]	83.37%
SIFT (SVM with RBF kernel) [111]	86.39%
HOG (SVM with RBF kernel) [111]	89.53%
CSPL (SVM with unknown kernel) [144]	89.89%
OR (linear SVM) [111]	91.44%
AURF (linear SVM) [111]	92.22%
Our approach (without landmarks, linear SVM)	**93.04%**
Our approach (with landmarks, linear SVM)	**94.65%**

other expressions. This may be caused by the limited number of training samples in these categories compared to other expressions.

4.4.1.2 Kaggle Facial Expression Challenge Dataset

In this section, we report the results on the Kaggle facial expression challenge dataset [142]. This dataset contains unconstrained images collected using Google image search. There are 28709 training images and 7178 testing images with seven expression categories: Neutral, Anger, Disgust, Fear, Happy, Sad, and Surprise. Figure 4.9 shows some examples of faces with different expressions in the dataset. It can be seen that this is a very challenging dataset due to the appearance variations of the face images as a result of pose, illumination, occlusion, as well as other

Table 4.2: Confusion matrix for expression recognition on the CK+ dataset using our method with landmarks.

	Neutral	Anger	Contempt	Disgust	Fear	Happy	Sad	Surprise
Neutral	97.86	0.61	1.53	0	0	0	0	0
Anger	1.48	91.11	2.96	2.22	0	0	2.22	0
Contempt	5.56	1.85	88.89	0	0	0	0	3.7
Disgust	0	0	0	94.92	0	1.69	1.69	1.69
Fear	0	0	0	0	84.00	4.00	0	12.00
Happy	0	0	0	0	0	98.55	0	1.45
Sad	0	7.14	3.57	3.57	0	0	85.71	0
Surprise	0	0	1.20	1.20	0	0	1.20	96.39

factors. Furthermore, as the resolution of the images is low (48×48), we cannot reliably detect landmarks from the face and thus, only report the result obtained by considering the whole face. In order to increase the number of training samples and avoid significant over-fitting, the training set is supplemented with images obtained by performing similarity transformations on the training images. The comparisons between different methods are shown in Table 4.3. The method proposed by Ionescu *et al.* [145] uses SIFT and multiple kernel learning (MKL) to perform the classification. It achieves the best recognition rate (67.48%) in the challenge for methods that do not use feature learning. The top performer in the challenge [146] uses the primal objective of an SVM as the loss function for training a deep network and obtains a recognition rate of 71.16%. Although our approach does not perform as

(a) Neutral (b) Anger (c) Disgust (d) Fear

(e) Happy (f) Sad (g) Surprise

Figure 4.9: Examples of different expressions in the Kaggle facial expression challenge dataset.

well, it does not require label information in the feature learning process. However, it is possible for the proposed approach to accommodate label information in the learning process by training dictionaries discriminatively using approaches such as [147, 148]. A drawback of these approaches is that they are computationally expensive compared to learning a generative dictionary and may not be applicable with high-dimensional data. We intend to pursue this as a future work.

4.4.2 Age Class and Gender Classification

In this section, we report the results of age class and gender classification on the Images of Groups dataset [143] and gender classification on the LFW dataset [50].

Table 4.3: Expression recognition accuracy on the Kaggle dataset.

Method	Accuracy
Radu + Marius + Cristi [145]	67.48%
RBM [146]	71.16%
Our approach	**69.35**%

4.4.2.1 Images of Groups Dataset

The Images of Groups dataset contains 28231 faces from 5080 images collected from Flickr. Many faces in the dataset have low resolution with the median face having only 18.5 pixels between the eye centers, and 25% of the faces have under 12.5 pixels. All faces are normalized to 61×49 based on the eye centers. As a result, the feature learning in our approach is performed for the whole face, not at each landmark location. Each face in the dataset is labeled with gender and one of seven age classes: 0-2, 3-7, 8-12, 13-19, 20-36, 37-65, and 66+. Figure 4.10 shows some example faces from the dataset with different gender and age class.

Following the experimental setup in [143], we randomly sample 3500 faces that are uniformly distributed among all age categories to use as the training set. An independent set of 1050 randomly sampled faces is used as the testing set. In our approach, the same feature representation is used for both classifying age class and gender in the dataset. For age classification, the accuracy of an exact match (AEM) and the accuracy of allowing an error of one age category (AEO) (e.g. a 3-7 year old classified as 8-12) are used as evaluation measures [149]. Table 4.4 reports the

Figure 4.10: Example face images with different age class and gender from the Images of Groups dataset.

age class recognition accuracies of different approaches on the dataset. It can be seen from the table that our approach achieve more than 3% improvement in the AEM metric over the next best result in [124]. The age classification accuracy with respect to each age class is shown in Figure 4.11. Intuitively, it is expected that the proposed approach performs really well for the infant age class (0-2) and the elder age class (66+) as the appearance of human faces in these two classes are very different from the remaining age classes.

Gender classification results of different approaches are shown in Table 4.5. The proposed method also outperforms all other algorithms used in the comparison

Table 4.4: Age classification results on the Images of Groups dataset.

Method	Classifier	AEM	AEO
Appearance [143]	Gaussian maximum likelihood	38.3%	71.3%
Appearance + Context [143]	Gaussian maximum likelihood	42.9%	78.1%
Gabor [124]	Adaboost	43.7%	80.7%
LBP [124]	Adaboost	44.9%	83.0%
Boosted Gabor [124]	SVM (RBF kernel)	48.4%	84.4%
Boosted LBP [124]	SVM (RBF kernel)	50.3%	87.1%
PLO [149]	Ordinal hyperplane ranker [150]	48.5%	88.0%
Our approach	**Linear SVM**	**53.4**%	**90.7**%

for gender classification on this dataset. The performance of the proposed method for age and gender classification on this dataset is very encouraging given that the dataset was taken in unconstrained conditions and thus, is very challenging. Figure 4.12 shows the gender classification accuracy for different age classes. We can observe from the figure that it is more difficult to recognize gender for infant and young children (from 0 to 12 years old) from their face images than for older humans. This is because their facial features have not been discriminative enough given the ages.

4.4.2.2 Labeled Faces in the Wild (LFW) Dataset

The LFW dataset [50] contains 13233 labeled images from 5749 individuals collected from the web. There are 2977 females and 10256 males in the dataset. The gender classification results are reported for both feature learning with and

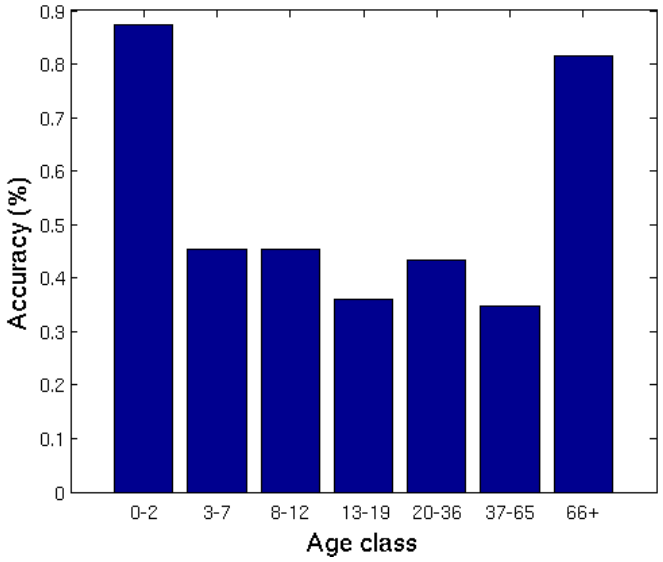


Figure 4.11: Age classification accuracies for different age classes on the Images of Groups dataset.

without using landmarks over 5-fold cross-validation. The folds can be downloaded from `http://face.cs.kit.edu/download/LFW-gender-folds.dat`. All images of individual subjects are only in one fold at a time in order to prevent the algorithm from learning the identity of the persons rather than the gender. When landmarks are not used, all the extracted faces are resized to 48×48 in order to test the performance of the algorithm under low resolution. We also do not perform any preprocessing step to align the faces before learning the features in this case.

Table 4.6 compares the gender classification results of different approaches on the LFW dataset. It is worth mentioning that the methods in [125] are performed only on 7443 frontal faces of the dataset. They do not consider non-frontal faces as well as faces that are difficult to establish the ground truth. Furthermore, in their methods, all the faces were aligned with a commercial face alignment software

Table 4.5: Gender classification results on the Images of Groups dataset.

Method	Accuracy
Appearance [143]	69.6%
Appearance + Context [143]	74.1%
Gabor + Adaboost [124]	70.2%
LBP + Adaboost [124]	71.0%
Boosted Gabor + SVM [124]	73.3%
Boosted LBP + SVM [124]	74.9%
Our approach (without landmarks, linear SVM)	**77.7%**

and have high-resolution (250×250). As a result, it is very encouraging to see that the performance of our approach using feature learning without landmarks is only marginally less than their results, even we perform classification low-resolution, unaligned, and non-frontal faces. When landmarks are used, the performance of our method is significantly better (98.38%) compared to the ones obtained by other approaches. It can be seen from the table that the classification rates for female are always lower than that for male due to the imbalance of the number of training samples between two classes.

4.5 Conclusions

We have presented a hierarchical approach for performing feature learning using sparse coding with applications to facial attribute analysis. The proposed approach compares favorably, and in many cases, significantly outperforms state-

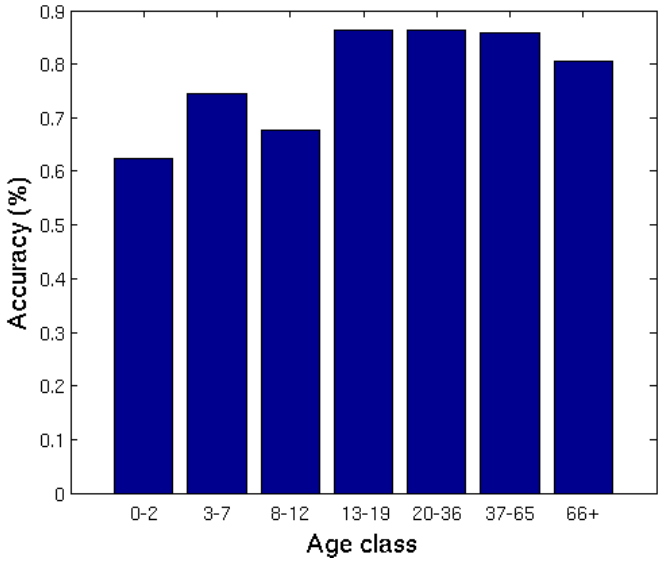


Figure 4.12: Gender classification accuracies for different age classes on the Images of Groups dataset.

Table 4.6: Gender classification results on the LFW dataset.

Method	Classifier	Accuracy		
		Female	Male	Overall
Standard LBP [125]	SVM with RBF kernel	89.78%	95.73%	93.38%
Boosted LBP [125]	AdaBoost	91.58%	95.98%	94.40%
Boosted LBP [125]	SVM with RBF kernel	92.02%	96.64%	94.81%
Our approach (low resolution, unaligned, without landmarks)	**Linear SVM**	**83.56%**	**95.18%**	**92.57%**
Our approach (with landmarks)	**Linear SVM**	**95.64%**	**99.17%**	**98.38%**

of-the-art methods in different classification tasks. Unlike other feature learning algorithms that use label information in the training process, the feature learning process in our method is generative and a common feature representation can be

used to train an arbitrary set of classifiers for different facial attributes such as expression, age class, and gender.

Chapter 5: Head Pose Estimation using Randomly Projected Dense SIFT Descriptors

5.1 Introduction

Head pose estimation is the process of finding the 3D orientation of a human head from an input face image. It has been widely employed in many applications such as human-computer interaction, gaze direction detection and multi-view face recognition. For instance, in human-computer interaction, especially for computer gaming, the ability to accurately estimate the head pose plays an important role in interpreting head gesturing [151]. In driver monitoring, it is critical to be able detect the driver's eye gaze direction in order to help avoid vehicle accidents. It was shown by Langton *et al.* [152] that the head pose is highly correlated with the gaze direction. In several pose-invariant face recognition algorithms that render frontal views from non-frontal face images [27,35,36], head pose estimation is an important pre-processing step in the synthesizing process.

Head pose estimation is a very challenging problem due to several factors including projective geometry, illumination variations, facial expressions, subject variability and camera distortion. Different techniques have been proposed in order

to tackle these challenges. One of the most popular approaches is using Support Vector Regressors (SVRs) [20] trained from face images captured at different viewing directions to predict the head pose [153, 154]. However, the main drawback of these approaches is that a single SVR may not capture complex variations in the face image space resulted from varying the head pose. An improvement proposed by Guo *et al.* [155] employs Support Vector Machines (SVMs) [19] to add a small correction to the head pose estimate returned by SVR. However, if the initial estimate obtained using SVR deviates far away from the true head pose, the correction by SVM may not be sufficient to bring the it close to the true value.

In this chapter, an automatic method for estimating the head pose from a single 2D face image is presented. In the proposed approach, rather than employing a single SVR to predict the whole range of head pose, an SVM is first applied to provide a coarse estimation. Multiple SVRs, each trained for a different interval of the head pose range, is then used to refine the initial estimate. Rather than using original intensity values, dense SIFT descriptors are extracted from image grid points in order to obtain a representation that is robust to noise and illumination variations. Random Projection (RP) is used to reduce the dimension of the concatenated descriptor vector for efficient processing. The advantage of the proposed approach is that it does not depend on the extraction of facial feature points such as the mouth and eye corners and the nose tip, which by itself is a challenging process. In addition, the proposed method is fully automatic. The overview of our approach is illustrated in Figure 5.1.

Organization of the chapter: Section 5.2 discusses some related works. The

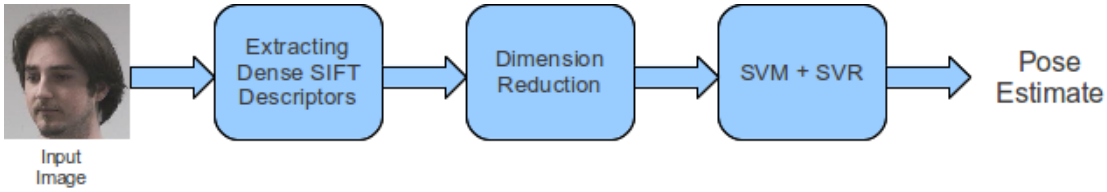


Figure 5.1: Overview of the proposed head pose estimation method.

proposed head pose estimation algorithm is presented in Section 5.3. Experimental results on different datasets are given in Section 5.4.

5.2 Related Work

Many algorithms have been proposed in the literature in order to solve the problem of head pose estimation. This section provides a brief review of different head pose estimation approaches. A more detailed survey on head pose estimation methods can be found in [13].

By directly comparing a given image with a set of exemplars, *appearance template methods* estimate the head pose of the input face as the 3D angles of the most similar template. The comparison is carried out by either using mean squared error [156], normalized cross-correlation [157], or Gabor wavelets [158]. The main advantage of these methods is that the reference set can be expanded anytime to adapt to changing conditions. Furthermore, they do not require facial feature points or negative training samples [13]. However, these techniques are sensitive to noise caused by illumination and expression changes as the matching processes are based on pair-wise similarities. In addition, they are only capable of inferring discrete pose values.

Classification-based methods [159] learn head pose classifiers by dividing the training images into a discretized space of poses. The most commonly used classifiers for this task are multi-class SVMs [160] or multi-class linear discriminant analysis (LDA) [161]. The improvement of these approaches over the appearance template methods is that they learn to ignore the appearance variations not corresponding to the changes in head pose. However, both the appearance template and classification-based methods can only return discrete poses and also suffer from non-uniform sampling in the training data.

In order to obtain continuous pose estimates, *regression-based methods* learn continuous mapping functions between the face image and the pose space. The regression can be performed using algorithms such as Support Vector Regression (SVR) [153, 154], Gaussian Process Regression (GPR) [162], or neural networks [163, 164]. A recent work by Haj *et al.* [165] applies Partial Least Squares (PLS) to the problem of head pose estimation. The main drawback of these approaches is that it is not clear whether the learned mapping function is able to capture the complex variations in the data well enough [13].

Manifold embedding methods [166–168] assume that the variations in head pose lie in a low-dimensional manifold. In these approaches, the manifold embedding is learned from the training data and the head pose estimation is performed on the low-dimensional space. The main weakness of manifold embedding methods is that appearance variation is a result of not only pose changes but also other factors such as identity and lighting changes.

By employing the relative configuration of facial features such as eyes, mouth

corners and the nose tip, *geometric methods* [169–171] can obtain an estimate of the pose using projective geometry. The main advantage of these techniques is that they are very fast and simple once the facial feature points are obtained. On the other hand, these techniques depend on the feature extraction process and are susceptible to outliers and missing features.

5.3 Head Pose Estimation

5.3.1 Dense SIFT Descriptors

The Scale Invariant Feature Transform (SIFT), proposed by Lowe [11], is one of the most popular algorithms for extracting keypoints from an image. At each detected keypoint, a local descriptor is created by forming a histogram of gradient orientations and magnitudes of image pixels in a small window centered at this point. The size of the local window is usually chosen at 16×16. It is then divided into sixteen 4×4 sub-windows. Gradient orientations and magnitudes are estimated within each sub-window and put into an 8 bin histogram. The histograms of the sub-windows are concatenated to create a 128-dimensional feature vector (descriptor) of the keypoint.

In our approach, local SIFT descriptors are extracted at regular image grid points, rather than only at keypoints, in order to form a dense description of the input face image. This dense representation was also employed successfully for image alignment and gender classification in [18] and [172], respectively. The advantage of this representation is that it does not depend on the matching of keypoints, which

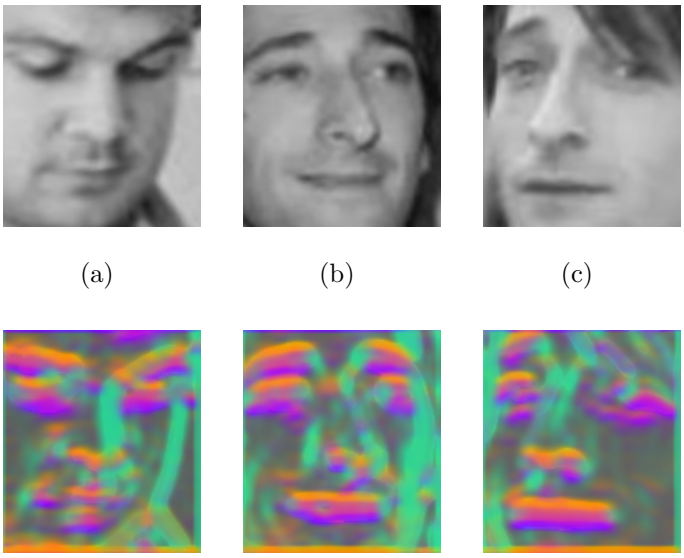


Figure 5.2: Input face images at different poses and the corresponding visualizations of their dense SIFT descriptors.

is often challenging when significant pose and illumination variations are present between the input images. Figure 5.2 shows the input face images at different poses and their corresponding dense SIFT descriptors. In the second row of the figure, the first three principal components of each descriptor are mapped into the principle components of the RGB color space in order to visualize purpose. Similar to [18], the first component is mapped into R+G+B, the second and third components are mapped into R-G and R/2+G/2-B, respectively.

5.3.2 Dimension Reduction using Random Projection (RP)

As the length of the descriptor extracted at each image grid point is 128, the dimension of the concatenated feature vector for the whole input face image

will become significant. In order to improve the efficiency of the proposed algorithm, PCA [83] can be used to project the concatenated feature vector into a lower-dimensional subspace. However, the eigenvalue decomposition of the data covariance matrix is very computationally expensive due to the large dimension of the feature space. A more efficient way to reduce the dimension of the feature vectors is by projecting them onto a random lower-dimensional subspace.

The key idea of random projection comes from the Johnson-Lindenstrauss (JL) lemma [173]:

Lemma 5.3.1. *(Johnson-Lindenstrauss) Let $\epsilon \in (0,1)$ be given. For every set Q of $\#(Q)$ points in $\mathcal{R}^N$, if n is a positive integer such that $n > n_0 = O\left(\frac{ln(\#(Q))}{\epsilon^2}\right)$, there exists a Lipschitz mapping $f : \mathcal{R}^N \to \mathcal{R}^n$ such that*

$$(1-\epsilon)||\mathbf{u}-\mathbf{v}||^2 \leq ||f(\mathbf{u})-f(\mathbf{v})||^2 \leq (1+\epsilon)||\mathbf{u}-\mathbf{v}||^2 \tag{5.1}$$

for all $\mathbf{u}, \mathbf{v} \in Q$.

Basically, this lemma states that the pairwise distances between any two points are approximately maintained when the points are projected onto a random subspace of suitably high dimension. It is often the case that the performance of a wide variety of machine learning algorithms when given access to only randomly projected data is *essentially the same* as their performance on the original dataset [174].

Because the majority of patches in a face image are uniform, when estimating the SIFT descriptors, there are many bins in the histogram of image gradients with zero values. As a result, the concatenated descriptor vector is sparse. Figure 5.3 shows the SIFT descriptors extracted from two different locations in a face image.

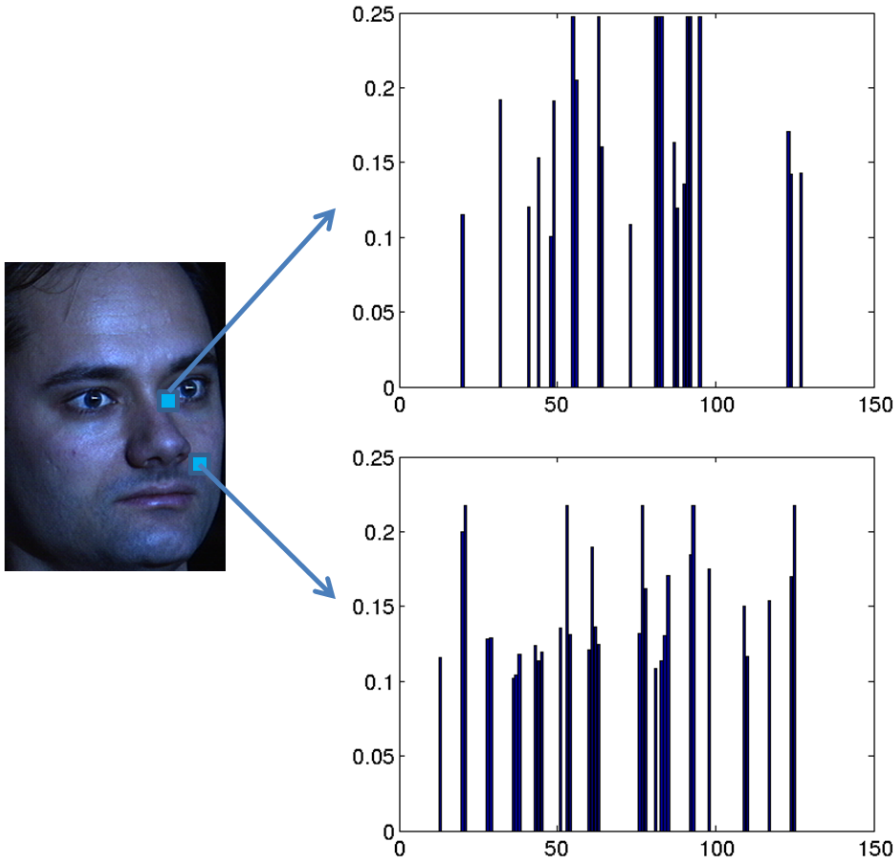


Figure 5.3: SIFT descriptors extracted from two different locations in a face image.

The sparsity of the concatenated SIFT descriptor vectors helps to further improve the efficiency of the random projection. For K-sparse signals (i.e. have at most K non-zero entries), the computational complexity of the random projection reduces from $O(nNC)$ to $O(nKC)$ for a dataset containing C vectors [175]. Furthermore, the embedding subspace dimension now depends only on the information content K of the dataset, not on its *cardinality* C as in the case of non-sparse signals. In other words, if the signals are K-sparse, the JL lemma holds for $n = O(K \log N)$ [174].

In our implementation, each element $\phi_{i,j}$ of the random projection matrix Φ

is generated independently according to the following simple distribution:

$$\phi_{i,j} = \sqrt{\frac{3}{n}} \begin{cases} +1 & \text{with probability } 1/6 \\ 0 & \text{with probability } 2/3 \\ -1 & \text{with probability } 1/6 \end{cases} \tag{5.2}$$

The mapping given by this matrix satisfies the JL lemma and is more computationally efficient compare to Gaussian distributed random matrices [175]. By using this discrete random matrix in performing random projection, it is able to avoid costly matrix multiplication operations. As a result, this helps to significantly improve the efficiency of the proposed approach.

5.3.3 Support Vector Machines and Support Vector Regressions

Given a set of labeled training data $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$, where $\mathbf{x}_i \in \mathcal{R}^n$ and $y_i \in \{-1, +1\}$, an SVM tries to find a separating hyperplane parameterized by the pair $(\mathbf{w}, b)$ that achieves the maximum margin [19]. The value of $(\mathbf{w}, b)$ is obtained by solving the following constrained optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i \\ \text{subj. to} \quad & y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned} \tag{5.3}$$

where ξ_i are slack variables. The above optimization problem can be solved by the method of Lagrange multipliers. After obtaining the pair $(\mathbf{w}, b)$ from training, the predicted label for a test sample $\mathbf{x}$ is given by:

$$y = \text{sign}(\mathbf{w}\cdot\mathbf{x} + b) \quad . \tag{5.4}$$

In order to handle multi-class classification problems, we can either train SVMs for each pair of classes (AVA) or train classifiers for each class against the rest (OVA). In our implementation, the AVA approach was employed to train the multi-class SVMs.

An extension of SVM for regression problems, called Support Vector Regression (SVR) was proposed by Drucker *et. al.* [20]. In SVR, the regression function $f(\mathbf{x})$ is optimized such that it has most ϵ deviation from the output value $y_i \in \mathcal{R}$ for a training sample $\mathbf{x}_i \in \mathcal{R}^n$, and is as flat as possible at the same time. In other words, we need to solve the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b,\xi^+,\xi^-} \quad & \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}(\xi_i^+ + \xi_i^-) \\ \text{subj. to} \quad & y_i - \mathbf{w}\cdot\mathbf{x}_i - b \leq \epsilon + \xi_i^+ \\ & \mathbf{w}\cdot\mathbf{x}_i + b - y_i \leq \epsilon + \xi_i^- \\ & \xi_i^+, \xi_i^- \geq 0 \end{aligned} \tag{5.5}$$

The predicted value for a test sample $\mathbf{x}$ is obtained as:

$$y = f(\mathbf{x}) = \mathbf{w}\cdot\mathbf{x} + b \quad . \tag{5.6}$$

In order to improve the performance in non-linearly separable cases, SVMs and SVRs with Gaussian kernels (also called RBF kernels) are used in our approach. An RBF kernel has the following form:

$$K_\gamma^{(RBF)}(\mathbf{x},\mathbf{z}) = \exp\left[-\gamma||\mathbf{x}-\mathbf{z}||^2\right] \tag{5.7}$$

where γ is the parameter controlling the width the kernel.

5.3.4 Predicting by Combining Classification and Regression

In the proposed method, rather than just performing a regression on the head pose of an input face image, classification and regression are combined together in order to obtain a more robust estimate. First, the space of possible head pose configurations is divided into a fixed number of bins. Face images whose poses lie in the same bin have the same label and a multi-class SVM is trained for these labels. It can be seen that this multi-class SVM provides a coarse prediction of the head pose. Face images in the same bin are then used to train an SVR in order to refine the pose estimate. As a result, the number of SVRs will be equal to the number of bins (classes). It is clear that this number is decided based on the number of available training images and how fine we want the post estimate to be.

As the SVR method tries to avoid over-fitting by finding a flat curve within a small ϵ margin, a single SVR may not capture irregular curves like the one in Figure 5.4 [155]. The advantage of combining SVC and SVR over a global SVR is that it provides a better approximation to the data distribution by training the SVRs "locally".

5.4 Experiments

5.4.1 Training

The proposed algorithm was trained on 2D images generated from the 3D faces in the USF 3D database [25]. The 2D views were synthesized at different viewing

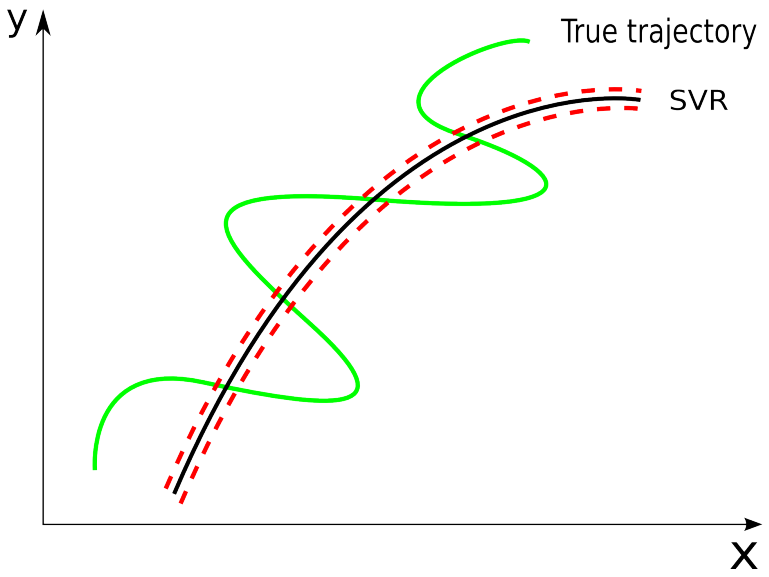


Figure 5.4: An illustration of using SVR to approximate an irregular curve.

angles by rotating the 3D models and projecting into the image plane. Figure 5.5 shows the 2D face images of a person in the database generated at different poses and the visualization of their corresponding dense SIFT descriptors. As the USF database contains the geometry as well as the texture information of the 3D faces, the face images at different illumination conditions can also be generated from the surface normals and albedo using the *Lambert's Cosine Law*:

$$I_{i,j} = \rho_{i,j} \max(\mathbf{n}_{i,j}^T \mathbf{s}, 0) \tag{5.8}$$

where $\mathbf{s}$ is the direction of the light source, $I_{i,j}$, $\mathbf{n}_{i,j}$ and $\rho_{i,j}$ are the image intensity, surface normal and albedo at the pixel (i, j), respectively. This is necessary in order for the method to handle possible illumination variations in the test images.

In our experiments, all training images were scaled to a fixed size of 50×50. The gap between two image grid points is set to 5. As a result, there are 100 $(= 10 \times 10)$ descriptors obtained from each 50×50 image. Each image is padded in

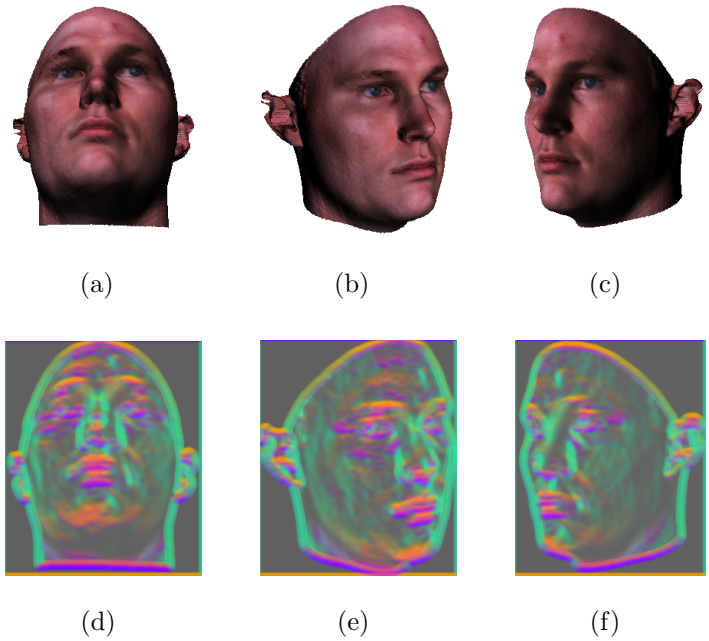
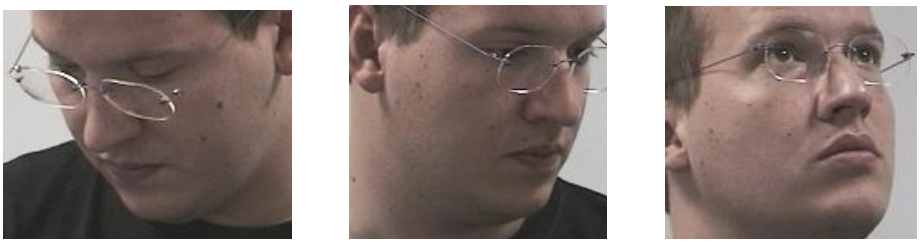


Figure 5.5: First row: face images of a person in the USF 3D database generated at different viewing angles. Second row: visualization of the corresponding dense SIFT descriptors.

order to obtain SIFT descriptors for grid points on the borders. As the dimension of each descriptor is 128, the length of the concatenated feature vector for each resized face image is 12800. Random projection is applied to bring the dimension of the feature vector down to 2000.

5.4.2 Pointing '04 database

In order to evaluate the proposed approach, it was first tested on the Pointing '04 head pose database [48]. This database consists of 15 sets of images. In each set, there are 2 series of 93 images of the same person with varying yaw and pitch angles. The head poses are quantized into nine angles of pitch: $\{-90°, -60°, -30°, -15°,$



(a) (b) (c)

Figure 5.6: Face images of a subject in the Pointing '04 database at different head poses.

$0°, 15°, 30°, 60°, 90°\}$ and thirteen angles of yaw: $\{-90°, -75°, -60°, -45°, -30°, -15°, 0°, 15°, 30°, 45°, 60°, 75°, 90°\}$. Figure 5.6 shows example images of a subject in the Pointing '04 database at different head poses.

The proposed approach is compared with methods presented in [161], [155] and [165]. The results obtained using dense SIFT descriptors with PCA are also included for comparison. The measure of performance used in the comparison is the *Mean Absolute Error* (MAE). It is defined as the average of the absolute errors between the estimated and ground-truth poses. The comparison of the MAEs between different approaches on the Pointing '04 database is shown in Table 5.1. It can be seen from Table 5.1 that the proposed method outperforms all other algorithms used in the comparison.

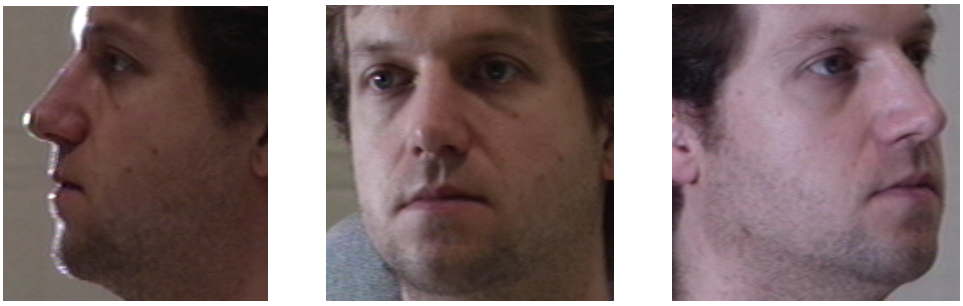
5.4.3 Multi-PIE database

Experiments on the Multi-PIE dataset [3] were also performed in order to better assess the performance of the proposed algorithm. We employed the same

Table 5.1: Comparison of the MAEs between different approaches on the Pointing '04 database.

Method	Yaw Error	Pitch Error
Local-PCA [161]	24.5°	37.6°
Local-LPP [161]	29.2°	40.2°
Local-LDA [161]	19.1°	30.7°
LARR2 [155]	9.23°	7.69°
Kernel PLS [165]	6.56°	6.61°
Dense SIFT + PCA	6.17°	6.42°
Our approach (Dense SIFT + RP)	6.05°	5.84°

experiment setup as in [165]: 2700 face images of 144 subjects, under frontal illumination and varying expressions, were used. There are thirteen discrete yaw angles in the images, varying between $-90°$ and $90°$ with increments of $15°$. Example images from the Multi-PIE database are shown in Figure 5.7. Table 5.2 shows the MAEs obtained using our approach and other methods on the Multi-PIE dataset. The results of the methods based on linear PLS, kernel PLS and Principal Component Regression (PCR) were obtained from [165]. It can be seen from the table that our method based on dense SIFT and PCA was comparable to the kernel PLS and outperformed the algorithms based on linear PLS and PCR. When RP was used instead of PCA for dimension reduction, it further reduced the estimation error from $5.63°$ to $5.11°$.



(a) (b) (c)

Figure 5.7: Face images of a subject in the Multi-PIE database at different head poses.

Table 5.2: Comparison of the MAEs in the yaw angle between different approaches on the Multi-PIE dataset.

	Linear PLS [165]	Kernel PLS [165]	PCR [165]	Dense SIFT + PCA	Dense SIFT + RP
Yaw Error	9.11°	5.31°	11.03°	5.63°	5.11°

5.5 Conclusions

In this chapter, we have presented an automatic method for head pose estimation from a single image. By extracting dense SIFT descriptors from the input image, we obtain a high dimensional feature vector that is robust to noise and illumination variations. The dimension of the feature vector is reduced using RP for efficient processing. A combination of SVM and SVR is used improve the prediction of the head pose. The advantage of the proposed approach is that it does not depend on the extraction of facial features such as the eye corners, nose tip and

mouth corners from the input image. Experimental results on the Pointing '04 and CMU-PIE databases demonstrate the effectiveness of the approach.

Chapter 6: Directions for Future Work

In this chapter, different future directions for exploring the research presented in the dissertation are discussed.

6.1 3D Face Reconstruction

One of the possible future research directions is extending the method for synthesizing frontal faces proposed in Chapter 2 to the task of 3D face reconstruction (Figure 6.1). By dividing the input face image into patches and employing a similar MRF framework, a set of 3D parameters of the patches can be obtained in order to inferred the 3D structure of the input face. If each patch is assumed to be planar as in the Make3D algorithm [176], the 3D parameters can be the 3D locations and orientations of the patches. However, in order to capture the complex geometry of a 3D face, it is better to represent each 2D image patch as the projection of a parametric curved surface in the 3D space.

6.2 Explicitly Synthesize Out-of-Plane Rotations and Expressions

Another future work is to explicitly incorporate other variation factors such as 3D pose and expression into the face recognition approach based on multifactor

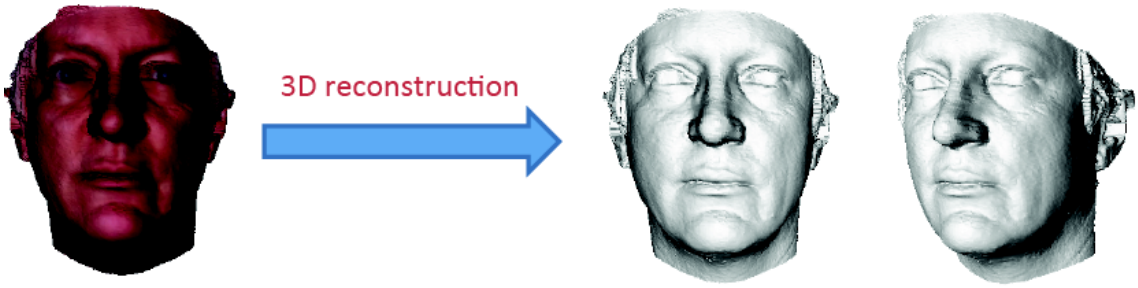


Figure 6.1: Visualization of 3D reconstruction from a 2D face image.

analysis discussed in Chapter 3. As a result, it is necessary to investigate on how to obtain analytical representations of the image space resulting from the variations in head pose or facial expression. It was proved in [177] that the transformed images of a 3D object under all viewing directions form a parametric manifold in a 6-dimensional linear subspace. Although there have been many works on manifold representations of facial expression [178, 179], most of them were based on non-parametric appearance manifolds. It is desirable to obtain an analytical representation of the expression manifold in order to systematically sample face images of a subject at different expressions.

6.3 Simultaneous Feature and Multitask Learning

One drawback of the approach in Chapter 4 is that the classifier for each task is learned separately using the corresponding labels. It may be possible to improve the performance of these classifiers by training them together using multitask learning [180]. Multitask learning is a machine learning technique that allows the learner to use the commonality among the tasks in order to obtain better generalization.

Assume that there are T tasks (linear classification or regression) to be learned. For each task $t \in [1, \ldots, T]$, there are M_t samples. For simplicity, assume that all samples have the same dimension d. Our goal is to learn a common dictionary D, a matrix A containing the sparse codes of all data samples, and a weight matrix W simultaneouly. The dimension of the dictionary D is $d \times k_D$ where k_D is the number of atoms. The vector $\boldsymbol{\alpha}_{ti}$ is the sparse code for the data sample $\mathbf{x}_{ti}$ (sample i in task t). Each $\boldsymbol{\alpha}_{ti}$ is a column of the matrix A of dimension $k_D \times M$ where $M = \sum_{t=1}^{T} M_t$. Column t of W (denoted by $\mathbf{w}_t$) is the classifier (or regressor) for task t. The dimension of W is $k_D \times T$.

The optimal values of D, A, and W can be obtained by solving the following minimization problem:

$$\{D^*, A^*, W^*\} = \underset{D,A,W}{\operatorname{argmin}} \sum_{t=1}^{T} \Big\{ \sum_{i=1}^{M_t} \big\{ ||\mathbf{x}_{ti} - D\boldsymbol{\alpha}_{ti}||_2^2 + \lambda_1 ||\boldsymbol{\alpha}_{ti}||_1 + \gamma_1 C(y_{ti}, \mathbf{w}_t \cdot \boldsymbol{\alpha}_{ti}) \big\} + \lambda_2 ||\mathbf{w}_t||_2^2 \Big\} \tag{6.1}$$

where y_{ti} is the training label (or output) for sample $\mathbf{x}_{ti}$. $C[y_{ti}(\mathbf{w}_t \cdot \boldsymbol{\alpha}_{ti})]$ is the loss function. In the case of classification, we employ the logistic loss $C(y, \hat{y}) = \log(1 + \exp(-y\hat{y}))$ as it is similar to the hinge loss in SVM as well as differentiable. For regression tasks, the squared loss $C(y, \hat{y}) = ||y - \hat{y}||^2$ is used.

Bibliography

[1] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The FERET Evaluation Methodology for Face-Recognition. *IEEE Trans. PAMI*, 22(10):1090–1104, 2000.

[2] T. Sim, S. Baker, and M. Bsat. The CMU Pose, Illumination, and Expression Database. *IEEE Trans. PAMI*, 25(12):1615–1618, 2003.

[3] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-PIE Dataset. In *Proc. FG*, pages 807–813, 2008.

[4] H. Jia and A. M. Martinez. Support Vector Machines in Face Recognition with Occlusions. In *Proc. CVPR*, pages 136–141, 2009.

[5] Y. Chen, V. M Patel, P. J. Phillips, and R. Chellappa. Dictionary-Based Face Recognition from Video. In *Proc. ECCV*, pages 766–779, 2012.

[6] S. Biswas, G. Aggarwal, and R. Chellappa. Robust Estimation of Albedo for Illumination-Invariant Matching and Shape Recovery. *IEEE Trans. PAMI*, 31(5):884–899, 2009.

[7] A. M. Martinez and R. Benavente. The AR Face Database. *CVC Technical Report*, 24, 1998.

[8] P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In *Proc. CVPR*, pages 511–518, 2001.

[9] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Robust Discriminative Response Map Fitting with Constrained Local Models. In *Proc. CVPR*, pages 3444–3451, 2013.

[10] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face Recognition: A Literature Survey. *ACM Computing Surveys*, 35(4):399–458, 2003.

[11] D. J. Lowe. Distinctive Image Features From Scale-Invariant Keypoints. *IJCV*, 60(2):91–110, 2004.

[12] T. Ahonen, A. Hadid, and M. Pietikainen. Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE Trans. PAMI*, 28(12):2037–2041, 2006.

[13] E. Murphy-Chutorian and M. M. Trivedi. Head Pose Estimation in Computer Vision: A Survey. *IEEE Trans. PAMI*, 31(4):607–626, 2009.

[14] A. B. Ashraf, S. Lucey, and T. Chen. Fast Image Alignment in the Fourier Domain. In *Proc. CVPR*, pages 2480–2487, 2010.

[15] N. Komodakis and G. Tziritas. Image Completion using Efficient Belief Propagation via Priority Scheduling and Dynamic Pruning. *IEEE Trans. Image Proc.*, 16(11):2649–2661, 2007.

[16] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting Visual Category Models to New Domains. In *Proc. ECCV*, pages 213–226, 2010.

[17] B. Kulis, K. Saenko, and T. Darrell. What You Saw is Not What You Get: Domain Adaptation using Asymmetric Kernel Transforms. In *Proc. CVPR*, pages 1785–1792, 2011.

[18] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. SIFT Flow: Dense Correspondence across Different Scenes. In *Proc. ECCV*, pages 28–42, 2008.

[19] V. N. Vapnik. *Statistical Learning Theory.* John Wiley, 1998.

[20] H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. N. Vapnik. Support Vector Regression Machines. In *NIPS*, pages 155–161, 1996.

[21] X. Zhang and Y. Gao. Face Recognition Across Pose: A Review. *PR*, 42(11):2876–2896, 2009.

[22] M. Turk and A. Pentland. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3:72–86, 1991.

[23] K. Etemad and R. Chellappa. Discriminant Analysis for Recognition of Human Face Images. *Journal of Optical Society America A*, 14:1724–1733, 1997.

[24] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using Class Specific Linear Projection. *IEEE Trans. PAMI*, 19:711–720, 1997.

[25] V. Blanz and T. Vetter. A Morphable Model for the Synthesis of 3D Faces. In *SIGGRAPH*, pages 187–194, 1999.

[26] S. Biswas and R. Chellappa. Pose-Robust Albedo Estimation from a Single Image. In *Proc. CVPR*, pages 2683–2690, 2010.

[27] A. Asthana, T. Marks, M. Jones, K. Tieu, and R. MV. Fully Automatic Pose-Invariant Face Recognition via 3D Pose Normalization. In *Proc. ICCV*, pages 937–944, 2011.

[28] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active Appearance Models. *IEEE Trans. PAMI*, 23(6):681–685, 2001.

[29] R. Gross, I. Matthews, and S. Baker. Appearance-Based Face Recognition and Light-Fields. *IEEE Trans. PAMI*, 26(4):449–465, 2004.

[30] S. J. D. Prince, J. Elder, J. H. Warrell, and F. M. Felisberti. Tied Factor Analysis for Face Recognition across Large Pose Differences. *IEEE Trans. PAMI*, 30(6):970–984, 2008.

[31] C. D. Castillo and D. W. Jacobs. Using Stereo Matching with General Epopolar Geometry for 2D Face Recognition across Pose. *IEEE Trans. PAMI*, 31(12):2298–2304, 2009.

[32] M. S. Sarfraz and O. Hellwich. Probabilistic Learning for Fully Automatic Face Recognition across Pose. *Image and Vision Computing*, 28:744–753, 2010.

[33] T. Kanade and A. Yamada. Multi-Subregion Based Probabilistic Approach toward Pose-Invariant Face Recognition. In *Proc. Symp. CIRA*, pages 954–959, 2005.

[34] A. B. Ashraf, S. Lucey, and T. Chen. Learning Patch Correspondences for Improved Viewpoint Invariant Face Recognition. In *Proc. CVPR*, 2008.

[35] X. Chai, S. Shan, X. Chen, and W. Gao. Locally Linear Regression for Pose-Invariant Face Recognition. *IEEE Trans. Image Proc.*, 16(7):1716–1725, 2007.

[36] A. Li, S. Shan, X. Chen, and W. Gao. Maximizing Intra-Individual Correlations for Face Recognition Across Pose Differences. In *Proc. CVPR*, pages 605–611, 2009.

[37] S. R. Arashloo and J. Kittler. Pose-Invariant Face Matching using MRF Energy Minimization Framework. In *Proc. EMMCVPR*, pages 56–69, 2009.

[38] S. Liao and A. C. S. Chung. A Novel Markov Random Field Based Deformable Model for Face Recognition. In *Proc. CVPR*, pages 2675–2682, 2010.

[39] S. Baker, R. Gross, I. Matthews, and T. Ishikawa. Lucas-Kanade 20 Years On: A Unifying Framework: Part2. Technical Report CMU-RI-TR-03-01, Robotics Institute, February 2003.

[40] B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proc. IJCAI* , pages 674–679, 1981.

[41] G. D. Hager and P. N. Belhumeur. Efficient Region Tracking with Parametric Models of Geometry and Illumination. *IEEE Trans. PAMI*, 20(10):1025–1039, 1998.

[42] C. Liu and H. Wechsler. Gabor Feature Based Classification using the Enhanced Fisher Linear Discriminant Model for Face Recognition. *IEEE Trans. Image Proc.*, 11:467–476, 2002.

[43] A. V. Oppenheim and A. S. Willsky, editors. *Signals & Systems*. Prentice Hall, 2nd edition, 1996.

[44] S. Baker and I. Matthews. Equivalence and Efficiency of Image Alignment Algorithms. In *Proc. CVPR*, June 2001.

[45] J. Pearl, editor. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, 1988.

[46] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding Belief Propagation and Its Generalizations. *Exploring Artificial Intelligence in the New Millenium*, pages 239–269, 2003.

[47] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Belief Propagation for Early Vision. *IJCV*, 70(1):41–54, 2006.

[48] N. Gourier, D. Hall, and J. Crowley. Estimating Face Orientation from Robust Detection of Salient Facial Features. In *Proc. ICPR Pointing '04 Workshop*, 2004.

[49] D. Little, S. Krishna, J. Black, and S. Panchanathan. A Methodology for Evaluating Robustness of Face Recognition Algorithms with respect to Changes in Pose and Illumination Angle. In *Proc. ICASSP*, 2005.

[50] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[51] Y. Bouguet. Pyramidal Implementation of the Lucas Kanade Feature Tracker. *OpenCV Document, Intel Microprocessor Research Labs*, 2000.

[52] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang. Local Gabor Binary Pattern Histogram Sequence (LGBPHS): A Novel Non-Statistical Model fro Face Representation and Recognition. In *Proc. ICCV*, pages 786–791, 2005.

[53] H. Gao, H. K. Ekenel, and R. Stiefelhagen. Pose Normalization for Local Appearance-based Face Recognition. In *Proc. Intl. Conf. on Advances in Biometrics*, pages 32–41, 2009.

[54] P. Vageeswaran, K. Mitra, and R. Chellappa. Blur and Illumination Robust Face Recognition via Set-Theoretic Characterization. *IEEE Trans. Image Processing*, 22(4):1362–1372, 2013.

[55] K. C. Lee, J. Ho, and D. J. Kriegman. Acquiring Linear Subspaces for Face Recognition under Variable Lighting. *IEEE Trans. PAMI*, 27(5):684–698, 2005.

[56] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1):151–175, 2010.

[57] R. Gopalan, R. Li, and R. Chellappa. Domain Adaptation for Object Recognition: An Unsupervised Approach. In *Proc. ICCV*, pages 999–1006, 2011.

[58] Y. M. Lui. Advances in Matrix Manifolds for Computer Vision. *Imag. Vis. Comp.*, 30:380–388, 2012.

[59] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A Global Geometric Framework for Non-Linear Dimensionality Reduction. *Science*, pages 2319–2323, 2000.

[60] S. T. Roweis and L. K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, pages 2323–2326, 2000.

[61] M. A. O. Vasilescu and D. Terzopoulos. Multilinear Analysis of Image Ensembles. In *Proc. ECCV*, pages 447–460, 2002.

[62] M. A. O. Vasilescu and D. Terzopoulos. Multilinear Projection for Appearance-based Recognition n the Tensor Framework. In *Proc. ICCV*, pages 1–8, 2007.

[63] Y. M. Lui, J. R. Beveridge, and M. Kirby. Action Classifications on Product Manifolds. In *Proc. CVPR*, pages 833–839, 2010.

[64] S. W. Park and M. Savvides. The Multifactor Extension of Grassmann Manifolds for Face Recognition. In *Proc. FG*, pages 464–469, 2011.

[65] S. W. Park and M. Savvides. Multifactor Analysis based on Factor-Dependent Geometry. In *Proc. CVPR*, pages 2817–2824, 2011.

[66] Hal Daumé III and Daniel Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26(1):101–126, 2006.

[67] J. Hoffman, B. Kulis, T. Darrell, and K. Saenko. Discovering Latent Domains for Multisource Domain Adaptation. In *Proc. ECCV*, pages 702–715, 2012.

[68] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic Flow Kernel for Unsupervised Domain Adaptation. In *Proc. CVPR*, pages 2066–2073, 2012.

[69] J. Zheng, M.-Y. Liu, R. Chellappa, and J. Phillips. A Grassmann Manifold-Based Domain Adaptation Approach. In *Proc. ICPR*, pages 2095–2099, 2012.

[70] Y. Shi and F. Sha. Information-Theoretical Learning of Discriminative clusters for Unsupervised Domain Adaptation. In *Proc. ICML*, 2012.

[71] I.-H. Jhuo, D. Liu, D.T. Lee, and S.-F. Chang. Robust Visual Domain Adaptation with Low-Rank Reconstruction. In *Proc. CVPR*, pages 2168–2175, 2012.

[72] Qiang Qiu, VishalM. Patel, Pavan Turaga, and Rama Chellappa. Domain Adaptive Dictionary Learning. In *Proc. ECCV*, pages 631–645, 2012.

[73] J. Ni, Q. Qiu, and R. Chellappa. Subspace Interpolation via Dictionary Learning for Unsupervised Domain Adaptation. In *Proc. CVPR*, 2013.

[74] S. Shekhar, V. Patel, H. Nguyen, and R. Chellappa. Generalized Domain-Adaptive Dictionaries. In *Proc. CVPR*, 2013.

[75] J. Yang, R. Yan, and A. Hauptmann. Cross-Domain Video Concept Detection using Adaptive SVMs. In *Proc. ACM MM*, pages 188–197, 2007.

[76] L. Duan, I.W. Tsang, D. Xu, and T.-S. Chua. Domain Adaptation from Multiple Sources via Auxiliary Classifiers. In *Proc. ICML*, pages 289–296, 2009.

[77] L. Duan, I.W. Tsang, D. Xu, and T.-S. Chua. Domain Transfer Multiple Kernel Learning. *IEEE Trans. PAMI*, 34(3):465–479, 2012.

[78] A. M. Martinez. Recognizing Imprecisely Localized, Partially Occluded, and Expression Variant Faces from a Single Sample per Class. *IEEE Trans. PAMI*, 24(6):748–763, 2009.

[79] J. Liu, S. Chen, Z. Zhou, and X. Tan. Single Image Subspace for Face Recognition. In *Proc. AMFG*, pages 205–219, 2007.

[80] Y. M. Lui and J. R. Beveridge. Grassmann Registration Manifolds for Face Recognition. In *Proc. ECCV*, volume 2, pages 44–57, 2008.

[81] O. Arandjelović. Unfolding a Face: from Singular to Manifold. In *Proc. ACCV*, pages 203–213, 2009.

[82] J. Lu, Y. P. Tan, and G. Wang. Discriminative Multi-Manifold Analysis for Face Recognition from a Single Training Sample per Person. In *Proc. ICCV*, pages 1943–1950, 2011.

[83] I. T. Joliffe. *Principal Component Analysis*. Springer-Verlag, 1986.

[84] Y. Li, Y. Du, and X. Lin. Kernel-based Multifactor Analysis for Image Synthesis and Recognition. In *Proc. ICCV*, pages 114–119, 2005.

[85] S. W. Park and M. Savvides. An Extension of Multifactor Analysis for Face Recognition based on Submanifold Learning. In *Proc. CVPR*, pages 2645–2652, 2010.

[86] R. Gopalan, S. Taheri, P. Turaga, and R. Chellappa. A Blur-Robust Descriptor with Applications to Face Recognition. *IEEE Trans. PAMI*, 34(6):1220–1226, 2012.

[87] A. Edelman, T. A. Arias, and S. T. Smith. The Geometry of Algorithms with Orthogonality Constraints. *SIAM J. Matrix Analysis and Applications*, 20:303–353, 1999.

[88] A. Björck and G. H. Golub. Numerical Methods for Computing Angles between Linear Subspaces. *Mathematics of Computations*, pages 579–594, 1973.

[89] J. Hamm and D. Lee. Grassmann Discriminant Analysis: A Unifying View on Subspace-based Learning. In *Proc. ICML*, pages 376–383, 2008.

[90] J. M. Lee. *Introduction to Topological Manifolds*. Springer, 2 edition, 2010.

[91] E. Begelfor and M. Werman. Affine Invariance Revisited. In *Proc. CVPR*, pages 2087–2094, 2006.

[92] G. Shakhnarovich, J. W. Fisher, and T. Darell. Face Recognition from Long Term Observations. In *Proc. ECCV*, pages 851–868, 2002.

[93] O. Arandjelović, G. Shakhnarovich, J. Fisher, R. Cipolla, and T. Darrell. Face Recognition with Image Sets using Manifold Density Divergence. In *Proc. CVPR*, pages 581–588, 2005.

[94] M. J. Brooks and B. K. P. Horn. Shape and Source from Shading. In *Proc. IJAI*, pages 932–936, 1985.

[95] R. Basri and D. W. Jacobs. Lambertian Reflectance and Linear Subspaces. *IEEE Trans. PAMI*, 25(2):218–233, 2003.

[96] K. C. Lee, J. Ho, M. H. Yang, and D. J. Kriegman. Visual Tracking and Recognition using Probabilistic Appearance Manifolds. *CVIU*, 99(3):303–331, 2005.

[97] V. Ojansivu and J. Heikkilä. Blur Insensitive Texture Classification using Local Phase Quantization. In *Proc. ICISP*, pages 236–243, 2008.

[98] M. Nishiyama, A. Hadid, H. Takeshima, J. Shotton, T. Kozakaya, and O. Yamaguchi. Facial Deblur Inference using Subspace Analysis for Recognition of Blurred Faces. *IEEE Trans. PAMI*, 33(4):838–845, 2011.

[99] H. Jia and A. M. Martinez. Face Recognition with Occlusions in the Training and Testing Sets. In *Proc. FG*, pages 1–6, 2008.

[100] T. K. Kim, J. Kittler, and R. Cipolla. Discriminative Learning and Recognition of Image Set Classes using Canonical Correlations. *IEEE Trans. PAMI*, 29(6):1005–1018, 2007.

[101] R. Wang and X. Chen. Manifold Discriminant Analysis. In *Proc. CVPR*, pages 429–436, 2009.

[102] H. Cevikalp and B. Triggs. Face Recognition Based on Image Sets. In *Proc. CVPR*, pages 2567–2573, 2010.

[103] Y. Hu, A. S. Mian, and R. Owens. Sparse Approximated Nearest Points for Image Set Classification. In *Proc. CVPR*, pages 27–40, 2011.

[104] C. Sanderson and B. C. Lovell. Multi-Region Probabilistic Histograms for Robust and Scalable Identity Inference. In *Proc. ICB*, pages 199–208, 2009.

[105] E. Nowak and F. Jurie. Learning Visual Similarities Measures for Comparing Never Seen Objects. In *Proc. CVPR*, 2007.

[106] L. Wolf, T. Hassner, and Y. Taigman. Descriptor Based Methods in the Wild. In *Faces in Real-Life Images Workshop in ECCV*, 2008.

[107] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Trans. PAMI*, 24(7):971–987, 2002.

[108] N. Pinto, J. J. Dicarlo, and D. D. Cox. How Far Can You Get With a Modern Face Recognition Test Set Using Only Simple Features. In *Proc. CVPR*, 2009.

[109] G. B. Huang, V. Jain, and E. Learned-Miller. Unsupervised Joint Alignment of Complex Images. In *Proc. ICCV*, 2007.

[110] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang. Probabilistic Elastic Matching for Pose Invariant Face Recognition. In *Proc. CVPR*, 2013.

[111] M. Liu, S. Li, S. Shan, and X. Chen. AU-Aware Deep Networks for Facial Expression Recognition. In *Proc. FG*, 2013.

[112] A. Coates and A. Y. Ng. The Importance of Encoding versus Training with Sparse Coding and Vector Quantization. In *Proc. IMCL*, pages 921–928, 2011.

[113] L. Bo, X. Ren, and D. Fox. Hierarchical Matching Pursuit for Image Classification: Architecture and Fast Algorithms. In *Proc. NIPS*, pages 2115–2123, 2011.

[114] R. Socher, C. C. Lin, A. Ng., and C. Manning. Parsing Natural Scenes and Natural Language with Recursive Neural Networks. In *Proc. ICML*, pages 129–136, 2011.

[115] X. Zhu and D. Ramanan. Face Detection, Pose Estimation, and Landmark Localization in the Wild. In *Proc. CVPR*, pages 2879–2886, 2012.

[116] Y. Freund and R.E. Schapire. A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting. *J. Comput. System Sci.*, 55(1):119–139, 1997.

[117] G. Shakhanarovich, P. A. Viola, and B. Moghaddam. A Unified Learning Framework for Real Time Face Detection and Classification. In *Proc. FG*, pages 14–21, 2002.

[118] J. Whitehill and C. W. Omlin. Haar Features for FACS AU Recognition. In *Proc. FG*, pages 97–101, 2006.

[119] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Recognizing Facial Expression: Machine Learning and Application to Spontaneous Behavious. In *Proc. CVPR*, pages 568–573, 2005.

[120] Y. Ma, T. Xiong, Y. Zou, and K. Wang. Person-Specific Age Estimation under Ranking Framework. In *Proc. ICMR*, 2011.

[121] C. Shan, S. Gong, and P. W. McOwan. Facial Expression Recognition based on Local Binary Patterns: A Comprehensive Study. *Imag. Vis. Comp.*, 27:803–816, 2009.

[122] Z. Yang H. Ai. Demographic Classification with Local Binary Patterns. In *Proc. ICB*, pages 464–473, 2007.

[123] U. Tarig, K. Lin, Z. Li, X. Zhou, Z. Wang, V. Le, T. Huang, X. Lv, and T. Han. Emotion Recognition from an Ensemble of Features. In *Proc. FG*, pages 872–877, 2011.

[124] C. Shan. Learning Local Features for Age Estimation on Real-life Faces. In *Proc. MPVA*, pages 23–28, 2010.

[125] C. Shan. Learning Local Binary Patterns for Gender Classification on Real-World Face Images. *PRL*, 33(4):431–437, 2012.

[126] P. Turaga, S. Biswas, and R. Chellappa. The Role of Geometry in Age Estimation. In *Proc. ICASSP*, pages 946–949, 2010.

[127] S. Taheri, P. Turaga, and R. Chellappa. Towards View-Invariant Expression Analysis using Analytic Shape Manifolds. In *Proc. FG*, pages 306–313, 2011.

[128] B. Manjunath and R. Chellappa. A Unified Approach to Boundary Perception: Edges, Textures, and Illusory Contours. *IEEE Trans. Neural Networks*, 4(1):96–108, 1993.

[129] G. Hinton, S. Osindero, and Y. Teh. A Fast Learning Algorithm for Deep Belief Nets. *Neur. Comp.*, 18(7):1527–1554, 2006.

[130] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol. Extracting and Composing Robust Features with Denoising Autoencoders. In *Proc. ICML*, 2008.

[131] H. Lee, R. Grosse, R. Ranganath, and A. Ng. Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. In *Proc. ICML*, 2009.

[132] K. Yu, Y. Lin, and J. Lafferty. Learning Image Representations from the Pixel Level via Hierarchical Sparse Coding. In *Proc. CVPR*, pages 1713–1720, 2013.

[133] J. M Saragih, S. Lucey, and J. F. Cohn. Deformable Model Fitting by Regularized Landmark Mean-Shift. *IJCV*, 91(2):200–215, 2011.

[134] M. Elad. *Sparse and Redundant Representations*. Springer, 2010.

[135] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online Dictionary Learning for Sparse Coding. In *Proc. ICML*, pages 689–696, 2009.

[136] Y. C. Pati, R. Rezalifar, and P. S. Krishnaprasad. Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition. In *Proc. Asilomar Conf. on Signals, Systems and Computers*, pages 40–44, 1993.

[137] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least Angle Regression. *The Annals of Statistics*, 32(2):407–499, 2004.

[138] R. Rubinstein, M. Zibulevski, and M. Elad. Efficient Implementation of the K-SVD Algorithm using Batch Orthogonal Matching Pursuit. Technical report, CS Technion, 2008.

[139] R. Rigamonti, M. A. Brown, and V. Lepetit. Are Sparse Representation Really Relevant for Image Classification? In *Proc. CVPR*, pages 1545–1552, 2011.

[140] Y-L. Boureau, J. Ponce, and Y. LeCun. A Theoretical Analysis of Feature Pooling in Visual Recognition. In *Proc. ICML*, 2010.

[141] P. Lucey, J. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The Extended Cohn-Kanade Dataset (CK+): A Complete Dataset for Action Unit and Emotion-Specified Expression. In *CVPR Workshop*, 2010.

[142] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, and et al. Challenges in Representation Learning: A Report on Three Machine Learning Contests. In *Neural Information Processing*, pages 117–124, 2013.

[143] A. Gallagher and T. Chen. Understanding Images of Groups of People. In *Proc. CVPR*, pages 256–263, 2009.

[144] L. Zhong, Q. Liu, P. Yang, B. Liu, J. Huang, and D. N. Metaxas. Learning Active Facial Patches for Expression Analysis. In *Proc. CVPR*, pages 2562–2569, 2012.

[145] R. T. Ionescu, M. Popescu, and C. Grozea. Local Learning to Improve Bag of Visual Words Model for Facial Expression Recognition. In *ICML Workshop on Challenges in Representation Learning*, 2013.

[146] Yichuan Tang. Deep Learning using Linear Support Vector Machines. In *ICML Workshop on Challenges in Representation Learning*, 2013.

[147] I. Ramírez, P. Sprechmann, and G. Sapiro. Classification and Clustering via Dictionary Learning with Structured Incoherence and Shared Features. In *Proc. CVPR*, pages 3501–3508, 2011.

[148] M. Yang, L. Zhang, X. Feng, and D. Zhang. Fisher Discrimination Dictionary Learning for Sparse Representation. In *Proc. ICCV*, pages 543–550, 2011.

[149] C. Li, Q. Liu, J. Liu, and H. Lu. Learning Ordinal Discriminative Features for Age Estimation. In *Proc. CVPR*, pages 2570–2577, 2012.

[150] K.-Y. Chang, C. S. Chen, and Y.P. Hung. Ordinal Hyperplane Ranker with Cost Sensitivities for Age Estimation. In *Proc. CVPR*, pages 585–592, 2011.

[151] D. Huang, M. Storer, F. De la Torre, and H. Bischof. Supervised Local Subspace Learning for Continuous Head Pose Estimation. In *Proc. CVPR*, pages 2921–2928, 2011.

[152] S. R. Langton, H. Honeyman, and E. Tessler. The Influence of Head Contour and Nose Angle on the Perception of Eye-Gaze Direction. *Percep Psychophys*, 66(5):752–771, 2004.

[153] Y. Li, S. Gong, and H. Liddell. Support Vector Regression and Classification Based Multi-View Face Detection and Recognition. In *Proc. FG*, pages 300–305, 2000.

[154] E. Murphy-Chutorian and M. M. Trivedi. Head Pose Estimation and Augmented Reality Tracking. *IEEE Trans. Intel. Trans. Sys.*, 11:300–311, 2010.

[155] G. Guo, Y. Fu, C. R. Dyer, and T. S. Huang. Head Pose Estimation: Classification or Regression. In *Proc. ICPR*, pages 1–4, 2008.

[156] S. Niyogi and W. T. Freeman. Example-based Head Tracking. In *Proc. FG*, pages 374–378, 1996.

[157] D. J. Beymer. Face Recognition under Varying Pose. In *Proc. CVPR*, pages 756–761, 1994.

[158] J. Sherrah, S. Gong, and E. J. Ong. Face Distributions in Similarity Space under Varying Head Pose. *Image and Vision Computing*, 19(12):807–819, 2001.

[159] J. Huang, X. Shao, and H. Weschsler. Face Pose Discrimination using Support Vector Machines. In *Proc. ICPR*, pages 154–156, 1998.

[160] S. Z. Li, Q. Fu, L. Gu, B. Scholkopf, Y. Cheng, and H. Zhang. Kernel Machine Based Learning for Multi-View Face Detection and Pose Estimation. In *Proc. CVPR*, pages 674–679, 2001.

[161] Z. Li, Y. Fu, J. Yuan, T. S. Huang, and Y. Wu. Query Driven Localized Linear Discriminant Models for Head Pose Estimation. In *Proc. ICME*, pages 1810–1813, 2007.

[162] A. Ranganathan and M. H. Yang. Online Sparse Matrix Gaussian Process Regression and Vision Applications. In *Proc. ECCV*, pages 468–482, 2008.

[163] R. Rae and H. J. Ritter. Recognition of Human Head Orientation based on Artificial Neural Networks. *IEEE Trans. Nerual Networks*, 9(2):257–265, 1998.

[164] H. A. Rowley, S. Baluja, and T. Kanade. Rotation Invariant Neural Network-Based Face Detection. In *Proc. CVPR*, pages 38–44, 1998.

[165] M. A. Haj, J. Gonzalez, and L. S. Davis. On Partial Least Squares in Head Pose Estimation: How to Simultaneously Deal with Misalignment. In *Proc. CVPR*, pages 2602–2609, 2012.

[166] S. Srinivasan and K. L. Boyer. Head Pose Estimation using View Based Eigenspaces. In *Proc. ICPR*, pages 302–305, 2002.

[167] B. Raytchev, I. Yoda, and K. Sakaue. Head Pose Estimation by Nonlinear Manifold Learning. In *Proc. ICPR*, pages 462–466, 2004.

[168] V. N Balasubramanian, Y. Jieping, and S. Panchanathan. Biased Manifold Embedding: A Framework for Person-independent Head Pose Estimation. In *Proc. CVPR*, pages 1–7, 2007.

[169] A. H. Gee and R. Cipolla. Determining the Gaze of Faces in Images. *Image and Vision Computing*, 12(10):639–647, 1994.

[170] T. Horprasert, Y. Yacoob, and L. S. Davis. Computing 3-D Head Orientation from a Monocular Image Sequence. In *Proc. FG*, pages 242–247, 1996.

[171] G. Wang and E. Sung. EM Enhancement of 3D Head Pose Estimated by Point at Infinity. *Img. Vis. Comp.*, 25(12):1864–1874, 2007.

[172] J. G. Wang, J. Li, W. Y. Yaw, and E. Sung. Boosting Dense SIFT Descriptors and Shape Contexts of Face Images for Gender Recognition. In *Proc. CVPR*, pages 96–102, 2010.

[173] W. Johnson and J. Lindenstrauss. Extensions of Lipschitz Mappings into a Hilbert Space. In *Proc. Modern Anal. and Prob.*, pages 189–206, 1984.

[174] C. Hedge, M. A. Davenport, M. Wakinm, and R. G. Baraniuk. Efficient Machine Learning using Random Projections. In *NIPS Workshop on Efficient Machine Learning*, 2007.

[175] E. Bingham and H. Mannila. Random Projection in Dimensionality Reduction: Applications to Image and Text Data. In *Proc. ACM SIGKDD*, pages 245–250, 2001.

[176] A. Saxena, M. Sun, and A. Y. Ng. Make3D: Learning 3D Scene Structure from a Single Still Image. *IEEE Trans. PAMI*, 31(5):824–840, 2009.

[177] X. Zhang, Y. Gao, and T. Caelli. Parametric Manifold of an Object under Different Viewing Directions. In *Proc. ECCV*, pages 186–199, 2012.

[178] C. Shan, S. Gong, and P. W. Mcowan. Appearance Manifold of Facial Expression. In *Proc. ICCV Workshop on HCI*, 2005.

[179] Y. Chang, C. Hu, R. Feris, and M. Turk. Manifold Based Analysis of Facial Expression. *Image and Vision Computing*, 24(6):605–614, 2006.

[180] R. Caruana. Multi-task Learning. *Machine Learning*, 28(1):41–75, 1997.