CROSS-POSE FACIAL EXPRESSION RECOGNITION

by

Fatma Güney

B.S., Computer Engineering, Bilkent University, 2010

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2012

CROSS-POSE FACIAL EXPRESSION RECOGNITION

APPROVED BY:

Prof. Lale Akarun (Thesis Supervisor)	.....
Assist. Prof. Hazım K. Ekenel (Thesis Co-supervisor)	.....
Assist. Prof. Hatice Köse Bağcı	.....
Assist. Prof. Arzucan Özgür	.....
Assoc. Prof. Zehra Çataltepe	.....

DATE OF APPROVAL: 12.09.2012

ACKNOWLEDGEMENTS

I would like to thank my thesis supervisors Prof. Lale Akarun and Assist. Prof. Hazım Kemal Ekenel for their insight, guidance, and support. I am grateful to Assoc. Prof. Zehra Çataltepe, Assist. Prof. Hatice Köse Bağcı, and Assist. Prof. Arzucan Özgür for participating in my thesis jury, their criticism and valuable feedback.

I would like to offer my gratitude to the Scientific and Technological Research Council of Turkey (TÜBİTAK) for supporting me with National Scholarship Programme for M.Sc. Students.

I would like to thank my friends Duygu Sinem Yıldırım, N. Çağrı Kılıboz, Belma Engin, Kaan Bekmezci, and Safiye Çelik for always being there to have fun with. I am especially grateful to my dear friend, Beyza Ermiş for alleviating the burden of this painful process. I would also like to thank the members of MediaLab and PILAB for providing an enjoyable working environment.

I would like to express my deepest gratitude to my parents Şaziye and Beytullah Güney, and my sister Büşra Güney for their endless love, support and patience throughout this process.

I am specifically grateful to N. Murat Arar. I honestly could not bear the burden without his support and motivation.

ABSTRACT

CROSS-POSE FACIAL EXPRESSION RECOGNITION

Automatic facial expression recognition is a popular research topic due to its interesting applications in a wide variety of areas. The existing studies have achieved high accuracies in various formulations of the same problem. One direction which is not fully explored is multi-view facial expression recognition. Variations caused by different poses impose extra burden on the task of recognizing expressions, which is already a difficult problem due to large differences across subjects. In this thesis, we present a method to recognize six prototypic facial expressions of an individual across different pose angles. We use Partial Least Squares (PLS) to map the expressions from different poses into a common subspace, in which correlation between them is maximized. Recently, PLS has been successfully used for pose invariant face recognition problem. We show that, PLS can be effectively used for facial expression recognition across poses by training on coupled expressions of the same identity from two different poses. This way of training lets the learned bases model the differences between expressions of different poses by excluding the effect of the identity. We first align the faces and then extract block features around two eyes and the mouth on the aligned image. We experiment with Gabor filters and direct intensity values for local face representation. We demonstrate that two representations perform similarly in case frontal is the input pose, but Gabor representation outperforms intensity representation for other pose pairs. We also perform a detailed analysis of the parameters used in the experiments to show their effects on the results and to find the optimal ones for the expression recognition problem.

ÖZET

FARKLI BAKIŞ AÇILARI ARASI YÜZ İFADESİ TANIMA

Yüz ifadelerinin otomatik olarak tanınması, geniş kullanım alanlarına bağlı olarak oldukça populer bir araştırma konusudur. Var olan çalışmalar, bu problemin farklı türlerinde oldukça yüksek başarı oranları elde ettiler. Bu problemin, üzerinde daha az çalışılmış bir alanı da çoklu açılardan yüz ifadesi tanımadır. Farklı bakış açıları, farklı kişilerden kaynaklanan değişikliklerden dolayı zaten zor olan ifade tanıma problemini daha da zorlaştırır. Bu çalışmada, bir kişinin farklı bakış açılarından altı temel yüz ifadesini tanımak için bir yöntem öneriyoruz. Farklı bakış açılarından yüz ifadelerini, aralarındaki korelasyonun en yüksek olduğu ortak bir alt uzaya atmak için Kısmi En Az Kare Farkı yöntemini kullanıyoruz. Son zamanlarda, KEAKF yöntemi, bakış açısından bağımsız yüz tanıma problemi için başarılı bir şekilde kullanıldı. Eğitimde, bir insanın farklı açılardan yüz ifadeleri arasında bir ilişki kurulması yoluyla, aynı yöntemin, yüz ifadesi tanıma problemi için de başarılı bir şekilde kullanılabileceğini gösteriyoruz. Bu tür bir eğitim, kişisel farklardan bağımsız bir şekilde bakış açısı farklarını modeller. Yüz imgelerini önce hizalama adımından geçiririz, daha sonra hizalanmış yüzler üzerinde, gözler ve ağızdan yerel bloklar halinde öznitelikler çıkarırız. Öznitelik olarak, Gabor öznitelikleri ve piksel değerlerini kullandık. Ön yüz girdi bakış açısı olarak kullanıldığında, Gabor ve piksel değerlerinin yakın sonuçlar ürettiğini, ama diğer bakıç açısı ikilileri için Gabor özniteliklerinin daha iyi sonuçlar verdiğini deneylerimizde gösterdik. Ayrıca, kullanılan parametrelerin sonuçlar üzerindeki etkisini göstermek ve en iyi değerlerini bulmak için, parametrelerin detaylı analizlerini içeren deneyler yaptık.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

b	Slope of the regression line
$\hat{b}$	Estimated b
b_0	Intercept of the regression line
$\hat{b_0}$	Estimated b_0
b_j	Regression coefficient for the jth explanatory variable
$\boldsymbol{b}$	Vector of regression coefficients
$\hat{\boldsymbol{b}}$	Estimated $\boldsymbol{b}$
$\boldsymbol{B}$	Matrix of regression coefficients
$\hat{\boldsymbol{B}}$	Estimated $\boldsymbol{B}$
C	A constant term
$\boldsymbol{c}$	Weight vector for $\boldsymbol{Y}$
$\boldsymbol{C}$	Weight matrix for $\boldsymbol{Y}$
d_{eyes}	Distance between two eyes on the aligned face
$d_{ncc}(\boldsymbol{x}, \boldsymbol{y})$	NCC distance between $\boldsymbol{x}$ and $\boldsymbol{y}$
dx_{mouth}	Horizontal offset of the mouth from the point between the eyes in the profile view
$\boldsymbol{E}$	Residual matrix for $\boldsymbol{X}$
$\boldsymbol{f}$	Noise term
f	Spacing factor between kernels in the frequency domain
$\boldsymbol{F}$	Matrix of random errors, residual
f_{mouth}	Mouth offset parameter
h	Height of the aligned face
h_b	Height of local blocks
n	Number of samples in $\boldsymbol{X}$ and $\boldsymbol{Y}$
k_{max}	Maximum frequency of the Gabor wavelet
$k_{\nu,\mu}$	Wave vector of the Gabor wavelet
$\boldsymbol{p}$	A loading vector of $\boldsymbol{X}$
$\boldsymbol{P}$	Matrix of loadings for $\boldsymbol{X}$
$\boldsymbol{q}$	A loading vector of $\boldsymbol{Y}$

$\boldsymbol{Q}$	Matrix of loadings for $\boldsymbol{Y}$
r_{xy}	Sample correlation between $\boldsymbol{x}$ and $\boldsymbol{y}$
$s_{l.eye}$	Position of the left eye
s_{mouth}	Position of mouth
$s_{r.eye}$	Position of the right eye
$s_{v.eye}$	Position of the visible eye
s_x^2	Sample variance of $\boldsymbol{x}$
s_x	Standard deviation of $\boldsymbol{x}$
$\boldsymbol{t}$	A score vector of $\boldsymbol{X}$
$\boldsymbol{T}$	Matrix of scores for $\boldsymbol{X}$
T	Homogeneous similarity transformation matrix between the input and aligned face images
$\boldsymbol{u}$	A score vector of $\boldsymbol{Y}$
$\boldsymbol{U}$	Matrix of scores for $\boldsymbol{Y}$
v_{xy}	Sample covariance between $\boldsymbol{x}$ and $\boldsymbol{y}$
$\boldsymbol{V}_{XY}$	Covariance of $\boldsymbol{X}$ and $\boldsymbol{Y}$ matrices
w	Width of the aligned face
$\boldsymbol{w}$	Weight vector for $\boldsymbol{X}$
$\boldsymbol{W}$	Weight matrix for $\boldsymbol{X}$
w_b	Width of local blocks
$\boldsymbol{X}$	Input data matrix
$\boldsymbol{x}$	A sample in $\boldsymbol{X}$
$\bar{x}$	Sample mean of $\boldsymbol{x}$
$\hat{\boldsymbol{x}}$	Projected $\boldsymbol{x}$
x_{center}	x center of the aligned face
$\boldsymbol{x}_{centered}$	Centered $\boldsymbol{x}$ obtained by subtracting its mean
$\boldsymbol{x}_{scaled}$	Scaled $\boldsymbol{x}$ obtained by dividing by its standard deviation
$\boldsymbol{Y}$	Output data matrix
$\boldsymbol{y}$	A sample in $\boldsymbol{Y}$
$\hat{\boldsymbol{y}}$	Projected $\boldsymbol{y}$
y_{eyes}	y coordinate of eyes on the aligned face
y_{mouth}	y coordinate of mouth on the aligned face

μ	Orientation of the Gabor wavelet
ν	Scale of the Gabor wavelet
σ	Gaussian size
$\psi(\vec{x};\nu,\mu)$	Gabor wavelet
ϕ	Pose angle
Δ_{mouth}	Horizontal offset for mouth block

LIST OF ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
AAM	Active Appearance Model
AU	Action Units
BU3DFE	Binghamton University 3D Facial Expression Database
CCA	Canonical Correlation Analysis
DCT	Discrete Cosine Transform
FACS	Facial Action Coding System
GMI	Gabor Magnitude Images
GMM	Gaussian Mixture Model
GPR	Gaussian Process Regression
GPU	Graphical Processing Unit
HCI	Human Computer Interaction
HoG	Histogram of Oriented Gradients
knn	k-nearest Neighbor Classifier
LBP	Local Binary Patterns
LDA	Linear Discriminant Analysis
LGBP	Local Gabor Binary Patterns
LPP	Locality Preserving Projection
LR	Linear Regression
LVM	Latent Variable Methods
MFA	Marginal Fisher Analysis
MLR	Multiple Linear Regression
NCC	Normalized Cross Correlation
NIPALS	Non-linear Iterative Partial Least Squares
NN	Nearest Neighbor
NPE	Neighborhood Preserving Embedding
PCA	Principal Components Analysis
PCR	Principal Components Regression

PLS	Partial Least Squares
RCM	Region Covariance Matrix
RVR	Relevance Vector Regression
SIFT	Scale Invariant Feature Transform
SVD	Singular Value Decomposition
SVM	Support Vector Machines
SVR	Support Vector Regression

1. INTRODUCTION

Automated analysis of facial expressions has been an active research topic in computer vision over the last years. Facial expressions constitute an essential part of non-verbal communication between human beings. Interpretation of these expressions gives a great deal of information about the thoughts and emotional state of an individual. Automatic extraction of this information is possible with computer systems that can analyze facial expressions.

Automatic facial expression analysis finds interesting applications in several areas. The best known application is the robots that can understand and even show facial expressions in Human Computer Interaction (HCI) systems. There are also many potential commercial applications such as intelligent customer services, call centers, game and entertainment industries. Various research areas including psychology, psychiatry, and behavioral sciences benefit from the findings of the automatic facial expression analysis.

Due to these wide variety of uses, automatic analysis of facial expressions has attracted great attention from researchers. Most of the facial expression recognition systems attempt to recognize a set of prototypic emotional expressions, that is happiness, surprise, anger, sadness, fear, and disgust. This practice follows from the studies of Ekman and Friesen [1] and Ekman [2] that propose that basic emotions have corresponding prototypic facial expressions. Other studies use Facial Action Coding System (FACS) [3,4] for describing facial expressions by Action Units (AUs). Activity of facial muscles or muscle groups and their intensities are measured and assigned to different predetermined AU classes. These classes can be divided into upper and lower face action units. Upper face AUs are related to eyes, their surrounding areas, and eyebrows; while lower face AUs define deformations of mouth and cheeks.

Automatic recognition of expressions can be a very challenging task, since there are many underlying factors that affect the appearance of facial expressions. One

factor is the presence of subject differences such as texture of the skin, hair style, age, gender, and ethnicity. All these factors have a large influence on the appearance of the face, and consequently, on facial expressions. In addition to the differences in appearance, there might be differences in expressiveness; that is, individuals perform expressions differently from each other. Expressiveness specifically refers to the the degree of facial plasticity, morphology, frequency of intense expression, and overall rate of expression [5]. These individual differences are an important aspect of identity and characteristic facial actions may be used as a biometric property to improve the accuracy of the facial recognition algorithms [6].

Another factor that makes automatic facial expression recognition a hard problem is the presence of pose variations. The difficulty comes from the fact that the pose change causes a non-linear transformation of the 2D face image. Moreover, some areas of the face become self-occluded and some areas might have a very different appearance from different viewpoints. Most of the existing studies focus on recognizing expressions from frontal or nearly frontal view facial images. Different approaches that handle pose variations are necessary for recognizing expressions from arbitrary viewpoints, since the appearance of facial expressions significantly changes from one pose to another.

There are some other factors that are common to all computer vision problems such as variations caused by illumination changes, scene complexity, image acquisition and resolution, reliability of ground truth and databases. To sum up, facial expression recognition has a large problem space with multiple dimensions. There might be several problem formulations and different methodologies can be developed to solve these problems.

1.1. Motivation

In this thesis, we focus on multi-view facial expression recognition. Expression recognition systems have to deal with arbitrary viewpoints to be able to work under the uncontrolled conditions like the real-word situations. Most of the existing multi-view studies discretize the viewpoints into a set of intervals and use a separate model for

each viewpoint. Each model functions as a recognizer for a particular pose angle and needs representative data from that pose angle for its training.

The problem starts in the collection of the data. Most of the available datasets for facial expression recognition are taken from frontal or near frontal views. Even when we record expressions of people for data collection purposes, they naturally show expressions from the frontal view. If we also want to recognize expressions from arbitrary viewpoints, we need to collect more data. An exhaustive dataset, with all expression-viewpoint combinations need to be collected from each individual, for training a pose-specific system. One way to go is using 3D models, which is a cumbersome process.

Our motivation is developing a pose-independent expression recognition system without the need for data from all viewpoints. For example, we have only frontal data of each person displaying six prototypic expressions and we want to be able to recognize expressions of that person from arbitrary viewpoints. This requires establishing a relation between expressions of an individual from different viewpoints, and we realize that by learning a mapping from one pose to another.

1.2. Literature Review

Facial expression recognition is a widely studied research topic in computer vision. The existing studies have achieved high accuracies in various formulations of the same problem. Extensive reviews of these approaches can be found in [5,7–9]. In this section, we first review multi-view facial expression recognition methods in the literature and then give examples of using Partial Least Squares in multi-view facial image analysis.

1.2.1. Multi-view Facial Expression Recognition

The pose change is a challenging problem in the research of facial expression recognition. To overcome this problem, some multi-view approaches have been developed. Here, we present an overview of these studies on multi-view facial expression

Table 1.1. Multi-view Facial Expression Recognition in the Literature.

Study	Scheme	Features	Dim. Reduction	Classification
[10]	pose-specific	2D displacement vectors	PCA, LDA, LPP	QBN, Parzen, SVM, knn
[11]	two-step and composite	Hog, LBP, SIFT	PCA, LDA, LPP	NN
[12]	two-step	variants of LBP, LGBP	-	multi-class SVMs
[13]	-	SIFT	PCA	minimizing Bayes error
[14]	-	dense SIFT	a new discriminant analysis	RCM
[15]	-	SIFT	-	πSIFT
[16]	two-step, AAMs	SIFT, DCT	F-score	multi-class SVMs
[17]	mapping	coordinates of landmarks	-	LR, SVR, RVR, GPR

recognition. A summary of the methods reviewed are shown in Table 1.1.

In [10], Hu *et al.* train different classifiers for each pose to compare the performance of non-frontal view classifiers with the frontal view ones. For representing expression classes, they use geometric 2D displacement of facial feature points around the mouth, eyes, and eyebrows between expression and neutral faces of a person at corresponding angles. After normalizing the features to zero mean and unit variance, they train different classifiers and compare their results including Linear Bayes Normal classifier, Quadratic Bayes Normal classifier, Parzen classifier, Support Vector Machines (SVM) with linear kernel, and k-nearest neighbor (k-NN) classifier. They also use Principal Components Analysis (PCA), Linear Discriminant Analysis (LDA), and Locality Preserving Projection (LPP) with k-NN. They obtain lower error rates for non-frontal views. The authors conclude that the reason for non-frontal views to achieve better results than the frontal view might be that frontal faces contain redundancy due to the symmetry of the face, while faces rotated by around 45 degrees additionally contain depth information.

In [11], they utilize two different classification schemes. The first one is a 2-step cascade classification, which is first, a pose classifier, then pose specific expression classifiers are used and the second one is composite classification, which treats each pose-emotion combination as a class. They compare different feature descriptors extracted at ground-truth facial landmark points and dimensionality reduction techniques for these schemes. They use Histogram of Oriented Gradients (HoG), Local Binary Patterns (LBP), and Scale Invariant Feature Transform (SIFT) as feature descriptors.

They first apply a feature selection method, and then perform classification by using Nearest Neighbor classifier. They experiment with PCA, LDA, and LPP as feature selection methods. LPP produces the lowest error rate with SIFT and LBP features. Also, experiments based on a combination of SIFT+LPP, HoG+LPP and LBP+LPP classifiers are performed, and these combinations produce the best results.

Similar to their first scheme, Moore and Bowden utilize a two-step classification approach in [12]. They analyze different forms of LBP including uniform rotation invariant LBP, rotation invariant LBP, uniform LBP obtained from the gradient magnitude image, standard LBP, multi-scale LBP, and Local Gabor Binary Patterns (LGBP).They use multi-class SVMs as pose specific classifiers. Their experiments show that multi-scale LBP and LGBP perform the best and their combination improves results further.

In [13], Zheng *et al.* develop a unified Bayes theoretical framework by using SIFT descriptors extracted from annotated landmarks as features. PCA is used to reduce the dimensionality of the feature vector. They formulate the recognition problem by minimizing an upper bound of the Bayes error. They use power iteration approach to find optimal solutions.

In [14], the authors use dense SIFT descriptors, which were extracted on a grid of patches on the face image. The covariance of the SIFT features is then used for calculating the Region Covariance Matrix (RCM) to model the facial deformations. They propose a new discriminant analysis theory to reduce dimensionality and preserve the most discriminative information by minimizing an estimated multi-class Bayes error derived under the Gaussian Mixture Models (GMM).

In [15], Soyel *et al.* study matching of SIFT features across different poses. They present a scheme called affine transform based discriminative pose invariant SIFT to reduce SIFT mismatches.

In [16], the authors first automatically extract facial landmark points by fitting a

pose-specific Active Appearance Model (AAM) to the face image. They represent the face by SIFT and Discrete Cosine Transform (DCT) extracted from the located facial feature points. After the F-score feature selection method is applied, pose-specific linear multi-class SVM classifiers are used for classification.

One study, which aims to establish a relationship between different poses is [17]. Rudovic *et al.* learn a mapping of facial landmarks, like mouth corner, from non-frontal to frontal views, so that a frontal classifier can also be used for non-frontal views. They compare a number of regression models for this mapping including Linear Regression (LR), Support Vector Regression (SVR), Relevance Vector Regression (RVR), and Gaussian Process Regression (GPR). The goal is to learn mapping functions to predict locations of facial landmarks in frontal view faces given the locations in non-frontal views.

1.2.2. PLS in Multi-view Facial Image Analysis

In face related problems, linear subspace methods such as PCA or LDA are often applied. However, appearances of expressions in 2D highly vary from one pose to another. In other words, pose variations separate the faces into several different subspaces [18]. Then, problem of matching faces from two different poses requires comparing two vectors from different subspaces, therefore such methods fail in case of large pose variations. This motivated the studies on face recognition to seek to find pose-independent latent spaces. We review the approach of these studies in this section.

Prince *et al.* [19] propose a model called tied factor analysis to find a latent identity space for each pair of poses. The model describes how the identity as an underlying factor created the appearance changed by the pose. They build local models for facial features and combine information from each model by using the naive Bayes classifier for recognition.

Sharma et al. *et al.* use partial least squares (PLS) to project faces of an individ-

ual from two different poses into a common linear subspace, in which they are highly correlated. Recognition is then performed in this latent space by using the nearest neighbor algorithm. They also apply this approach to other multi-modal face recognition schemes such as sketch-photo recognition and variations in resolution. In [20], same authors extend their approach to generalized multi-view analysis, where they extend several feature extraction techniques such as PCA, LDA, LPP, Neighborhood Preserving Embedding (NPE), and Marginal Fisher Analysis (MFA) into the multi-view case.

Li *et al.* [21] propose that faces can be well represented by a linear subspace and the coefficients of the linear combinations representing faces are pose invariant. They employ the Ridge and Lasso regression methods to find these regression coefficients and report improved results by using an alignment method and local Gabor features. In [18], the same authors use partial least squares to find pose-independent feature vectors for face recognition. Similarly, Fischer *et al.* perform a detailed analysis of PLS for cross-pose face recognition by comparing the holistic and local representation methods in [22].

1.3. Approach and Outline of the Thesis

In this thesis, we present a method to recognize six prototypic facial expressions of an individual across different pose angles. We first align the faces and then extract block features around two eyes and the mouth on the aligned image. We experiment with Gabor filters and direct intensity values for local face representation. Then, we use PLS to map the expressions from different poses into a common subspace, in which correlation between them is maximized. We train PLS on coupled expressions of the same identity from two different poses. This way of training lets the learned bases model the differences between expressions of different poses by excluding the effect of the identity. Feature vectors are projected into the latent space by using these PLS bases as projection matrices. During testing, we compute distance between the projected features of blocks for each present block pair and average the result over the number of present local blocks. Then, classification is performed by using the nearest

neighbor algorithm.

We first give a brief mathematical background for Latent Variable Methods in Chapter 2. General formulation of regression analysis is described and a mathematical baseline is constituted for PLS through the explanation of the Non-linear Iterative Partial Least Squares (NIPALS) algorithm for PCA.

In Chapter 3, we present our approach and discuss each step in detail. We first describe alignment and feature extraction methods. Then, we explain the NIPALS algorithm for PLS and formulate the cross-pose facial expression recognition based on this algorithm.

In Chapter 4, we first describe the dataset and experimental setup we used. Then, we present experimental results for pose pairs and discuss the effect of parameters on these results.

2. MATHEMATICAL BACKGROUND FOR LATENT VARIABLE METHODS

Latent Variable Methods (LVM) assume that observable data or relations between data are generated from an underlying latent factor. LVMs model the mapping between the observed and latent space by extracting value from data. Learned model can be later used for analysis, prediction or generation purposes.

In this thesis, we use one such method, PLS to model the relations between different poses of an expression. We assume that there is an underlying pose-invariant representation for each expression and that representation creates the observed data, which is the pose-varying facial expressions. We propose an approach that creates a mapping from this idealized expression space to the observed data space. In expression space, the representation for each expression does not vary with pose.

Classical problem of linking up two variables X and Y is called regression analysis and there are a variety of methods developed for modeling and analyzing these variables by estimating the relation between them. These methods aim to understand how y varies as a function of x and ultimate goal is to be able to predict y from x.

PLS method is built on the classical regression methods. In this section, we provide an initial insight for the basic statistics and regression methods, which are essential to explain PLS method in the next section. Initial form of the Non-linear Iterative Partial Least Squares (NIPALS) algorithm for the PLS method is described here for Principal Components Analysis (PCA).

In this section, we follow the conventions in the PLS tutorial by Geladi and Kowalski [23]. Here, we briefly describe the baseline of the regression methods from the perspective of PLS methods. For more information, please refer to the original text in [23].

2.1. Statistical Concepts and Notations

Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be $(n \times 1)$ sample vectors [1] of random variables $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively and $\mathbf{1}$ be an$(n \times 1)$ vector of 1's:

Mean is the long-run average value of random variable $\boldsymbol{X}$. The sample mean of $\boldsymbol{x}$:

$$\bar{x} = \frac{1}{n}(x_1 + x_2 + ... + x_n) = \frac{1}{n}\mathbf{1}^T\boldsymbol{x} \tag{2.1}$$

Centering a vector $\boldsymbol{x}$ means subtracting its mean: $\boldsymbol{x}_{centered} = \boldsymbol{x} - \mathbf{1}\bar{x}$

Variance is the long-run average of squared deviation from mean. The sample variance of $\boldsymbol{x}$:

$$\begin{aligned} s_x^2 &= \frac{1}{n-1}\{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + ... + (x_n - \bar{x})^2\} && (2.2) \\ &= = \frac{1}{n-1}||\boldsymbol{x} - \mathbf{1}\bar{x}||^2 && (2.3) \end{aligned}$$

For a centered vector $\boldsymbol{x}$, $\bar{x} = 0$ and s_x^2 becomes:

$$\begin{aligned} s_x^2 &= \frac{1}{n-1}||x||^2 && (2.4) \\ &= = \frac{1}{n-1}\boldsymbol{x}^T\boldsymbol{x} && (2.5) \end{aligned}$$

Standard deviation is the square-root of variance. Sample standard deviation of $\boldsymbol{x}$:

$$s_x = \sqrt{\frac{1}{n-1}\{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + ... + (x_n - \bar{x})^2\}} \tag{2.6}$$

[1] The terminology sample quantity here refers to the variation between the n samples for a given variable, i.e. variation between the n values of a column.

Scaling a centered vector $\boldsymbol{x}$ means dividing it by its standard deviation s_x: $\boldsymbol{x}_{scaled} = \frac{1}{s_x}\boldsymbol{x}$

Covariance between two random variables is the long-run average of product of deviations from means. The sample covariance between $\boldsymbol{x}$ and $\boldsymbol{y}$:

$$\begin{aligned} v_{xy} &= \frac{1}{n-1}\{(x_1-\bar{x})(y_1-\bar{y}) + ... + (x_n-\bar{x})(y_n-\bar{y})\} && (2.7) \\ &= \frac{1}{n-1}(\boldsymbol{x}-\mathbf{1}\bar{x})^T(\boldsymbol{y}-\mathbf{1}\bar{y}) && (2.8) \end{aligned}$$

For centered vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, v_{xy} becomes:

$$v_{xy} = \frac{1}{n-1}\boldsymbol{x}^T\boldsymbol{y} \tag{2.9}$$

This also holds for matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$, that is covariance matrix $\boldsymbol{V}_{XY}$ is proportional to $\boldsymbol{X}^T\boldsymbol{Y}$.

Correlation between two random variables is simply their covariance scaled by the standard deviations. The sample correlation between $\boldsymbol{x}$ and $\boldsymbol{y}$:

$$r_{xy} = \frac{v_{xy}}{s_x s_y} \tag{2.10}$$

The correlation is always between -1 and 1: $-1 \leq r_{xy} \leq 1$. The two variables are highly positively correlated if the correlation is near 1. The two variables are highly negatively correlated if the correlation is near -1. A correlation of exactly $|1|$ means completely correlated variables and that the two random variables are exactly linearly dependent.

Rank of a matrix is the maximum number of linearly independent columns (or rows) of this matrix. It is a number expressing true underlying dimensionality of a matrix [23].

2.2. Least Squares (Simple Linear Regression)

Simple Linear Regression is the basic way of linking two variables $\boldsymbol{x}$ and $\boldsymbol{y}$. Let $\boldsymbol{y}$ be the response variable and $\boldsymbol{x}$ centered explanatory variable:

$$\boldsymbol{y} = \mathbf{1}b_0 + \boldsymbol{x}b + \boldsymbol{f} \tag{2.11}$$

where b is the slope and b_0 is the intercept of regression line, and $\boldsymbol{f}$ is the noise term. $\boldsymbol{f}$ refers to the unmodelled components of the linear model which can be caused from measurement error or other random variation. $\boldsymbol{y}$ is often called as dependent variable and $\boldsymbol{x}$ is called as independent variable, since $\boldsymbol{y}$ depends on $\boldsymbol{x}$.

We want parameter estimates for b_0 and b. The least squares estimators for this model can be obtained by minimizing the sum of squared errors.

$$\hat{b_0} = \bar{y} \tag{2.12}$$

$$\hat{b} = \frac{\boldsymbol{x}^T\boldsymbol{y}}{\boldsymbol{x}^T\boldsymbol{x}} \tag{2.13}$$

$$= \frac{v_{xy}}{s_x^2} \tag{2.14}$$

After inserting the value of $b_0 = \bar{y}$:

$$\boldsymbol{y} = \mathbf{1}\bar{y} + \boldsymbol{x}b + \boldsymbol{f} \tag{2.15}$$

$$\boldsymbol{y} - \mathbf{1}\bar{y} = \boldsymbol{x}b + \boldsymbol{f} \tag{2.16}$$

Finally equation becomes $\boldsymbol{y} = \boldsymbol{x}b + \boldsymbol{f}$ where both $\boldsymbol{y}$ and $\boldsymbol{x}$ are centered.

2.3. Multiple Linear Regression (MLR)

Simple Linear Regression problem deals with single $\boldsymbol{x}$ and $\boldsymbol{y}$ variables. When input is a vector or both input and output are vectors, a different modeling applies. Multiple Linear Regression (MLR) is a classical calibration method which can be considered as the prototype for all other calibration methods. In MLR method, $\boldsymbol{y}$ is defined as a constant plus a weighted sum of the explanatory variables:

$$y_i = b_0 + x_{i1}b_1 + x_{i2}b_2 + \ldots + x_{ik}b_k + f_i \tag{2.17}$$

for $i = 1, \ldots, n$, where b_j is the regression coefficient for the jth explanatory variable, b_0 is the intercept of regression line, n is the sample size, k is the length of x variable, and f_i is the noise term.

Consider $\boldsymbol{X}$ as a $n \times k$ matrix and regression coefficients $\boldsymbol{b}$ as a vector of $k \times 1$, then we can rewrite MLR as

$$\boldsymbol{y} = \mathbf{1}b_0 + \boldsymbol{X}\boldsymbol{b} + \boldsymbol{f} \tag{2.18}$$

where $\boldsymbol{y}$ is $n \times 1$ vector.

By minimizing the sum of squares of the errors, the least squares estimators for this model is obtained.

$$\hat{b_0} = \bar{y} \tag{2.19}$$

$$\hat{\boldsymbol{b}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \tag{2.20}$$

$$= \boldsymbol{V}_X^{-1}\boldsymbol{V}_{Xy} \tag{2.21}$$

Similar to Simple Linear Regression case, after inserting the value of b_0, Equation

2.18 becomes

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{b} + \boldsymbol{f} \tag{2.22}$$

where both $\boldsymbol{y}$ and $\boldsymbol{X}$ are centered.

2.4. Multivariate MLR

When both input and output are multivariate, that is $\boldsymbol{Y}$ is also a matrix of size $n \times m$, problem becomes multivariate MLR. In that case, above MLR model holds for each column of $\boldsymbol{Y}$, it can be written similar to $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{b} + \boldsymbol{f}$ in centered form:

$$\boldsymbol{y}_l = \boldsymbol{X}\boldsymbol{b}_l + \boldsymbol{f}_l \tag{2.23}$$

for $l = 1, ..., m$.

Let $\boldsymbol{B}$ be the matrix of regression coefficients of size $k \times m$, $\boldsymbol{F}$ matrix of random errors of size $n \times m$, then equation for centered variables of multivariate MLR problem can be written as $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{B} + \boldsymbol{F}$. Multivariate MLR is equivalent to performing MLR on each column of $\boldsymbol{Y}$ separately.

The least squares estimators for this model is:

$$\hat{\boldsymbol{b}_0} = \bar{\boldsymbol{y}} \tag{2.24}$$

$$\hat{\boldsymbol{B}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y} \tag{2.25}$$

$$= \boldsymbol{V}_X^{-1}\boldsymbol{V}_{XY} \tag{2.26}$$

$\hat{\boldsymbol{B}}$ is defined in terms of the variance of $\boldsymbol{X}$ and covariance between $\boldsymbol{X}$ and $\boldsymbol{Y}$. This solution does not take into account correlations between different $\boldsymbol{y}$ variables ($\boldsymbol{V}_Y$).

2.5. Principal Components Analysis (PCA)

PCA finds the best summary of data $\boldsymbol{X}$ with the fewest number of summary variables, called scores, $\boldsymbol{T}$. There are a few ways of calculating the PCA model including Eigenvalue Decomposition, Singular Value Decomposition (SVD), and Non-linear Iterative Partial Least Squares (NIPALS) algorithm. Here, we only explain the NIPALS algorithm to establish the relation with the PLS method.

2.5.1. From the Perspective of Loadings and Scores

PCA can be seen as a method of writing data matrix $\boldsymbol{X}$ of rank r as a sum of r matrices of rank 1 [23]:

$$\boldsymbol{X} = \boldsymbol{M}_1 + \boldsymbol{M}_2 + ... + \boldsymbol{M}_r \tag{2.27}$$

These rank 1 matrices can be written as products of two vectors: a score $\boldsymbol{t}_i$ and a loading $\boldsymbol{p}_i$:

$$\boldsymbol{X} = \boldsymbol{t}_1\boldsymbol{p}_1^T + \boldsymbol{t}_2\boldsymbol{p}_2^T + ... + \boldsymbol{t}_r\boldsymbol{p}_r^T \tag{2.28}$$

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^T \tag{2.29}$$

The principal component is the best-fit line for the data points as shown in Figure 2.1. Left side of the Figure 2.1 shows $\boldsymbol{p}_i$, which is a row vector and its elements p_1 and p_2 are direction cosines. They are projections of a unit vector along the principal component on the axes. Right side of the Figure 2.1 shows the score vector $\boldsymbol{t}_i$ as a $n \times 1$ column vector, where n is the number of points. The elements of the score vector correspond to the coordinates of the respective points on the principal component direction.

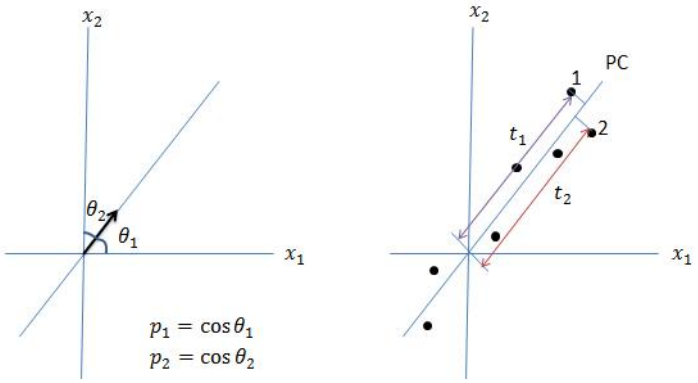

Figure 2.1. Illustration of the principal component, loadings, and scores for a simple case of two variables.

2.5.2. NIPALS Algorithm for PCA

The Non-linear Iterative Partial Least Squares (NIPALS) algorithm iteratively calculates scores $\boldsymbol{T}$ and loadings $\boldsymbol{P}$ such that $\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^T + \boldsymbol{F}$ where $\boldsymbol{F}$ is the residual [23]. It calculates $\boldsymbol{t}_1$ and $\boldsymbol{p}_1$ from the $\boldsymbol{X}$ matrix. Then, $\boldsymbol{t}_1\boldsymbol{p}_1^T$ is subtracted from $\boldsymbol{X}$ and the residual $\boldsymbol{F}_1$ is calculated. This procedure is called the deflation and it removes the part we can explain. Then, residual which is the part that remains unexplained, is used to calculate $\boldsymbol{t}_2$ and $\boldsymbol{p}_2$:

$$\begin{aligned} \boldsymbol{F}_1 &= \boldsymbol{X} - \boldsymbol{t}_1\boldsymbol{p}_1^T \\ \boldsymbol{F}_2 &= \boldsymbol{F}_1 - \boldsymbol{t}_2\boldsymbol{p}_2^T \\ &\ldots \\ \boldsymbol{F}_i &= \boldsymbol{F}i - 1 - \boldsymbol{t}_i\boldsymbol{p}_i^T \end{aligned} \tag{2.30}$$

It can be shown that $\boldsymbol{t}_1$ and $\boldsymbol{F}_1$ are orthogonal. Since $\boldsymbol{t}_2$ is initially picked from $\boldsymbol{F}_1$, it is also calculated as orthogonal to $\boldsymbol{t}_1$.

The NIPALS algorithm is as follows:

(i) Choose $\boldsymbol{t}_j$ as any column of $\boldsymbol{X}$

(ii) Calculate $\boldsymbol{p}_j$ by projecting columns of $\boldsymbol{X}$ onto $\boldsymbol{t}_j$: $\boldsymbol{p}_j = \boldsymbol{X}^T\boldsymbol{t}_j / \boldsymbol{t}_j^T\boldsymbol{t}_j$

(iii) Normalize $\boldsymbol{p}_j$ to length 1: $\boldsymbol{p}_j = \boldsymbol{p}_j / ||\boldsymbol{p}_j||$

(iv) Calculate $\boldsymbol{t}_j$ by projecting rows of $\boldsymbol{X}$ onto $\boldsymbol{p}_j$: $\boldsymbol{t}_j = \boldsymbol{X}\boldsymbol{p}_j / \boldsymbol{p}_j^T\boldsymbol{p}_j$

(v) Compare $\boldsymbol{t}_j$ with $\boldsymbol{t}_{j-1}$. If it is unchanged (or not changed significantly) stop, else continue with step 2.

After a component is calculated, $\boldsymbol{X}$ in the algorithm is replaced by its residual as in Equation 2.30.

To show how NIPALS describes the covariance in $\boldsymbol{X}$, we can replace scalars $\boldsymbol{t}_j^T\boldsymbol{t}_j$ and $\boldsymbol{p}_j^T\boldsymbol{p}_j$ by a general constant term C:

$$C\boldsymbol{p}_j = \boldsymbol{X}^T\boldsymbol{t}_j \tag{2.31}$$

$$C\boldsymbol{t}_j = \boldsymbol{X}\boldsymbol{p}_j \tag{2.32}$$

We can substitute Equation 2.32 into Equation 2.31:

$$C\boldsymbol{p}_j = \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{p}_j \tag{2.33}$$

This is the eigenvalue equation for $\boldsymbol{X}^T\boldsymbol{X}$. Similarly, when we substitute 2.31 into 2.32, we obtain eigenvalue equation for $\boldsymbol{X}\boldsymbol{X}^T$.

NIPALS algorithm calculates one component at a time, therefore it is well-suited for large datasets. Both Eigenvalue Decomposition and SVD calculate all components at once, even when a smaller dimensionality is required. Therefore, all software packages use NIPALS to compute PCA. Other approaches that are used to compute PCA cannot handle the missing data. It is possible to modify the NIPALS algorithm to take missing data into account [24]. There are further disadvantages of Eigenvalue Decomposition for large matrices such as difficult matrix operations and numerical overflows. However, they are slightly more accurate than NIPALS since error is spread over all components. Error in the NIPALS algorithm increases with more components.

NIPALS algorithm converges, although it might be slow at some cases. Convergence of the NIPALS algorithm is fast if the eigenvalues are well separated. Two close eigenvalues lead to very slow convergence, followed by a very fast convergence for the next iteration.

2.5.3. Principal Components Regression (PCR)

The idea behind the PCR is that a data matrix $\boldsymbol{X}$ can be represented by its score matrix $\boldsymbol{T}$. $\boldsymbol{T}$ has a smaller dimensionality and it still retains important features of $\boldsymbol{X}$. Then MLR can be performed with $\boldsymbol{T}$ in place of $\boldsymbol{X}$:

$$\boldsymbol{Y} = \boldsymbol{T}\boldsymbol{B} + \boldsymbol{F} \tag{2.34}$$

where regression coefficient becomes $\hat{\boldsymbol{B}} = (\boldsymbol{T}^T\boldsymbol{T})^{-1}\boldsymbol{T}^T\boldsymbol{Y}$. It is considered as a better representation, since $\boldsymbol{T}$ has the orthogonality property and therefore always invertible unlike $\boldsymbol{X}^T\boldsymbol{X}$.

3. CROSS-POSE FACIAL EXPRESSION RECOGNITION BASED ON PARTIAL LEAST SQUARES

In this section, we present the steps of the PLS approach for cross-pose facial expression recognition. We first describe the alignment method we used. Then, we describe Gabor wavelets in general and explain the details of the extraction of local blocks from the aligned image. We describe the PLS method for two block case in detail and explain the steps of the NIPALS algorithm. We continue with our problem formulation by using the PLS method and finish with the details of the classification step.

3.1. Alignment

Most previous works on both frontal and multi-view facial image analysis use alignment methods to improve the performance. The alignment step transforms face images into a common form so that their corresponding features can be correctly matched in later steps. This is generally performed by using the manually specified points for facial landmarks.

In case of only frontal faces, alignment is generally performed by using a reference eye row and interocular distance. Face image can be aligned by an euclidean transform using the provided or detected eye coordinates so that for all images the eyes are at the same position based on the given eye row and interocular distance.

In case of different pose angles, an eye may not be visible due to the face pose. This alignment method which is based on both eye coordinates cannot be applied, therefore a different approach is necessary. [25] propose a pose-specific alignment which uses manually specified target points for the facial landmarks in the aligned image. [21] annotate occluded landmarks by estimating the position of the landmark. Both approaches have some drawbacks. For pose-specific alignment, new points have to be

specified by hand for different poses. In the second case, alignment depends on manual annotation of invisible facial landmarks.

Similarly to [22], we utilize an alignment method which works for all pose angles, and gives a consistent scale and rotation of the face. Parameters of the alignment are illustrated in Fig.3.1.

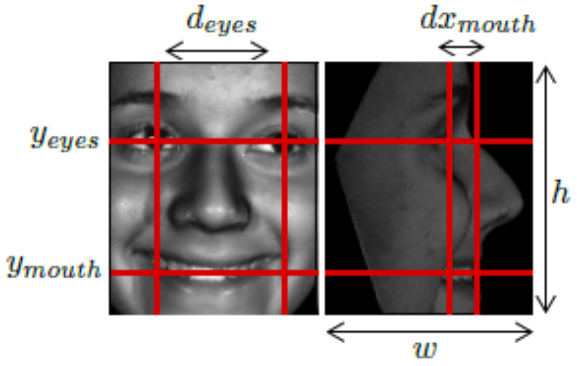

Figure 3.1. Illustration of alignment parameters.

We first calculate the positions of the visible eyes, $s_{l.eye}$ and/or $s_{r.eye}$, and the mouth center s_{mouth} in the input image. Then, we compute a similarity transform, T by specifying two point correspondences between the input image and the aligned image:

The first point correspondence is always the mouth center. Let $x_{center} = \frac{(w-1)}{2}$ be the horizontal center of the aligned image and ϕ be the pose angle. The positions in the input and aligned image are computed as follows:

$$s_1 = s_{mouth} \qquad t_1 = \begin{pmatrix} x_{center} + dx_{mouth}\ sin(\phi) \\ y_{mouth} \end{pmatrix} \tag{3.1}$$

In the case that both eyes are visible, the point between the eyes is used for the second correspondence:

$$s_2 = \frac{s_{l.eye} + s_{r.eye}}{2} \qquad t_2 = \begin{pmatrix} x_{center} \\ y_{eyes} \end{pmatrix} \tag{3.2}$$

In the case that only one eye is visible, the visible eye $s_{v.eye}$ is used for the second correspondence:

$$s_2 = s_{v.eye} \qquad t_2 = \begin{pmatrix} x_{center} \pm dx_{mouth}\ cos(\phi) \\ y_{eyes} \end{pmatrix} \tag{3.3}$$

The homogeneous transformation matrix, T is computed by solving the system of linear equations given by the two point correspondences: $\acute{s_1} = T\acute{t_1}$ and $\acute{s_2} = T\acute{t_2}$, where $\acute{u} = (u_x\ u_y\ 1)^T$ denotes the homogeneous coordinates of u, for $u \in \{s_1, s_2, t_1, t_2\}$.

3.2. Feature Extraction

In this thesis, we use a local face representation by either using direct intensity values or Gabor wavelets as features. In facial image analysis, local face representations have shown better performance than holistic representations. Gabor wavelets were inspired by 2D receptive field profiles of the mammalian cortical simple cells. They capture the local structure corresponding to spatial frequency (scale), spatial localization, and orientation selectivity, therefore, they can be successfully used in the facial image analysis [26].

3.2.1. Gabor Features

Gabor wavelets have been extensively used for facial image analysis due to their powerful representation capabilities and their biological relevance [26]. The conventional Gabor wavelets (kernels, filters) can be defined as follows:

$$\psi(\vec{x}; \nu, \mu) = \frac{k_{\nu,\mu}^2}{\sigma^2} e^{(-\frac{k_{\nu,\mu}^2 ||\vec{x}||^2}{2\sigma^2})} [e^{(ik_{\nu,\mu}\vec{x})} - e^{(-\frac{\sigma^2}{2})}] \tag{3.4}$$

where μ and ν define the orientation and scale of the Gabor kernels, and the wave vector $k_{\nu,\mu}$ is defined as follows:

$$k_{\nu,\mu} = \frac{k_{max}}{f^{\nu}} e^{\frac{i\pi\mu}{8}} \tag{3.5}$$

where k_{max} is the maximum frequency and f is the spacing factor between kernels in the frequency domain [26].

In this formulation, $e^{(ik_{\nu,\mu}\vec{x})}$ is the oscillatory wave function, whose real part and imaginary parts are the cosine and sine functions, respectively. ν controls the scale of Gabor wavelet, which mainly determines the center of the Gabor filter in the frequency domain; μ controls the orientation of the Gabor filters. In most cases, facial image analysis studies use Gabor wavelets of five different scales $\nu \in \{0, 1, 2, 3, 4\}$ and eight orientations $\mu \in \{0, 1, .., 7\}$ with the following parameters $f = \sqrt{2}$, $k_{max} = \frac{\pi}{2}$, with Gaussian size $\sigma = 2\pi$ [26]. In our experiments, we seek to find the optimal values for k_{max} and σ.

We perform full convolution of the Gabor wavelet with the aligned face image to obtain the Gabor Magnitude Images (GMI) in different scales and orientations.

3.2.2. Extraction of Local Blocks

After obtaining the GMIs, we extract local blocks around facial landmarks similar to [21] and [22] from each GMI. More specifically, we use local blocks around left eye, right eye and mouth since these regions provide the most discriminative information for facial expression recognition. In order to avoid border effects, we perform an additional padding of 32 pixels to the GMI before the extraction of the local blocks.

We extract blocks of size $w_b \times h_b$ centered on the eye centers and the mouth center by using the provided annotations of BU3DFE database. For frontal pose, local blocks can be extracted directly. On the other hand, in non-frontal poses, some of the directly extracted local blocks may include more and more of the background. This

causes performance degradation because the background information does not help for the recognition. In order to avoid background, we apply the following modifications as in [22]: First, we discard the eye block unless it is clearly seen in the non-frontal aligned face image. Secondly, we compute a horizontal offset Δ_{mouth} as in [22] for the mouth block. Then, we shift the mouth block horizontally and move it further into the face to decrease the number of background pixels in the block. The horizontal offset is computed as follows:

$$\Delta_{mouth} = -f_{mouth} \ w_{mouth} \ sin(\phi) \tag{3.6}$$

where f_{mouth} is the mouth coefficient, and ϕ is the pose angle of the face. Figure 4.3 shows some examples of the blocks extracted on aligned face images.

3.3. Partial Least Squares (PLS)

We use PLS for cross-pose facial expression recognition. PLS which is also an acronym for "Projection to Latent Scores" is a class of techniques for modeling relations between blocks of observed variables by means of latent variables [27]. The term was originated by the studies of Herman Wold in the 1970's [28].

PLS has many variants. There are two modes of PLS, called A and B [27]. It can also be applied to two or more data blocks. PLS2 and PLS1, a special case of the first one, are variants of PLS, which are used as PLS regression methods in first chemometrics and then many areas. Mode B of Wold's algorithm [28] is also referred as Canonical Correlation Analysis (CCA).

In this study, we use the general term PLS for two block Mode A PLS, which is a special case of Wold's algorithm [27]. PLS has been shown effective for a variety of cross-modality recognition problems including face recognition under the pose variation. We first give mathematical definition of PLS method and NIPALS algorithm to explain its usage for expression recognition across pose.

PLS models the relationship between the $(n \times N)$ input matrix $\boldsymbol{X}$ and $(n \times M)$ output matrix $\boldsymbol{Y}$ where rows are sample observations and columns are variables. PLS specifically models $\boldsymbol{X}^T\boldsymbol{Y}$ that is covariance of $\boldsymbol{X}$ and $\boldsymbol{Y}$ by means of latent variables [27]. Most important property of PLS is that it primarily models covariance between $\boldsymbol{X}$ and $\boldsymbol{Y}$ rather than variance of $\boldsymbol{X}$ or $\boldsymbol{Y}$ variables. For a detailed proof through the derivation of eigenvalue problems, please refer to [29], which develops the mathematical and statistical structure of PLS.

By following the formulation in [30], centered and scaled $\boldsymbol{X}$ and $\boldsymbol{Y}$ are decomposed into the form:

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^T + \boldsymbol{E} \tag{3.7}$$

$$\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{Q}^T + \boldsymbol{F} \tag{3.8}$$

where T, U are $n \times p$ matrices of the p extracted score vectors (latent vectors), $N \times p$ matrix P and $M \times p$ matrix Q are the matrices of loadings and $n \times N$ matrix E and $n \times M$ matrix F are the matrices of residuals.

In PLS approach, input and output vectors are mapped into a common vector space in such a way that covariance between projected input and output vectors is maximized. Both $\boldsymbol{X}$ and $\boldsymbol{Y}$ variables are considered as indicators of p latent variables, or scores, $\boldsymbol{t}$ and $\boldsymbol{u}$, respectively. PLS models the cross-covariance by pairs of these scores such that $(\boldsymbol{t}_1, \boldsymbol{u}_1), ..., (\boldsymbol{t}_p, \boldsymbol{u}_p)$. Sets $\{\boldsymbol{t}_1, ..., \boldsymbol{t}_p\}$ and $\{\boldsymbol{u}_1, ..., \boldsymbol{u}_p\}$ are computed as the best representative column spaces of $\boldsymbol{X}$ and $\boldsymbol{Y}$ for $\boldsymbol{X}^T\boldsymbol{Y}$.

3.3.1. NIPALS Algorithm for PLS

Non-linear iterative partial squares (NIPALS) algorithm is the classical way of iteratively computing PLS basis vectors [30]. It maximizes the squares of covariance between the score vectors $\boldsymbol{t}$ and $\boldsymbol{u}$ by finding weight (basis) vectors $\boldsymbol{w}$ and $\boldsymbol{c}$ such that:

$$[cov(\boldsymbol{t}, \boldsymbol{u})]^2 = [cov(\boldsymbol{X}\boldsymbol{w}, \boldsymbol{Y}\boldsymbol{c})]^2 = max_{|r|=|s|=1}[cov(\boldsymbol{X}\boldsymbol{r}, \boldsymbol{Y}\boldsymbol{s})]^2 \tag{3.9}$$

where $cov(\boldsymbol{t}, \boldsymbol{u}) = \boldsymbol{t}^T\boldsymbol{u}/n$ denotes the sample covariance between score vectors $\boldsymbol{t}$ and $\boldsymbol{u}$.

It starts with random initialization of the output score vector $\boldsymbol{u}$ and iteratively computes these steps until convergence:

(i) $\boldsymbol{w} = \boldsymbol{X}^T\boldsymbol{u}/(\boldsymbol{u}^T\boldsymbol{u})$
(ii) $||\boldsymbol{w}|| = 1$
(iii) $\boldsymbol{t} = \boldsymbol{X}\boldsymbol{w}$
(iv) $\boldsymbol{c} = \boldsymbol{Y}^T\boldsymbol{t}/(\boldsymbol{t}^T\boldsymbol{t})$
(v) $||\boldsymbol{c}|| = 1$
(vi) $\boldsymbol{u} = \boldsymbol{Y}\boldsymbol{c}$

NIPALS algorithm for PLS can be considered as a two NIPALS algorithm for PCA at the same time, one for $\boldsymbol{X}$ (left column) and one for $\boldsymbol{Y}$ (right column). The point is that two are not completely separated relations. PLS does not just describe the variance of a data block free from the other block. Its modeling of relation between $\boldsymbol{X}$ and $\boldsymbol{Y}$ is improved by the exchange of scores. This way, each can get information about the other.

As in PCA, PLS is an iterative process. After the extraction of score vectors $\boldsymbol{t}$ and $\boldsymbol{u}$, $\boldsymbol{X}$ and $\boldsymbol{Y}$ are deflated and residuals are calculated for the next iteration. Loadings $\boldsymbol{p}$ and $\boldsymbol{q}$ are computed as coefficients of regressing $\boldsymbol{X}$ on $\boldsymbol{t}$ and $\boldsymbol{Y}$ on $\boldsymbol{u}$:

$$\boldsymbol{p} = \boldsymbol{X}^T t/(\boldsymbol{t}^T\boldsymbol{t}) \tag{3.10}$$

$$\boldsymbol{q} = \boldsymbol{Y}^T\boldsymbol{u}/(\boldsymbol{u}^T\boldsymbol{u}) \tag{3.11}$$

Different forms of deflations define several variants of PLS [30]. In PLS Mode A, which is originally designed by Herman Wold [28] to model the relations between two blocks of data, $\boldsymbol{X}$ and $\boldsymbol{Y}$ are deflated in a symmetric way:

$$\boldsymbol{X} = \boldsymbol{X} - \boldsymbol{t}\boldsymbol{p}^T \tag{3.12}$$

$$\boldsymbol{Y} = \boldsymbol{Y} - \boldsymbol{u}\boldsymbol{q}^T \tag{3.13}$$

According to [30], this approach is more appropriate for modeling existing relations between sets of variables in contrast to prediction purposes. PLS regression approaches PLS1 and PLS2 establish an asymmetric relationship between $\boldsymbol{X}$ and $\boldsymbol{Y}$ by defining a linear relation between the score vectors $\boldsymbol{t}$ and $\boldsymbol{u}$.

3.3.2. Projection to Latent Space

We will refer to representing $\boldsymbol{X}$ and $\boldsymbol{Y}$ by their corresponding score vectors $\boldsymbol{t}$ and $\boldsymbol{u}$ as projecting them to the latent space, where their samples are highly correlated.

Weight vectors $\boldsymbol{w}$ and $\boldsymbol{c}$ that maximize the covariance of the latent scores are computed by NIPALS algorithm and saved into the projection matrices $\boldsymbol{W}$ and $\boldsymbol{C}$, respectively. Then, test data can be projected into the latent space by using these projections:

$$\hat{\boldsymbol{x}} = \boldsymbol{W}^T\boldsymbol{x} \tag{3.14}$$

$$\hat{\boldsymbol{y}} = \boldsymbol{C}^T\boldsymbol{y} \tag{3.15}$$

After projection, facial expression recognition methods can be applied on pose-independent latent vectors $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$.

3.3.3. Discussion of PLS method for Cross-pose Recognition Problem

We want to use PLS to compare two face images of the same 3D scene from different viewpoints. PLS finds projections $\boldsymbol{w}$ and $\boldsymbol{c}$ that map two different data blocks $\boldsymbol{X}$ and $\boldsymbol{Y}$ into a common subspace. In our case, $\boldsymbol{X}$ and $\boldsymbol{Y}$ represent different viewpoints, that is difference that PLS needs to model is the pose changes.

Equation 3.9 shows that PLS will seek $\boldsymbol{w}$ and $\boldsymbol{c}$ that will produce highly correlated projected data blocks. For PLS to be effective in recognition, there has to exist such projections. It is possible to show that projections of two face images from different

viewpoints onto a subspace in which they are highly correlated exist [25].

Let $\boldsymbol{I}_k$ and $\boldsymbol{J}_k$ be the corresponding features of face images with the same expression from two different poses. We assume that there is an idealized version of this feature representing all characteristics of the feature, and call it $\boldsymbol{R}_k$. Then, we can write $\boldsymbol{R}_k$ as an underlying factor of $\boldsymbol{I}_k$ and $\boldsymbol{J}_k$:

$$\boldsymbol{I}_k = \boldsymbol{A}\boldsymbol{R}_k \tag{3.16}$$

$$\boldsymbol{J}_k = \boldsymbol{B}\boldsymbol{R}_k \tag{3.17}$$

We want to know whether it is possible to find $\boldsymbol{w}$ and $\boldsymbol{c}$ that projects $\boldsymbol{I}_k$ and $\boldsymbol{J}_k$ onto a space in which they are highly correlated, or equal as a simpler case:

$$\begin{aligned} \boldsymbol{w}^T\boldsymbol{I}_k &= \boldsymbol{c}^T\boldsymbol{J}_k \\ \boldsymbol{w}^T\boldsymbol{A}\boldsymbol{R}_k &= \boldsymbol{w}^T\boldsymbol{B}\boldsymbol{R}_k \\ \boldsymbol{w}^T\boldsymbol{A} &= \boldsymbol{c}^T\boldsymbol{B} \end{aligned} \tag{3.18}$$

LHS of Equation 3.18 is a linear combination of rows of $\boldsymbol{A}$ and RHS is a linear combination of the rows of $\boldsymbol{B}$. This means that Equation 3.18 can only be satisfied if row spaces of $\boldsymbol{A}$ and $\boldsymbol{B}$ intersect [25]. We need to find $\boldsymbol{A}$ and $\boldsymbol{B}$ that provides a correspondence between features of two images.

For the general case, we suppose that $\boldsymbol{A}$ is an identity matrix and $\boldsymbol{B}$ is a permutation matrix that changes the locations of features extracted. For some poses, there are occlusions due to the pose angle, that is some locations are not visible for these poses. In that case, $\boldsymbol{R}_k$ contains all possible values for the feature and $\boldsymbol{A}$ and $\boldsymbol{B}$ are binary matrices, whose each row contains only one 1. $\boldsymbol{A}$ creates feature values in $\boldsymbol{I}_k$ and $\boldsymbol{B}$ in $\boldsymbol{J}_k$. It is simply representing one side of a 3D face with one face image and other side with the other image. For a detailed proof, please refer to the related section of [25].

3.4. Cross-pose Facial Expression Recognition using PLS

Variations caused by different poses impose extra burden on the task of recognizing expressions, which is already a difficult problem due to the large differences across subjects. In this study, we propose using PLS to learn a relationship between faces of two different poses belonging to the same emotion and the same subject. The reason for using faces of the same subject is to exclude variations caused by identity and to reduce the problem to modeling of the variance in expressions caused by pose changes.

For each pose pair, we use PLS to compute a latent space for the existing blocks in both poses. We use a custom GPU implementation of the NIPALS algorithm [30] to compute the PLS bases. The input and output vectors are always centered by subtracting mean and scaled by dividing by standard deviation before training. Test data is transformed using the mean and standard deviation learned from training data.

3.4.1. Problem Formulation

In training, for a pose pair (p_i, p_j), we construct input $\boldsymbol{X}$ and output $\boldsymbol{Y}$ matrices, where samples in $\boldsymbol{X}$ are from pose p_i and samples in $\boldsymbol{Y}$ are from pose p_j. Corresponding samples are coupled by both identity and expression. We then perform PLS to compute projections $\boldsymbol{W}$ and $\boldsymbol{C}$ that maximize the the covariance of score vectors. Since training faces from two different poses are coupled by expression and identity, covariance between different poses of an expression is maximized. In testing, learned projections are used to estimate score vectors for new samples and then classification methods for expression recognition can be applied on these pose-independent latent vectors.

In faces with large pose angles, one of the eyes is not visible. This causes a problem in case of pose pair, where one pose has a large negative pose angle and other has a large positive angle. In one pose, only the right eye is visible and in the other only the left eye, therefore eyes cannot be used at all. To solve this problem, we assume that left and right eye are sufficiently symmetric and exploit this symmetry property

of eyes as in [22]. We train a PLS latent space for the opposite eye blocks.

3.4.2. Classification

We use the Nearest Neighbor (NN) algorithm for classification. NN algorithm compares a query image with target images in the database and assigns its label as the label of the target image that has the closest distance to the query image. We compute the distance between query and target face images in different poses by first extracting the blocks for both images as explained in the previous sections. Then, for those pairs of blocks that have trained PLS for the pose pair, we project the blocks into the latent space and compute the distance between the latent vectors. The computed differences for each block are averaged to yield the global difference.

The reason for using such a simple matching method is:

(i) Dimension of the input and output score vectors is the same and equal to the number of extracted PLS bases. Therefore, latent representations of input and output vectors lie in the same vector space.

(ii) PLS bases are learned based on a criterion that maximizes the covariance between the score vectors. As stated in [25], it is safe to assume that input and output latent scores are roughly embedded in a single linear manifold, since they are highly correlated.

To compute the score or the distance, several distance metrics can be used. In our experiments, we use L2-distance and Normalized Cross Correlation (NCC), which is known as a suitable distance metric for comparing Gabor features. NCC is defined as follows:

$$d_{ncc}(\boldsymbol{x}, \boldsymbol{y}) = 1 - \frac{(\boldsymbol{x} - \mu_x)\ (\boldsymbol{y} - \mu_y)}{(N-1)\,\sigma_x\,\sigma_y} \tag{3.19}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ denote the query and the target latent vectors, respectively and N is the length of $\boldsymbol{x}$ and $\boldsymbol{y}$.

4. EXPERIMENTS AND RESULTS

For the evaluation of the PLS approach for cross-pose facial expression recognition, extensive experiments were conducted. This section describes the data used for experiments and the experimental setup. Then, results of expression recognition across poses along with the effect of the parameters are presented.

4.1. Data

For the evaluation of the system, we use one of the most commonly used databases for multi-view facial expression recognition, the Binghamton University 3D Facial Expression Database (BU-3DFE) [31].

BU-3DFE contains 3D models of 100 subjects (56 female and 44 male) with texture and 83 annotated landmark points per model. Subjects in the database are of different age, ranging from 18 to 70 years, and a wide variety of ethnicities/races, including white, black, east-asian, middle-east-asian, hispanic-latino and others.

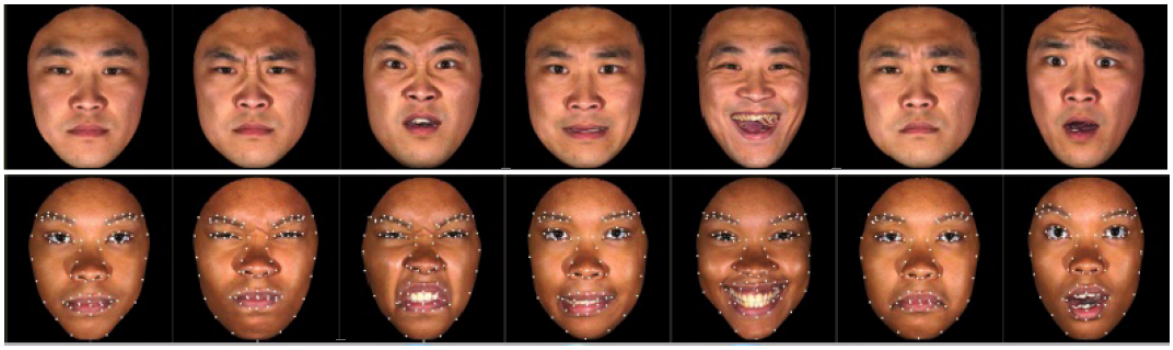

Figure 4.1. Example images and landmark points from BU-3DFE database. Shown expressions from left to right are: neutral, anger, disgust, fear, happiness, sadness, surprise.

Each subject shows 7 expressions, which are neutral, anger, disgust, fear, happiness, sadness and surprise (Figure 4.1). All subjects display all expressions except neutral at four different levels of intensity from low to high as shown in Figure 4.2. Consequently, for each subject there are 25 3D models present, which results in an

overall number of 2500 facial expression models.

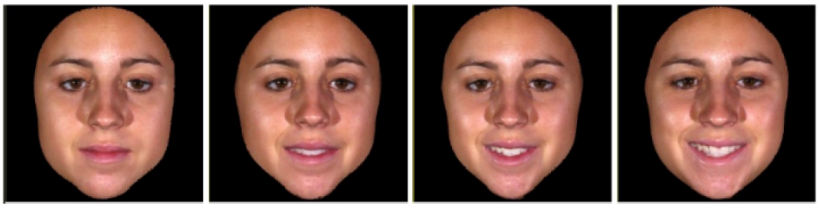

Figure 4.2. Example subject showing different levels of intensity for happiness class.

For this study, 2D images of facial expressions, taken from different view angles, are needed. Therefore, 3D models from the database are rendered together with the texture using VTK (The Visualization Toolkit). The models are rotated at yaw angle from -90 to $+90$ degrees in steps of 15 degrees. For every step, an image together with the coordinates of the landmark points is saved, resulting in 13 images per face model. After repeating this procedure for every model from the database, 13 poses are available, each containing 2500 images and 2500 sets of landmark points, which adds up to 32500 data elements (image + landmarks). The extracted images have a resolution of 300×300 pixels.

4.2. Experimental Setup

Data taken from the BU-3DFE database is divided into three sets of similar size, two sets containing 33 subjects, one set containing 34 subjects. One set is used for learning PLS bases, one for the optimization of parameters, and the last one for testing. In our experiments, we use a single image for each expression, which belongs to the highest intensity level (level 4). Later, we also experiment with other intensity levels to show the effect of intensity level on the results.

Before moving to the general execution, we first provide a clarification of the terms we use. A single execution of our experiments takes place between a pair of poses. Most of the time, we refer to these poses as input pose and output pose by following the convention in the NIPALS algorithm. Samples of the input matrix $\boldsymbol{X}$ are from the input pose and samples of the output matrix $\boldsymbol{Y}$ are from the output pose. By following the conventions of the face recognition studies, we can refer to the first

pose as gallery pose and second pose as probe pose, since input samples correspond to the gallery set and output samples to the probe set. Additionally, we sometimes refer to different individuals in the dataset as subjects.

For each expression in the training and the test set, first, the alignment and feature extraction steps are performed. For each block, input and output matrices are constituted as the corresponding expressions of a subject from the input and output poses, respectively. Then, normalization parameters as mean and standard deviation values are computed and saved for each feature from these matrices. After normalization, PLS bases are learned by using the NIPALS algorithm for present blocks of each pose pair in the training set.

During test, for corresponding pose pairs, samples from input and output poses are first normalized by using the normalization parameters saved during the training, and then projected into the latent space by using the learned PLS bases. For each present block pair, the distance between projected features of blocks are calculated and averaged over a number of present local blocks. Then, for each sample from the first pose, classification is performed by using the nearest neighbor algorithm, which is basically assigning its label as the label of the closest sample from the other pose. Recognition accuracy is calculated as the percentage of correct matchings across poses.

4.3. Results

4.3.1. Baseline Results

Most of the multi-view expression recognition studies train pose-specific classifiers and results are reported according to this scheme. It is natural to compare expressions from the same pose instead of matching expressions across poses and expect pose-specific classification to produce higher results. In order to show that it might not always be the case and to create a starting point for our approach, we first present results for each pose separately in Table 4.1.

Table 4.1. Pose-specific recognition rates as a baseline.

feat type	90l	75l	60l	45l	30l	15l	0	15r	30r	45r	60r	75r	90r
gabor	45.6	47.8	56.9	53.5	52.6	55.6	50.9	53.0	53.0	55.6	51.3	49.1	40.4
intensity	40.4	44.3	49.5	50.8	50.0	47.4	46.1	50.0	51.3	52.2	50.9	48.3	41.7
[16]	72.2	74.1	74.5	75.2	76.7	77.8	77.2	76.5	77.7	75.4	74.0	72.9	71.5

First two rows are the results of applying the nearest neighbor algorithm on the local blocks of Gabor and intensity features for each pose, separately. Non-frontal poses with large pose angles have lower recognition rates as expected. It should be noted that PLS has no part in these results in Table 4.1.

Recognition rates in the first two rows are low due to the simplicity of the classification method used. The nearest neighbor matching might be a good choice for comparing highly correlated feature vectors, however, it is not the case in here. Therefore, we also include the results of [16] which is a successful pose-specific facial expression recognition system.

4.3.2. Effects of Parameters

There are a number of factors that affect the recognition results in our experiments. These factors can be organized as alignment parameters, feature extraction details, number of PLS bases in the NIPALS algorithm, and the distance type used in classification. We performed a series of experiments with changing parameter settings to show the effects of these parameters.

In this section, we report results as the average of all pose pairs where input pose is always the frontal pose.

4.3.2.1. Alignment Parameters. Alignment is a common step for all of the facial image analysis problems. In our experiments, we used the optimal parameters reported in [22] for the alignment step, since they produce the best results for the face recognition problem. These are, $w = 104$, $h = 128$, $y_{eyes} = 42$, $y_{mouth} = 106$, $d_{eyes} = 62$, and $dx_{mouth} = 20$. Successful results for changing pose angles can be seen in Figure 4.3.

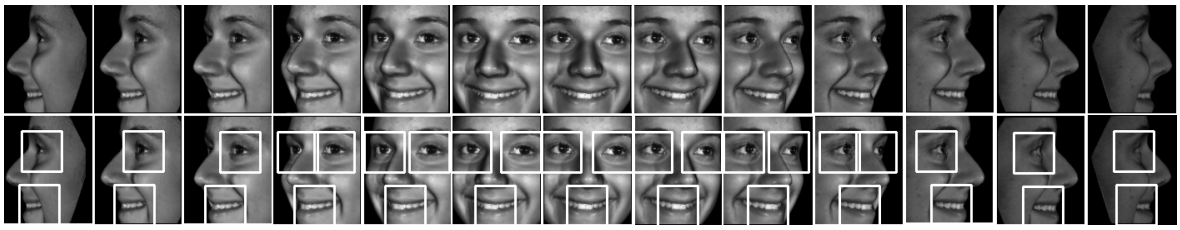

Figure 4.3. Example aligned face images in different poses (first row) and extracted local blocks used for Gabor feature extraction (second row). The pose angles change with an increment of 15 degrees left to right, from -90 degrees to +90 degrees.

4.3.2.2. Effects of Feature Extraction Parameters. Parameters of the feature extraction step depend on the feature type used. These are block size, mouth offset parameter for local blocks of intensity and Gabor features with additional Gabor specific parameters in case of Gabor features.

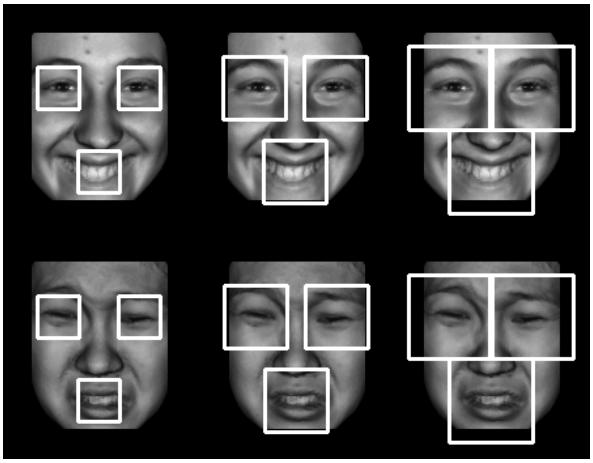

Figure 4.4. Visual representation of three different block sizes on two different expression faces.

In case of local blocks, we used three different block sizes: 32×32, 48×48, and 64×64. For computational purposes, we downscaled the intensity blocks to 32×32 pixels and downscaled all Gabor responses to 7×7 pixels. This way, we reduced the input size to $32 \times 32 = 1024$ for intensity features and $7 \times 7 \times 8 \times 5 = 1960$ for Gabor features.

We visually compare three different block sizes on two different expression faces in

Figure 4.4. Each row is the expression faces of a block size and each column corresponds to a block size from the smallest to the largest block size. As can be seen from the Figure 4.4, in case of the smallest block size, some areas related to the expression are not covered by any of the blocks. Especially, mouth block is too small to include curling up of the lips and wrinkles on the cheeks caused by the expression. In case of the largest block size, some of the background are included in the blocks. Even more background are included inside the blocks of faces from large pose angles.

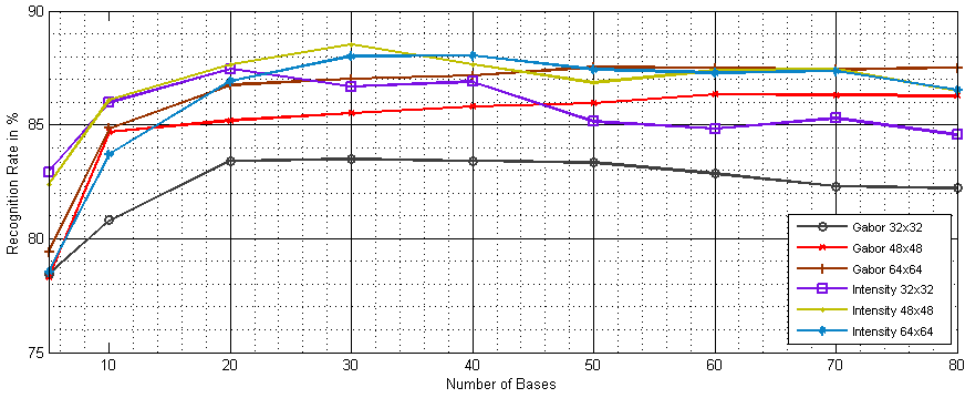

Figure 4.5. Effects of different block sizes for intensity values and Gabor features with changing number of PLS bases on the average recognition rate.

Figure 4.5 illustrates the effects of each block size and feature type on the recognition rates according to changing number of PLS bases. Local blocks of intensity values and Gabor features produce similar results for two largest block sizes. As can be seen from the Fig.4.5, block size of 32×32 gives the lowest results for both feature types as expected from the discussion above. For Gabor features, the highest recognition rates are obtained by using the largest block size, 64×64. On the other hand, block sizes of 64×64 and 48×48 result in similar recognition rates in case of intensity features, and the highest result is obtained by using block size of 48×48 at 30 bases. As discussed above, blocks also cover some of the background together with the additional areas on the face in case of the larges block size. Since we have a uniform background in BU3DFE, Gabor features are not affected from that and produce the highest results with the largest block size by benefiting from the extra areas covered. However, direct intensity values of the background are zero, and this causes the largest block size to

produce similar results with block size of 48×48.

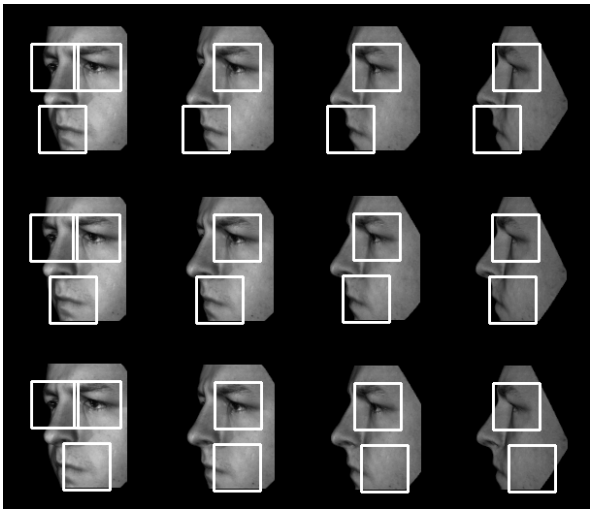

Figure 4.6. Visual representation of three different mouth offset parameters for expression faces from different viewpoints.

The mouth offset parameter is used to change the point which corresponds to the center of the mouth block position. It is calculated according to the pose angle and Fig 4.6 shows the necessity of changing the mouth block position, especially for the large pose angles. First row of the Figure 4.6 shows the position of the mouth block without using an offset parameter for a set of pose angles from 45 to 90, respectively. There are more background pixels included as the pose angle get larger. Second row shows the results with the optimal offset parameter found in our experiments. Results in the last row are the examples of using a large mouth offset parameter. A large mouth offset parameter shifts the mouth block more than necessary and some parts of the mouth become invisible.

To show the effects of the mouth offset parameter on the recognition rates, we experiment with a set of parameters: $0, 0.15, 0.35, 0.55, 0.75$. As can be seen from Figure 4.7, location of the mouth has an important effect on recognition rates, especially for large pose angles. The highest recognition rates for many of the pose angles are achieved when the mouth offset parameter equals to 0.35. This value minimizes background pixels inside the mouth block and still contains the outline of the mouth as shown in the examples in Figure 4.6.

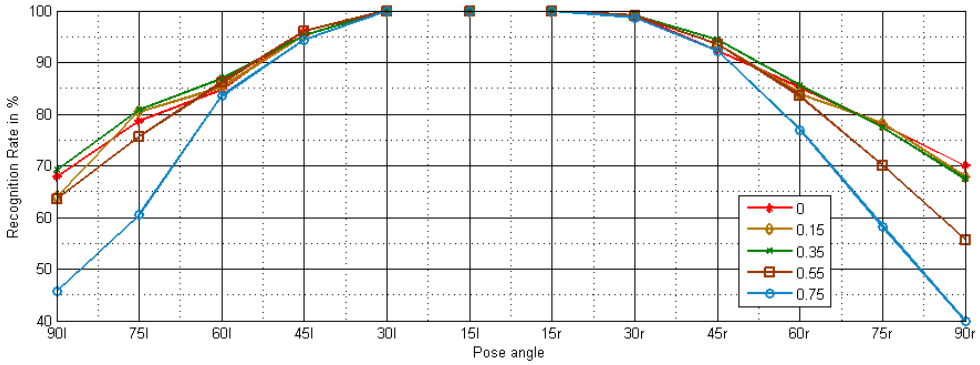

Figure 4.7. Effects of the mouth offset parameter for all pose angles.

Gabor representation produces recognition rates which are close to simply using intensity values. So far, we performed experiments by using the default values specified in Section 3.2.1. We experimented with different parameters of Gabor wavelets to improve Gabor representation. We specifically varied the values of k_{max} which is the scaling factor and Gaussian window width, σ. k_{max} is a factor in the wave vector $k_{\nu,\mu}$ which generates the Gabor kernels by scaling and rotation. k_{max} is responsible for the scaling part, and consequently is highly related to the size of the structures in the image. σ finds a compensation between the representation of coarse and fine structures in the image. Generally used default values of k_{max} and σ might not be the optimal ones for facial expression recognition. We perform a grid search for these parameters to find the optimal values of the parameters. As shown in Figure. 4.8, optimal values are different from the default values. The best results are achieved by using $k_{max} = \frac{\pi}{0.5}$ and $\sigma = 1.0\pi$.

4.3.2.3. Effects of Number of PLS bases. To show the effect of the number of PLS bases, we use a set of PLS bases from 10 to 80 in our experiments. Fig.4.5 and Fig.4.9 show the influence of the number of bases in different experiments. From these experiments, we realize that the optimal recognition performance is usually achieved when 30 PLS bases are used.

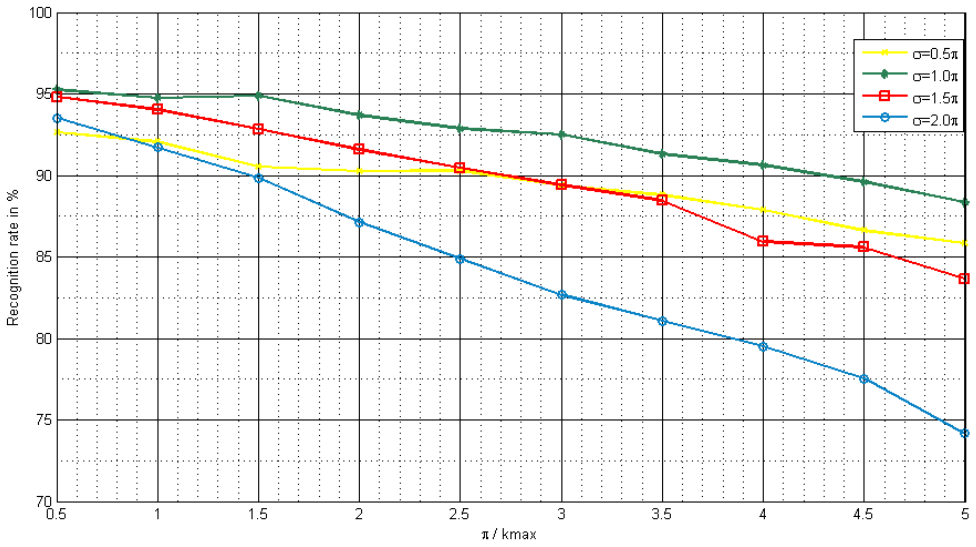

Figure 4.8. Effects of different Gabor parameters, k_{max} and σ.

4.3.2.4. Effects of the Distance Type. The results of two different distance types for both local blocks of Gabor features and intensity values can be seen from Fig.4.9 with changing number of PLS bases. For both feature type, high recognition rates are achieved starting from 30 bases. There is no significant advantage of using one distance type over another.

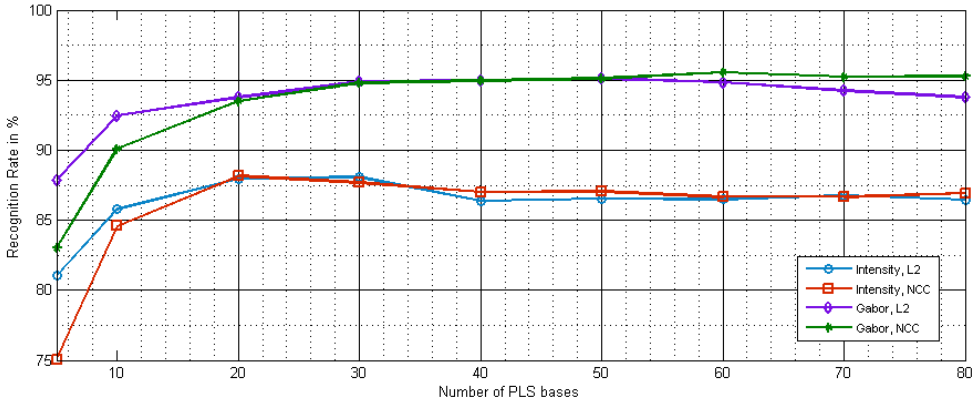

Figure 4.9. Results of two different distance types for both local blocks of Gabor features and intensity values with changing number of PLS bases.

Table 4.2. Results for all input and output pose pairs by using intensity features.

g/p	90l	75l	60l	45l	30l	15l	0	15r	30r	45r	60r	75r	90r	Avg.
90l	-	97.7	91.8	75.4	65.6	58.0	50.8	52.1	51.7	50.8	54.6	53.8	51.4	62.8
75l	99.3	-	99.8	93.1	85.3	76.7	67.2	65.1	61.4	63.8	65.2	66.5	59.1	75.2
60l	94.9	99.1	-	98.1	93.8	88.0	80.7	74.3	71.0	70.1	72.9	69.7	62.5	81.2
45l	81.9	94.6	98.3	-	98.8	96.7	87.9	80.1	72.1	68.9	71.6	67.4	60.2	81.5
30l	75.1	89.0	95.6	99.7	-	98.9	95.8	88.3	77.7	72.2	74.5	66.9	57.1	82.5
15l	72.1	84.6	91.3	98.9	100	-	99.8	97.6	91.1	81.9	77.0	69.2	56.4	84.9
0	64.0	76.9	85.2	92.7	98.0	100.0	-	99.9	98.0	91.4	85.3	75.7	63.0	85.8
15r	58.0	70.8	77.0	80.9	89.4	97.4	99.8	-	100	98.3	91.8	82.8	67.1	84.4
30r	56.7	68.3	72.0	72.0	78.1	87.9	95.2	99.0	-	99.5	94.8	87.4	73.0	81.9
45r	61.0	68.4	73.5	72.2	73.3	80.2	87.7	97.0	99.5	-	97.9	94.1	81.1	82.1
60r	60.7	68.1	73.4	69.0	70.1	73.3	79.9	87.7	94.3	98	-	99.9	95.7	80.8
75r	56.7	64.5	64.5	61.3	59.0	61.9	67.4	74.2	81.6	91.3	99.7	-	99.6	73.4
90r	52.1	53.8	54.4	52.1	50.2	51.6	55.2	58.2	61.4	71.3	91.3	98.2	-	62.4
Avg.	69.3	77.9	81.4	80.4	80.1	80.8	80.6	81.1	79.9	79.7	81.3	77.6	68.8	78.4

4.3.3. Cross-pose Recognition Results

Inside a test set, there are 13 different poses as projections of a 3D model from 90 degree left to 90 degree right with 15 degree intervals. We relate an expression image of a subject from one pose to another pose by using the PLS method. This relation shows how well an expression from a viewpoint can be recognized by matching expressions from another viewpoint, if we exclude the subject differences.

In this section, we evaluate our method by obtaining recognition rates for each pose pair. Results for each pose pair by using the intensity values of local blocks as features can be seen from Table 4.2 and Gabor features can be seen from Table 4.3. These results show that expressions of a subject from different poses are projected into a space in which they remain closer to each other than other expressions of the subject despite the differences caused by the pose change. We also see that overall performance of Gabor features only outperform intensity features significantly when using non-frontal poses as gallery. Using local Gabor features, we achieve a correct recognition rate of 86.6% when all pose pairs are considered, and 87.6% when only the frontal gallery pose is considered.

It is clear from the results in Table 4.2 and 4.3 that pose pairs whose angles are close to each other are likely to produce higher recognition rates. Therefore, table elements are higher as they get close to the diagonal. Although tables are not exactly

Table 4.3. Results for all input and output pose pairs by using Gabor features.

g/p	90l	75l	60l	45l	30l	15l	0	15r	30r	45r	60r	75r	90r	Avg.
90l	-	96.8	89.0	81.4	75.4	70.4	65.1	66.6	70.7	72.3	70.9	75.8	79.0	76.1
75l	97.8	-	98.7	95.0	90.1	83.0	73.2	75.9	78.3	81.6	84.3	85.4	76.5	84.9
60l	92.2	99.6	-	99.6	97.4	93.5	85.0	86.3	86.6	89.5	90.8	84.5	74.0	89.9
45l	84.2	95.6	99.5	-	99.9	97.8	91.5	90.4	91.9	92.7	86.5	80.8	72.5	90.2
30l	78.3	90.7	97.3	100	-	99.9	98.0	96.1	94.9	92.0	84.6	78.8	70.9	90.1
15l	73.0	84.5	93.7	99.1	99.9	-	99.8	98.9	97.1	92.1	84.6	78.2	69.8	89.2
0	69.7	78.7	87.2	94.2	99.1	99.9	-	99.9	98.4	94.6	85.6	76.5	68.1	87.6
15r	69.1	77.7	83.6	91.1	96.6	99.4	99.9	-	99.9	98.1	92.5	83.7	74.0	88.8
30r	69.2	77.7	84.6	90.7	94.0	95.1	97.8	100	-	99.9	97.9	89.7	79.8	89.7
45r	73.3	79.4	86.5	91.8	91.5	90.6	92.8	96.9	99.6	-	99.9	96.0	84.9	90.2
60r	73.8	84.4	91.5	89.2	85.6	85.0	84.8	91.2	97.3	99.8	-	99.7	93.6	89.6
75r	74.7	85.9	85.2	80.6	77.4	73.5	73.5	80.8	87.4	94.0	99.4	-	98.6	84.2
90r	79.6	76.3	73.3	71.0	69.2	65.8	64.5	66.8	74.6	79.3	90.5	98.4	-	75.7
Avg.	77.9	85.6	89.1	90.3	89.6	87.8	85.4	87.4	89.7	90.4	88.9	85.6	78.4	86.6

symmetric due to the stopping criterion in the NIPALS algorithm, recognition rates of the symmetric pose pairs are close to each other as expected as a consequence of the symmetric modeling of the input and output matrices in the algorithm. There is an obvious decrease in the recognition rates when input and output poses have opposite signed angles and at least one pose has a large angle ($|\phi| \geq 60$) that makes an eye occluded and consequently unavailable for the comparisons. In that case, decision is made by using the present eye block and mouth block. If both poses in the pose pair have large pose angles, then present eye blocks are matched by using the symmetric property of the left and right eyes as explained in Section 3.4.1. This prevents the decrease in recognition rates of these pairs, which would be the result of using only the mouth block for the classification.

4.3.4. Results for all Intensity Levels

There are four different intensity levels for each expression in BU3DFE as shown in Figure 4.2. In this section, we repeat our experiments for each intensity level by averaging the results of using frontal pose as the input pose and every other pose as the output pose. Here we used Gabor features with the optimal parameters discussed in Section 4.3.2.

According to Table 4.4, recognition rates are higher for higher intensity levels as

Table 4.4. Results for all intensity levels by using Gabor features.

intensity	90l	75l	60l	45l	30l	15l	15r	30r	45r	60r	75r	90r
1	63.4	69.5	83.9	89.1	96.5	100	99.5	97.3	89.5	80.8	73.0	61.7
2	73.3	74.6	89.5	94.7	99.5	100	100	98.6	92.5	88.2	80.3	72.0
3	75.3	83.1	92.2	97.4	98.7	100	100	99.1	95.6	91.7	84.8	80.9
4	84.3	90.0	96.1	98.6	99.5	100	100	99.5	96.9	96.1	92.1	85.2

Table 4.5. Results for matching expressions of unknown subjects for all input and output pose pairs by using Gabor features.

g/p	90l	75l	60l	45l	30l	15l	0	15r	30r	45r	60r	75r	90r	Avg.
90l	-	49.9	51.0	48.6	49.4	50.7	47.0	45.7	51.3	48.6	48.3	44.9	45.4	48.4
75l	51.3	-	55.1	52.5	55.7	51.9	51.0	51.6	52.5	51.9	51.2	49.1	47.1	51.7
60l	52.6	50.9	-	57.4	54.5	54.3	50.1	52.2	55.7	56.2	53.3	50.4	52.0	53.3
45l	51.7	55.8	55.7	-	54.9	54.8	53.3	53.9	54.1	56.1	57.0	49.6	49.6	53.8
30l	49.4	55.4	55.9	58.3	-	56.4	53.9	54.2	58.0	57.4	55.1	53.9	49.1	54.7
15l	50.4	47.7	53.2	53.9	54.6	-	53.9	53.5	54.1	52.6	55.1	47.1	46.5	51.8
0	46.2	49.2	52.7	54.4	52.7	52.7	-	53.9	53.8	53.73	51.7	47.9	45.9	51.2
15r	44.9	51.4	51.2	56.4	55.2	52.3	53.0	-	53.9	53.0	54.8	49.6	48.8	52.0
30r	49.6	53.0	59.3	59.0	58.8	57.4	55.4	55.4	-	57.8	55.4	52.6	45.9	54.9
45r	50.7	51.2	52.2	55.1	59.4	58.0	55.4	56.4	57.7	-	57.1	53.3	49.6	54.6
60r	52.0	52.8	56.1	57.7	58.1	56.5	53.8	54.5	56.8	55.7	-	54.6	49.0	54.8
75r	46.1	52.0	51.6	53.5	55.2	53.2	49.9	50.0	51.0	49.9	52.0	-	45.7	50.8
90r	42.9	47.7	47.0	47.8	47.7	48.7	48.0	48.4	46.1	46.8	49.3	46.8	-	47.2
Avg.	48.98	51.4	53.4	54.5	54.6	53.9	52.0	52.4	53.7	53.3	53.3	49.9	47.8	52.2

expected. Consequently, we show that it is harder to recognize expressions with low intensity values across poses.

4.3.5. Results for Unknown Subjects

We also evaluate our method in terms of matching expressions of unknown subjects across poses. In that case, gallery and probe are composed of expressions of different subjects, that is gallery and probe are from different sets. Recognition rates of all pose pairs for this setup are shown in Table 4.5. Average recognition rates in Table 4.3 are much better compared to the average recognition rates in Table 4.5.

5. CONCLUSION

In this thesis, we presented an approach to recognize an individual's expressions across different pose angles. Subject differences and pose variations are the two hardest aspects of automatic facial expression recognition. Each subject might perform an expression differently and pose changes cause significant changes on the appearance of the face. We keep the effect of subject differences constant and model the changes caused by the pose on the facial expression.

The proposed approach first aligns face images by using two point correspondences between the input face image and the aligned face image. The success of matching features extracted from two different faces requires them to refer to the same areas on the face, therefore it highly depends on the alignment step. All faces have the same eye row, mouth row, and eye distance thanks to the alignment we used. The alignment works in all pose angles with a small set of parameters.

The second step is the feature extraction step. We extracted local blocks of features around the visible eyes and mouth to represent faces. We experimented with direct intensity values and Gabor wavelets as feature representations. We showed that size and location of the extracted blocks affect the performance and parameters might differ for different feature representations. We extracted blocks of three different sizes and showed that a small block size might miss some information related to the expression, while larger block sizes might cover some of the background besides the larger area on the face. This extra information prevents the largest block size from achieving the highest recognition rates in case of intensity features. This might also cause a problem for the Gabor representation in case of non-uniform background. We experimented with a set of different values for the mouth offset parameter to find the parameter that gives the optimal position for the mouth block. We found that using the mouth offset parameter is essential for a good performance, especially for large pose angles. We also searched for the optimal parameters in the Gabor representation and found out that default parameters used in other facial image analysis problems the

literature do not achieve the best results in our case. The best results are achieved by using $k_{max} = \frac{\pi}{0.5}$ and $\sigma = 1.0\pi$.

In our experiments, we showed that using local blocks of intensity values performs almost as well as local blocks of Gabor features for the cases in which input pose is the frontal pose. However, Gabor features outperform intensity features significantly for other pose pairs. For all of the cases, results are comparable to the baseline results. This shows that cross-pose recognition might be a good alternative for pose-specific systems in multi-view facial expression recognition.

Throughout our experiments, we used a set of different number of PLS bases. We found that even a small number of bases might show comparable performance with large number of bases. During training, we couple face images by both identity and expression. The only difference between the face images that we try to match is the pose angle; therefore, even small bases are able to achieve a good performance. We specifically learn bases that model the changes between two pose angles free from any other difference during training.

In our experiments, we reported results for the highest intensity level only, but we have also experimented with different intensity levels available in the dataset. With these experiments, we showed that the PLS method is also good at matching expressions of different poses with lower intensity levels. Still, the recognition rates are higher for high intensity levels compared, as expected.

We lastly performed experiments by excluding identity information from the test and matched expressions of different subjects to measure the success of the method when the subject information is unavailable. Average recognition rates are lower in that case compared to the previous case where subject information is used during testing. This shows that results are significantly improved by the utilization of the identity information. This is the result of the coupling scheme we used for learning the PLS bases. In training, corresponding samples in input and output matrices are the expressions of the same subject from two different poses. This kind of learning causes

PLS bases to specifically model a subject's expressions across poses. This shows that the PLS method is very good at generalizing a subject's expressions over different pose angles. This might seem like a person-specific method, but it is not. Person-specific methods train a separate model for each subject and use this model for testing samples belonging to that subject. Here, we have only one PLS model that can classify any subject's expressions across poses with high recognition rates, as long as that subject's expressions are present in the gallery set.

Following that discussion, we list our future work as follows: First of the future work is the improvement of the results for unknown subjects by using more powerful representations such as SIFT features or using more appropriate formulations of the PLS method to handle the case where expressions of the test subject from the input pose are not available or subject information is not available. We might use state-of-the art multi-view facial recognition algorithms to first recognize identity and then classify its expression. Secondly, the current dataset is not sufficient for thoroughly testing the capabilities of the proposed approach. Expressions are taken at the consecutive time periods, therefore there is no change in the lightening or any other condition. We believe that the proposed approach can handle some changes in the illumination or small changes in the appearance of the expression face. Our next step is experimenting with a more challenging dataset.

REFERENCES

1. Ekman, P. and W. Friesen, "Pictures of Facial Affect", *Consulting Psychologists*, 1976.

2. Ekman, P., "Facial Expression and Emotion", *American Psychologist*, Vol. 48, pp. 384–392, 1993.

3. Ekman, P. and W. Friesen, "The Facial Action Coding System: A Technique for the Measurement of Facial Movement", *Consulting Psychologists*, 1978.

4. Ekman, P., W. Friesen and J. Hager, "The Facial Action Coding System", *Research Nexus eBook*, 2002.

5. De la Torre, F. and J. F. Cohn, *Guide to Visual Analysis of Humans: Looking at People*, chap. Facial Expression Analysis, Springer, 2011.

6. Cohn, J., K. Schmidt, R. Gross and P. Ekman, "Individual Differences in Facial Expression: Stability over Time, Relation to Self-reported Emotion, and Ability to Inform Person Identification", *Proceedings of the International Conference on Multimodal User Interfaces*, 2002.

7. Tian, Y.-l., T. Kanade and J. F. Cohn, "Recognizing Action Units for Facial Expression Analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, pp. 97–115, 2001.

8. Bettadapura, V., "Face Expression Recognition and Analysis: The State of the Art", *Computing Research Repository*, Vol. 1203.6722, 2012.

9. Zhihong, Z., M. Pantic, G. Roisman and T. Huang, "A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, No. 1, pp. 39 –58, 2009.

10. Hu, Y., Z. Zeng, L. Yin, X. Wei, J. Tu and T. Huang, "A Study of Non-frontal-view Facial Expressions Recognition", *19th International Conference on Pattern Recognition (ICPR 2008)*, pp. 1 –4, 2008.

11. Hu, Y., Z. Zeng, L. Yin, X. Wei and X. Zhou, "Multi-view facial expression recognition", *Face and Gesture Recognition*, pp. 1–6, IEEE, 2008.

12. Moore, S. and R. Bowden, "Local Binary Patterns for Multi-view Facial Expression Recognition", *Computer Vision and Image Understanding*, Vol. 115, No. 4, pp. 541–558, 2011.

13. Zheng, W., H. Tang, Z. Lin and T. S. Huang, "A Novel Approach to Expression Recognition from Non-frontal Face Images", *12th IEEE International Conference on Computer Vision (ICCV 2009)*, ICCV, pp. 1901–1908, IEEE, 2009.

14. Zheng, W., H. Tang, Z. Lin and T. Huang, "Emotion Recognition from Arbitrary View Facial Images", *The 11th European Conference on Computer Vision (ECCV 2010)*, pp. 490–503, Springer, 2010.

15. Soyel, H. and H. Demirel, "Improved SIFT Matching for Pose Robust Facial Expression Recognition", *International Conference on Automatic Face Gesture Recognition and Workshops (FG 2011)*, pp. 585 –590, IEEE, 2011.

16. Hesse, N., T. Gehrig, H. Gao and H. K. Ekenel, "Multi-view Facial Expression Recognition using Local Appearance Features", *21st International Conference on Pattern Recognition (ICPR 2012)*, IEEE, 2012.

17. Rudovic, O., I. Patras and M. Pantic, "Regression-based Multi-view Facial Expression Recognition", *20th International Conference on Pattern Recognition (ICPR 2010)*, pp. 4121–4124, IEEE, 2010.

18. Li, A., S. Shan, X. Chen and W. Gao, "Cross-pose Face Recognition based on Partial Least Squares", *Pattern Recognition Letters*, Vol. 32, No. 15, pp. 1948 –

1955, 2011.

19. Prince, S. J., J. H. Elder, J. Warrell and F. M. Felisberti, "Tied Factor Analysis for Face Recognition across Large Pose Differences", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30, pp. 970–984, 2008.

20. Jacobs, D. W., A. Kumar, H. Daume and A. Sharma, "Generalized Multiview Analysis: A Discriminative Latent Space", *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2160–2167, 2012.

21. Li, A., S. Shan and W. Gao, "Coupled Bias-Variance Tradeoff for Cross-Pose Face Recognition", *IEEE Transactions on Image Processing*, Vol. 21, No. 1, pp. 305 –315, 2012.

22. Fischer, M., H. K. Ekenel and R. Stiefelhagen, "Analysis of Partial Least Squares for Pose-Invariant Face Recognition", *IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems*, 2012.

23. Geladi, P. and B. Kowalski, "Partial Least-Squares Regression: a Tutorial", *Anal. Chim. Acta*, Vol. 185, No. C, pp. 1–17, 1986.

24. Bro, R., "Multiway Calibration. Multilinear PLS", *Journal of Chemometrics*, Vol. 10, pp. 47–61, 1996.

25. Sharma, A. and D. W. Jacobs, "Bypassing Synthesis: PLS for Face Recognition with Pose, Low-resolution and Sketch", *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pp. 593–600, IEEE, Washington, DC, USA, 2011.

26. Chengjun, L. and H. Wechsler, "Gabor Feature based Classification using the Enhanced Fisher Linear Discriminant Model for Face Recognition", *IEEE Transactions on Image Processing*, Vol. 11, No. 4, pp. 467 –476, 2002.

27. Wegelin, J. A. and T. Thanks, *A Survey of Partial Least Squares (PLS) Meth-*

ods, with Emphasis on the Two-Block Case, Tech. rep., Department of Statistics, University of Washington, 2000.

28. Wold, H., *International Perspectives on Mathematical and Statistical Modeling, Quantitative Sociology.*, chap. Path Models Latent Variables: The NIPALS Approach, pp. 307–357, Academic Press, New York, 1975.

29. Höskuldsson, A., "PLS Regression Methods", *J. Chemometrics*, Vol. 2, No. 3, 1988.

30. Roman, R. and K. Nicole, "Overview and Recent Advances in Partial Least Squares", *Lecture Notes in Computer Science in 'Subspace, Latent Structure and Feature Selection Techniques'*, pp. 34–51, Springer, 2006.

31. Yin, L., X. Wei, Y. Sun, J. Wang and M. J. Rosato, "A 3D Facial Expression Database For Facial Behavior Research", *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pp. 211–216, IEEE, 2006.