Cross-pose Facial Expression Recognition

Fatma Guney

Master Thesis

September 12, 2012

Computer Engineering Department
Bogazici University, Turkey
Automatic Facial Expression Recognition

- is an active research topic.
- finds interesting applications in several areas.
- has two types of class categories:
  - 6 prototypic emotional expressions: happiness, surprise, anger, sadness, fear, and disgust
  - Facial Action Coding System (FACS) [1], [2]

We follow the first class category definition.
Automatic Facial Expression Recognition can be a very challenging task due to the:

- subject differences
  - such as texture of the skin, hair style, age, gender, and ethnicity
  - or expressiveness
- pose variations
  - a non-linear transformation of the 2D face image
  - very different appearance from different viewpoints
  - self-occluded areas
Motivation

We focus on multi-view facial expression recognition.

- Existing studies discretize the viewpoints into a set of intervals and use a separate model for each viewpoint
  - that functions as a recognizer for a particular pose angle
  - that needs representative data from that pose angle for its training

- Problem in data collection
  - Most of the available datasets are frontal or near frontal views.
  - Natural to show expressions from frontal view during a recording.
Our motivation is developing a pose-independent expression recognition system without the need for data from all viewpoints.

- This requires establishing a relation between expressions of an individual from different viewpoints.
- We realize that by learning a mapping from one pose to another.
## Literature Review

### Table: Multi-view Facial Expression Recognition in the Literature

<table>
<thead>
<tr>
<th>Study</th>
<th>Scheme</th>
<th>Features</th>
<th>Dim. Reduction</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>[5]</td>
<td>pose-specific</td>
<td>2D displacement vectors</td>
<td>PCA, LDA, LPP</td>
<td>QBN, Parzen, SVM, knn</td>
</tr>
<tr>
<td>[6]</td>
<td>two-step and composite</td>
<td>Hog, LBP, SIFT</td>
<td>PCA, LDA, LPP</td>
<td>NN</td>
</tr>
<tr>
<td>[12]</td>
<td>two-step</td>
<td>variants of LBP, LGBP</td>
<td>-</td>
<td>multi-class SVMs</td>
</tr>
<tr>
<td>[22]</td>
<td>-</td>
<td>SIFT</td>
<td>PCA</td>
<td>minimizing Bayes error</td>
</tr>
<tr>
<td>[21]</td>
<td>-</td>
<td>dense SIFT</td>
<td>a new discriminant analysis</td>
<td>RCM</td>
</tr>
<tr>
<td>[17]</td>
<td>-</td>
<td>SIFT</td>
<td>-</td>
<td>πSIFT</td>
</tr>
<tr>
<td>[4]</td>
<td>two-step, AAMs</td>
<td>SIFT, DCT</td>
<td>F-score</td>
<td>multi-class SVMs</td>
</tr>
<tr>
<td>[15]</td>
<td>mapping</td>
<td>coordinates of landmarks</td>
<td>-</td>
<td>LR, SVR, RVR, GPR</td>
</tr>
</tbody>
</table>
Problem: Pose variations separate the faces into several different subspaces [8].

Solution: Find pose-independent latent spaces.

- Tied factor analysis to find a latent identity space for each pair of poses [13].
  The model describes how the identity as an underlying factor created the appearance changed by the pose.

- PLS to project faces of an individual from two different poses into a common linear subspace [16].
  They apply it to other cross modality problems such as sketch-photo recognition and variations in resolution.

- In [7], same authors extend their approach to generalized multi-view analysis.
  They extend PCA, LDA, LPP, Neighborhood Preserving Embedding (NPE), and Marginal Fisher Analysis (MFA) into the multi-view case.
[9] 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 [8], they use PLS to find pose-independent feature vectors for face recognition.

Outline

- Background for Latent Variable Methods
  - Statistical Concepts and Notations
  - A review of Regression Methods
  - Principal Components Analysis and Regression

- Cross-pose Facial Expression Recognition based on PLS
  - Alignment
  - Feature Extraction
  - PLS
  - Cross-pose Facial Expression Recognition using PLS

- Experiments and Results
  - Data
  - Experimental Setup
  - Results

- Conclusion
Background for Latent Variable Methods
Latent Variable Methods

- Extracting value from data
  - Learning from data
  - Prediction
- Principal Components Analysis (PCA)
- Projection to Latent Structures (PLS)
Statistical Concepts and Notations

Mean

\[ \bar{x} = \frac{1}{n} (x_1 + x_2 + \ldots + x_n) = \frac{1}{n} \mathbf{1}^T \mathbf{x} \]

\[ \mathbf{x}_{\text{centered}} = \mathbf{x} - \mathbf{1} \bar{x} \]

Variance

\[ s_x^2 = \frac{1}{n-1} \left\{ (x_1 - \bar{x})^2 + \ldots + (x_n - \bar{x})^2 \right\} \]

\[ = \frac{1}{n-1} ||\mathbf{x} - \mathbf{1} \bar{x}||^2 \]

For a centered vector \( \mathbf{x} \), \( \bar{x} = 0 \) and \( s_x^2 \) becomes:

\[ s_x^2 = \frac{1}{n-1} ||x||^2 \]

\[ = \frac{1}{n-1} \mathbf{x}^T \mathbf{x} \]
Statistical Concepts and Notations

Standard deviation

\[ s_x = \sqrt{\frac{1}{n-1} \left\{ (x_1 - \bar{x})^2 + \ldots + (x_n - \bar{x})^2 \right\}} \]

\[ x_{scaled} = \frac{1}{s_x} x \]

Covariance

\[ v_{xy} = \frac{1}{n-1} \left\{ (x_1 - \bar{x})(y_1 - \bar{y}) + \ldots + (x_n - \bar{x})(y_n - \bar{y}) \right\} \]

\[ = \frac{1}{n-1} (x - 1\bar{x})^T (y - 1\bar{y}) \]

For centered vectors \( x \) and \( y \), \( v_{xy} \) becomes:

\[ v_{xy} = \frac{1}{n-1} x^T y \]

\[ V_{XY} \propto X^T Y \]
Correlation

\[ r_{xy} = \frac{v_{xy}}{s_x s_y} \]

\[-1 \leq r_{xy} \leq 1\]

**Rank**  
maximum number of linearly independent columns (or rows) in a matrix.  
It is a number expressing true underlying dimensionality of a matrix [3].
Least Squares (Simple Linear Regression)

\[ y = 1b_0 + xb + f \]  \hspace{1cm} (1)

The least squares estimators for \( b_0 \) and \( b \) can be obtained by minimizing the sum of squared errors (SSE):

\[ \hat{b}_0 = \bar{y} \]
\[ \hat{b} = \frac{x^T y}{x^T x} = \frac{v_{xy}}{s_x^2} \]

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

\[ y = 1\bar{y} + xb + f \]
\[ y - 1\bar{y} = xb + f \]

Final equation becomes:

\[ y = xb + f \]  \hspace{1cm} (2)
Multiple Linear Regression (MLR)

\[ y_i = b_0 + x_{i1}b_1 + x_{i2}b_2 + \ldots + x_{ik}b_k + f_i \]

Let \( X \) be a \( n \times k \) matrix and \( b \) be a vector of \( k \times 1 \):

\[ y = 1b_0 + Xb + f \]  

(3)

By minimizing SSE, the least squares estimators:

\[ \hat{b}_0 = \bar{y} \]
\[ \hat{b} = (X^TX)^{-1}X^Ty \]
\[ = V_x^{-1}V_{Xy} \]

After inserting the value of \( b_0 \), Equation 3 becomes:

\[ y = Xb + f \]  

(4)
Multivariate MLR

When $Y$ is also a matrix of size $n \times m$, MLR model holds for each column of $Y$:

$$y_l = Xb_l + f_l$$

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

$$Y = XB + F$$

The least squares estimators for this model is:

$$\hat{b}_0 = \bar{y}$$

$$\hat{B} = (X^TX)^{-1}X^TY$$

$$= V^{-1}_X V_{XY}$$
Principal Components Analysis (PCA)

PCA finds the best summary of data $X$ with the fewest number of summary variables, called scores, $T$.

There are a few ways of calculating the PCA model:

- Eigenvalue Decomposition
- Singular Value Decomposition (SVD)
- Non-linear Iterative Partial Least-Squares (NIPALS) algorithm
PCA can be seen as a method of writing data matrix $X$ of rank $r$ as a sum of $r$ matrices of rank 1 [3]:

$$X = M_1 + M_2 + \ldots + M_r$$  \hspace{1cm} (5)

These rank 1 matrices can be written as products of two vectors: a score $t_i$ and a loading $p_i$:

$$X = t_1 p_1^T + t_2 p_2^T + \ldots + t_r p_r^T$$

$$X = TP^T$$
PCA - From the Perspective of Loadings and Scores

The principal component is the best-fit line for the data points.

$p_i$: $p_1$ and $p_2$ are direction cosines

$t_i$: coordinates of the respective points on the principal component direction.

$p_1 = \cos \theta_1$

$p_2 = \cos \theta_2$
The NIPALS algorithm iteratively calculates scores $T$ and loadings $P$ such that

$$X = TP^T + F$$

$$F_1 = X - t_1 p_1^T$$

$$F_2 = F_1 - t_2 p_2^T$$

$$\ldots$$

$$F_i = F_{i-1} - t_i p_i^T$$

(6)
NIPALS Algorithm for PCA

1. Choose $t_j$ as any column of $X$
2. Calculate $p_j$ by projecting columns of $X$ onto $t_j$: $p_j = X^T t_j / t_j^T t_j$
3. Normalize $p_j$ to length 1: $p_j = p_j / ||p_j||$
4. Calculate $t_j$ by projecting rows of $X$ onto $p_j$: $t_j = X p_j / p_j^T p_j$
5. Compare $t_j$ with $t_{j-1}$. If it is unchanged (or not changed significantly) stop, else continue with step 2.

After a component is calculated, $X$ in the algorithm is replaced by its residual $F$ as in Equation 6.
Let’s replace scalars $t_j^T t_j$ and $p_j^T p_j$ by a general constant term $C$:

$$Cp_j = X^T t_j$$  \hspace{1cm} (7)  \\
$$Ct_j = Xp_j$$  \hspace{1cm} (8) \\

We can substitute Equation 8 into Equation 7:

$$Cp_j = X^T Xp_j$$  \hspace{1cm} (9)  \\

Equation 9 is the eigenvalue equation for $X^T X$. Similarly, when we substitute 7 into 8, we obtain eigenvalue equation for $XX^T$. 

NIPALS Algorithm for PCA
Principal Components Regression (PCR)

Idea: $X$ can be represented by its score matrix $T$:

$$T = XP$$

$T$ has a smaller dimensionality and it still retains important features of $X$. Then MLR can be performed with $T$ in place of $X$:

$$Y = TB + F$$

The regression coefficient matrix becomes:

$$\hat{B} = (T^T T)^{-1} T^T Y$$
Cross-pose Facial Expression Recognition based on PLS
Steps of the Proposed Approach

- Alignment
- Feature Extraction
- Projection to Latent Space by using PLS
- Classification
Alignment
The Point Correspondences

As in [11], by using visible eyes, $s_{l.\text{eye}}$ and/or $s_{r.\text{eye}}$, and mouth center $s_{\text{mouth}}$ in the input image, we compute a similarity transform $T$ from two point correspondences between the input image and the aligned image:

1. mouth center

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

2. point between the eyes

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

   or the visible eye for some poses

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

Figure: Illustration of the alignment parameters
Computing the Transformation Matrix

The homogeneous transformation matrix, $T$ is computed by solving the system of linear equations given by the two point correspondences:

\[
\begin{align*}
\hat{s}_1 &= T\hat{t}_1 \\
\hat{s}_2 &= T\hat{t}_2
\end{align*}
\]

where $\hat{u} = (u_x \ u_y \ 1)^T$ denotes the homogeneous coordinates of $u$, for $u \in \{s_1, s_2, t_1, t_2\}$. 
Feature Extraction
Gabor Features

Gabor wavelet formulation:

\[
\psi(\vec{x}; \nu, \mu) = \frac{k_{\nu,\mu}^2}{\sigma^2} e\left(-\frac{k_{\nu,\mu}^2 \|\vec{x}\|^2}{2\sigma^2}\right) \left[e^{ik_{\nu,\mu}\vec{x}} - e^{-\frac{\sigma^2}{2}}\right] (10)
\]

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

\[
k_{\nu,\mu} = \frac{k_{\text{max}}}{f\nu} e^{i\frac{\pi \mu}{8}} (11)
\]

where \(k_{\text{max}}\) is the maximum frequency and \(f\) is the spacing factor between kernels in the frequency domain [10].

Default values: five scales \(\nu \in \{0, 1, 2, 3, 4\}\) and eight orientations \(\mu \in \{0, 1, \ldots, 7\}\) with \(f = \sqrt{2}, k_{\text{max}} = \frac{\pi}{2}\), with Gaussian size \(\sigma = 2\pi\)
Extraction of Local Blocks

- Local face representations vs. holistic representations
- Local blocks of size $w_b \times h_b$ around left eye, right eye and mouth on GMIs
- To handle the background pixels inside the blocks:
  - discard the eye block unless it is clearly seen in the non-frontal aligned face image
  - compute a horizontal offset $\Delta_{mou}th$ to shift the mouth block [11]:
    \[
    \Delta_{mou}th = -f_{mou}th \ w_{mou}th \ \sin(\phi)
    \]  
    (12)
    where $f_{mou}th$ is the mouth coefficient, and $\phi$ is the pose angle of the face.
Partial Least Squares (PLS)
or
Projection to Latent Scores (PLS)
"a class of techniques for modeling relations between blocks of observed variables by means of latent variables"

originated by the studies of Herman Wold in the 1970's

two modes of PLS, called A and B [18]

two or more data blocks

We use term PLS for two block Mode A PLS which is a special case of Wold’s algorithm [19].
PCA and PLS

- Principal components maximize the ability to describe the covariance or spread of the data in $X$:

  $X^TX$

- Problem: Principal variation in $X$ is not guaranteed to yield latent features that are good for predicting $Y$.

- PLS’s solution: Project to latent variables that maximize the covariation between $X$ and $Y$:

  $X^TY$
Centered and scaled $n \times N$ input matrix $X$ and $n \times M$ output matrix $Y$ are decomposed into the form [14]:

$$X = TP^T + E$$  \hspace{1cm} (13)
$$Y = UQ^T + F$$  \hspace{1cm} (14)

- $T, U$ 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
- $n \times N$ matrix $E$ and $n \times M$ matrix $F$ are the matrices of residuals
PLS models $X^T Y$ that is covariance of $X$ and $Y$ by means of latent variables [18]:

- $X$ and $Y$ are considered as indicators of $p$ latent variables, or scores, $t$ and $u$, respectively.
- PLS models the cross-covariance by pairs of these scores:

$$ (t_1, u_1), ..., (t_p, u_p) $$

(15)

- Sets $\{t_1, ..., t_p\}$ and $\{u_1, ..., u_p\}$ are computed as the best representative column spaces of $X$ and $Y$ for $X^T Y$. 
NIPALS Algorithm for PLS

NIPALS maximizes the squares of covariance between the score vectors \( t \) and \( u \) by finding weight (basis) vectors \( w \) and \( c \) such that [14]:

\[
[cov(t, u)]^2 = [cov(Xw, Yc)]^2 = \max_{|r| = |s| = 1} [cov(Xr, Ys)]^2 \quad (16)
\]

where \( cov(t, u) = t^T u / n \) denotes the sample covariance between score vectors \( t \) and \( u \).
NIPALS Algorithm for PLS

It starts with random initialization of $u$ and iteratively computes these steps until convergence:

1. $w = \frac{X^T u}{(u^T u)}$
2. $||w|| = 1$
3. $t = Xw$
4. $c = \frac{Y^T t}{(t^T t)}$
5. $||c|| = 1$
6. $u = Yc$
NIPALS Algorithm for PLS

- PLS is an iterative process. After the extraction of score vectors $t$ and $u$, $X$ and $Y$ are deflated and residuals are calculated for the next iteration.

- Loadings are computed as coefficients of regressing $X$ on $t$ and $Y$ on $u$:

  $$ p = \frac{X^T t}{(t^T t)} $$

  $$ q = \frac{Y^T u}{(u^T u)} $$

- $X$ and $Y$ are deflated in a symmetric way:

  $$ X = X - tp^T $$

  $$ Y = Y - uq^T $$
Projection to Latent Space

- **PLS**: represent $X$ and $Y$ by their corresponding score vectors $t$ and $u$.

\[ W = [w_1 \ w_2 \ \ldots \ w_p] \]
\[ C = [c_1 \ c_2 \ \ldots \ c_p] \]

- Projections:

\[ \hat{x} = xW \] (17)
\[ \hat{y} = yC \] (18)
Cross-pose Facial Expression Recognition using PLS
Problem Formulation

- differences across subjects + variations caused by different poses
- We use faces of the same subject
  - to exclude variations caused by identity
  - to reduce the problem to modeling of the variance in expressions caused by pose changes
Problem Formulation

We use PLS to compute a latent space for each block pair in two different poses:
Problem Formulation

For a pose pair \((p_i, p_j)\), we construct input \(X\) and output \(Y\) matrices:

\[
\begin{align*}
    x \in p_i & \quad X = [x_1; x_2; \ldots; x_n] \\
y \in p_j & \quad Y = [y_1; y_2; \ldots; y_n]
\end{align*}
\]

- Corresponding samples \(x_i\) and \(y_i\) are coupled by both identity and expression.
- PLS computes projections \(W\) and \(C\) that maximize the covariance of score vectors, so covariance between different poses of an expression is maximized.
Problem Formulation

In faces with large pose angles, one of the eyes is not visible.

Problem:
- One pose has a large negative 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.

Solution:
- Assume that left and right eye are sufficiently symmetric [11].
- Train a PLS latent space for the opposite eye blocks.
Classification

Classification can be done using the Nearest Neighbor algorithm:

1. Dimension of the score vectors is the same, they are in the same vector space.
2. Score vectors are highly correlated, since PLS bases are learned based on a criterion that maximize the covariance between them.

Distance metrics:

- L2 distance
- Normalized Cross Correlation (NCC)

\[
d_{ncc}(\mathbf{x}, \mathbf{y}) = 1 - \frac{(\mathbf{x} - \mu_x) (\mathbf{y} - \mu_y)}{(N - 1) \sigma_x \sigma_y}
\] (19)
Experiments and Results
Data

Experiments were conducted on Binghamton University 3D Facial Expression Database (BU3DFE) [20].

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

- 100 different subjects (56 female and 44 male)
- Different age, ranging from 18 to 70 years, and a wide variety of ethnicities
- 83 annotated landmarks
- 6 basic expression types and neutral
- Four different levels of intensity from low to high

Figure: Illustration of 4 different levels of intensity for happiness class
Data

- 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:
  - Image + coordinates of the landmark points is saved.
  - 13 different poses:
    - 90l, 75l, 60l, 45l, 30l, 15l, frontal, 15r, 30r, 45r, 60r, 75r, 90r
Experimental Setup

- Data is divided into 3 sets, each containing 33 different subjects (one 34).
  - one set for learning PLS bases
  - one set for the optimization of parameters
  - one set for testing
- We use a single image for each expression which belongs to the highest intensity level (level 4).
Baseline Results

Most of the multi-view expression recognition studies train pose-specific classifiers and results are reported according to this scheme.

Table: Pose-specific recognition rates as a baseline

<table>
<thead>
<tr>
<th>feat type</th>
<th>90l</th>
<th>75l</th>
<th>60l</th>
<th>45l</th>
<th>30l</th>
<th>15l</th>
<th>0</th>
<th>15r</th>
<th>30r</th>
<th>45r</th>
<th>60r</th>
<th>75r</th>
<th>90r</th>
</tr>
</thead>
<tbody>
<tr>
<td>gabor</td>
<td>45.6</td>
<td>47.8</td>
<td>56.9</td>
<td>53.5</td>
<td>52.6</td>
<td>55.6</td>
<td>50.9</td>
<td>53.0</td>
<td>53.0</td>
<td>55.6</td>
<td>51.3</td>
<td>49.1</td>
<td>40.4</td>
</tr>
<tr>
<td>intensity</td>
<td>40.4</td>
<td>44.3</td>
<td>49.5</td>
<td>50.8</td>
<td>50.0</td>
<td>47.4</td>
<td>46.1</td>
<td>50.0</td>
<td>51.3</td>
<td>52.2</td>
<td>50.9</td>
<td>48.3</td>
<td>41.7</td>
</tr>
<tr>
<td>[4]</td>
<td>72.2</td>
<td>74.1</td>
<td>74.5</td>
<td>75.2</td>
<td>76.7</td>
<td>77.8</td>
<td>77.2</td>
<td>76.5</td>
<td>77.7</td>
<td>75.4</td>
<td>74.0</td>
<td>72.9</td>
<td>71.5</td>
</tr>
</tbody>
</table>
Effects of Parameters

- Alignment parameters
- Feature extraction details
- Number of PLS bases in the NIPALS algorithm
- Distance type used in classification

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

Optimal parameters reported in [11]:

\[ w = 104 \quad h = 128 \quad y_{eyes} = 42 \]
\[ y_{mouth} = 106 \quad d_{eyes} = 62 \quad d_{x_{mouth}} = 20 \]

Figure: Examples of aligned face images for each pose angle (first row), extracted local blocks on the aligned face image (second row).
Effects of the Block Size

We experimented with three different block sizes: $32 \times 32$, $48 \times 48$, and $64 \times 64$.

**Figure**: Visual representation of three different block sizes on two different expression faces
Figure: Effects of different block sizes for intensity values and Gabor features with changing number of PLS bases on the average recognition rate.
Effects of the Mouth Offset Parameter

The mouth offset parameter is used to change the point which corresponds to the center of the mouth block position.

Figure: Visual representation of three different mouth offset parameters for expression faces from different viewpoints.
Effects of the Mouth Offset Parameter

We experimented with a set of parameters for mouth offset parameter: 0, 0.15, 0.35, 0.55, 0.75

Figure: Effects of the mouth offset parameter for all pose angles
Effects of Gabor Parameters

- Default values may not be the optimal ones for our case.
- We performed a grid search for:
  - scaling factor $k_{max}$ in the wave vector $k_{\nu,\mu}$ which generates the Gabor kernels by scaling and rotation.
  - Gaussian window width $\sigma$

Figure: Effects of different Gabor parameters, $k_{max}$ and $\sigma$
Effects of Number of PLS bases and Distance Type

Figure: Results of two different distance types for both local blocks of Gabor features and intensity values with changing number of PLS bases.
### Cross-pose Recognition Results

**Table:** Results for all input and output pose pairs by using intensity features

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

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

<table>
<thead>
<tr>
<th>g/p</th>
<th>90l</th>
<th>75l</th>
<th>60l</th>
<th>45l</th>
<th>30l</th>
<th>15l</th>
<th>0</th>
<th>15r</th>
<th>30r</th>
<th>45r</th>
<th>60r</th>
<th>75r</th>
<th>90r</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>90l</td>
<td>-</td>
<td>96.8</td>
<td>89.0</td>
<td>81.4</td>
<td>75.4</td>
<td>70.4</td>
<td>65.1</td>
<td>66.6</td>
<td>70.7</td>
<td>72.3</td>
<td>70.9</td>
<td>75.8</td>
<td>79.0</td>
<td>76.1</td>
</tr>
<tr>
<td>75l</td>
<td>97.8</td>
<td>-</td>
<td>98.7</td>
<td>95.0</td>
<td>90.1</td>
<td>83.0</td>
<td>73.2</td>
<td>75.9</td>
<td>78.3</td>
<td>81.6</td>
<td>84.3</td>
<td>85.4</td>
<td>76.5</td>
<td>84.9</td>
</tr>
<tr>
<td>60l</td>
<td>92.2</td>
<td>99.6</td>
<td>-</td>
<td>99.6</td>
<td>97.4</td>
<td>93.5</td>
<td>85.0</td>
<td>86.3</td>
<td>86.6</td>
<td>89.5</td>
<td>90.8</td>
<td>84.5</td>
<td>74.0</td>
<td>89.9</td>
</tr>
<tr>
<td>45l</td>
<td>84.2</td>
<td>95.6</td>
<td>99.5</td>
<td>-</td>
<td>99.9</td>
<td>97.8</td>
<td>91.5</td>
<td>90.4</td>
<td>91.9</td>
<td>92.7</td>
<td>86.5</td>
<td>80.8</td>
<td>72.5</td>
<td>90.2</td>
</tr>
<tr>
<td>30l</td>
<td>78.3</td>
<td>90.7</td>
<td>97.3</td>
<td>100</td>
<td>-</td>
<td>99.9</td>
<td>98.0</td>
<td>96.1</td>
<td>94.9</td>
<td>92.0</td>
<td>84.6</td>
<td>78.8</td>
<td>70.9</td>
<td>90.1</td>
</tr>
<tr>
<td>15l</td>
<td>73.0</td>
<td>84.5</td>
<td>93.7</td>
<td>99.1</td>
<td>99.9</td>
<td>-</td>
<td>98.9</td>
<td>97.1</td>
<td>92.1</td>
<td>84.6</td>
<td>78.2</td>
<td>69.8</td>
<td>89.2</td>
<td>76.6</td>
</tr>
<tr>
<td>0</td>
<td>69.7</td>
<td>78.7</td>
<td>87.2</td>
<td>94.2</td>
<td>99.1</td>
<td>99.9</td>
<td>-</td>
<td>99.8</td>
<td>98.4</td>
<td>94.6</td>
<td>85.6</td>
<td>76.5</td>
<td>68.1</td>
<td>87.6</td>
</tr>
<tr>
<td>15r</td>
<td>69.1</td>
<td>77.7</td>
<td>83.6</td>
<td>91.1</td>
<td>96.6</td>
<td>99.4</td>
<td>99.9</td>
<td>-</td>
<td>99.9</td>
<td>98.1</td>
<td>92.5</td>
<td>83.7</td>
<td>74.0</td>
<td>88.8</td>
</tr>
<tr>
<td>30r</td>
<td>69.2</td>
<td>77.7</td>
<td>84.6</td>
<td>90.7</td>
<td>94.0</td>
<td>95.1</td>
<td>97.8</td>
<td>100</td>
<td>-</td>
<td>99.9</td>
<td>97.9</td>
<td>89.7</td>
<td>79.8</td>
<td>89.7</td>
</tr>
<tr>
<td>45r</td>
<td>73.3</td>
<td>79.4</td>
<td>86.5</td>
<td>91.8</td>
<td>91.5</td>
<td>90.6</td>
<td>92.8</td>
<td>96.9</td>
<td>99.6</td>
<td>-</td>
<td>99.9</td>
<td>96.0</td>
<td>84.9</td>
<td>90.2</td>
</tr>
<tr>
<td>60r</td>
<td>73.8</td>
<td>84.4</td>
<td>91.5</td>
<td>89.2</td>
<td>85.6</td>
<td>85.0</td>
<td>84.8</td>
<td>91.2</td>
<td>97.3</td>
<td>99.8</td>
<td>-</td>
<td>99.7</td>
<td>93.6</td>
<td>89.6</td>
</tr>
<tr>
<td>75r</td>
<td>74.7</td>
<td>85.9</td>
<td>85.2</td>
<td>80.6</td>
<td>77.4</td>
<td>73.5</td>
<td>73.5</td>
<td>80.8</td>
<td>87.4</td>
<td>94.0</td>
<td>99.4</td>
<td>-</td>
<td>98.6</td>
<td>84.2</td>
</tr>
<tr>
<td>90r</td>
<td>79.6</td>
<td>76.3</td>
<td>73.3</td>
<td>71.0</td>
<td>69.2</td>
<td>65.8</td>
<td>64.5</td>
<td>66.8</td>
<td>74.6</td>
<td>79.3</td>
<td>90.5</td>
<td>98.4</td>
<td>-</td>
<td>75.7</td>
</tr>
<tr>
<td>Avg.</td>
<td>77.9</td>
<td>85.6</td>
<td>89.1</td>
<td>90.3</td>
<td>89.6</td>
<td>87.8</td>
<td>85.4</td>
<td>87.4</td>
<td>89.7</td>
<td>90.4</td>
<td>88.9</td>
<td>85.6</td>
<td>78.4</td>
<td>86.6</td>
</tr>
</tbody>
</table>
Cross-pose Recognition Results

- 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.

- 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.
Cross-pose Recognition Results

- Pose pairs whose angles are close to each other are likely to produce higher recognition rates.
- Close recognition rates of for symmetric pose pairs
- Obvious decrease in the recognition rates when input and output poses have opposite signed angles and at least one pose has a large angle ($|\phi| \geq 60$)
- Prevent the decrease in results of opposite signed large pose angles by using the symmetric property of eyes.
### Cross-pose Recognition Results

#### Table: Results for all intensity levels by using Gabor features

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

Recognition rates are higher for higher intensity levels.
Results for Unknown Subjects

In that case, gallery and probe are composed of expressions of different subjects, that is gallery and probe are from different sets.

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

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

- An approach to recognize an individual’s expressions across different pose angles
  - Keep expressiveness constant
  - Model the changes caused by the pose
- Alignment method that works for all pose angles
  - To match corresponding areas
  - All faces have the same eye row, mouth row, and eye distance
- Local blocks of intensity and Gabor features around the visible eyes and mouth to represent faces
  - Size and location of the extracted blocks affect the performance
  - Parameters might differ for different feature representations
  - A small block size might miss some information related to the expression
  - Larger block sizes might cover some of the background
  - Mouth offset parameter is essential for a good performance, especially for large pose angles
  - Optimal parameters in the Gabor representation are not the same with the default parameters
Conclusion

- 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.
- Gabor features outperform intensity features significantly for other pose pairs.
- Even a small number of bases might show comparable performance with large number of bases. Only difference between the face images that we try to match is the pose angle.
- Results are comparable to the baseline results.
- Cross-pose recognition might be a good alternative for pose-specific systems in multi-view facial expression recognition.
- PLS method is also good at matching expressions of different poses with lower intensity levels.
Conclusion

- Results are significantly improved by the utilization of the subject information.
- PLS bases are specifically trained to model a subject’s expressions across poses.
- PLS method is very good at generalizing a (any) subject’s expressions over different pose angles.
Future Work

- Improvement of the results for unknown subjects to handle the cases
  - expressions of the test subject from the input pose are not available
  - subject information is not available
- Using state-of-the-art multi-view facial recognition algorithms
  - to extract subject information automatically
  - and then classify its expression
- Experimenting with a more challenging dataset
P. Ekman and W.V. Friesen.  
The facial action coding system: A technique for the measurement of facial movement.  

P. Ekman, W.V. Friesen, and J.C. Hager.  
The facial action coding system.  

P. Geladi and B.R. Kowalski.  
Partial least-squares regression: a tutorial.  

Nikolas Hesse, Tobias Gehrig, Gao Hua, and Hazm Kemal Ekenel.  
In 21st International Conference on Pattern Recognition (ICPR 2012).

Yuxiao Hu, Zhihong Zeng, Lijun Yin, Xiaozhou Wei, Jilin Tu, and T.S. Huang.  
A study of non-frontal-view facial expressions recognition.  

Yuxiao Hu, Zhihong Zeng, Lijun Yin, Xiaozhou Wei, and Xi Zhou.
Multi-view facial expression recognition.

D. W. Jacobs, A. Kumar, H. Daume, and A. Sharma.
Generalized multiview analysis: A discriminative latent space.

Annan Li, Shiguang Shan, Xilin Chen, and Wen Gao.
Cross-pose face recognition based on partial least squares.

Annan Li, Shiguang Shan, and Wen Gao.
Coupled bias-variance tradeoff for cross-pose face recognition.

Chengjun Liu and H. Wechsler.
Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition.

Fischer Mika and Ekenel Hazm Kemal.
Analysis of partial least squares for pose-invariant face recognition.
S. Moore and R. Bowden.


Roman Rosipal and Nicole Krmer.

Ognjen Rudovic, Ioannis Patras, and Maja Pantic.
A. Sharma and D. W. Jacobs.
Bypassing synthesis: Pls for face recognition with pose, low-resolution and sketch.

H. Soyel and H. Demirel.
Improved sift matching for pose robust facial expression recognition.

Jacob A. Wegelin and The Thanks.
A survey of partial least squares (pls) methods, with emphasis on the two-block case.

H. Wold.
Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and Matthew J. Rosato.
A 3D Facial Expression Database For Facial Behavior Research.

Emotion recognition from arbitrary view facial images.

A novel approach to expression recognition from non-frontal face images.