Active Shape Model and Linear Predictors for Face Association Refinement

David Hurych and Tomáš Svoboda
Czech Technical University in Prague,
FEE, Dept. of Cybernetics, CMP,
Karlovo náměstí 13,
121 35 Praha 2, Czech Republic
|hurycd1, svoboda|@cmp.felk.cvut.cz

Jana Trojanová
Honeywell Prague Laboratory,
V Parku 2326/18
148 00 Praha 4, Czech Republic
Jana.Trojanova@honeywell.com

Yadhunandan US
Honeywell Tech Solutions Lab
151/1, Doraisanipalya,
560 076, Bangalore 10, IND
Yadhunandan.US@honeywell.com

Abstract

This paper summarizes results of face association experiments on real low resolution data from airport and the Labeled faces in the Wild (LFW) database. The objective of experiments is to evaluate different face alignment methods and their contribution to face association as such. The first alignment method used is Sequential Learnable Linear Predictor (SLLiP), originally developed for object tracking. The second method is well known face alignment method Active Shape Model (ASM). Both methods are compared versus face association without alignment. In case of high resolution LFW database the ASM rapidly increases the association results, on the other hand for real low resolution airport data the SLLiP method brought more improvement than ASM.

1. Introduction

Video surveillance systems are becoming part of everyday life. The number of surveillance cameras used in transportation (airports, railway stations, etc.) or public places (e.g. downtown) is growing exponentially. Due to a large amount of data streams the surveillance operators in monitoring room are unable to aggregate the information (suspicious behavior of some person) across a network of cameras, especially in crowded environments. Face association can help in cases when tracking a person (across large facility) either in on-line scenario or off-line (forensics) analysis where large amount of data is processed.

Face association in real scenarios is a challenging problem due several aspects. The main are low resolution, variable illumination, various facial expression, different pose, rough face location given by detector, etc. The datasets used for testing contain all mentioned aspects except the pose variation (experiments are concentrated on near frontal faces). The LFW [4] dataset contains images with face resolution around $90 \times 90$ pixels. Airport data have face resolution around $24 \times 24$ pixels.

This paper reports results from experiments with two different face alignment methods with respect to their contribution to two face association methods. The results of association for alignment methods are compared versus results without aligned faces (output of face detection).

1.1. Face Association

The goal of the face association is to find a similar face across different video tracks obtained from a network of non-overlapping cameras. The output of the face detection and tracking is used for this purpose. All face images of the same person are collected from the track of that person, and a model is built using this collection. When a face is found in another track, the similarity is computed between detected face and models which are already constructed.

The models for face association are mainly built using the subspace methods, two such methods Null-space LDA Mixture Models [3] and Kernel Direct Discriminant Analysis (KDDA) [5] are used for experiments in this paper.

1.2. Face Alignment

Face association methods (especially subspace methods) require having facial features like eyes, nose, and mouth at
the same position for all images. When testing the performance of face association the ground truth information (location of eyes, nose, and mouth) is used to manually align the faces. In real application the facial features are obtained by face alignment method, here contributions of two methods SLLiP [2] and ASM [6] for face association are compared. The goal of ASM is to match a 2D face shape with a facial image. The location of facial points corresponds with shape points. The goal of SLLiP is to find the transformation parameters for affine transformation of the face to upright position so the facial features would be in the same position in all images.

1.3. Paper Overview

This paper is organized as follows. Next section describes in detail used methodology. Section 3 provides a summary of conducted experiments and results for two face alignment methods. Section 4 concludes and discusses the overall results.

2. Methods Overview

Methodology of face alignment and association is presented in this section. For the face association the Kernel Direct Discriminant Analysis (KDDA) [5] and Null Space Linear Discriminative Analysis (LDA) [3] were used. Brief description is given in subsection 2.1 and 2.2. Sections 2.3 and 2.4 describe sequential learnable linear predictor (SLLiP) and active shape model (ASM) both used for face alignment. Section 2.5 describes the overall process of face association.

2.1. Kernel Direct Discriminant Analysis

Most of the algorithms based on LDA, PCA are naturally linear. Using the linear classifier on a data which is complex generally may not lead to good recognition results. One of the solutions is to use non-linear Kernel based methods. The Kernel Direct Discriminant Analysis (KDDA) method uses the Kernel machines to implement the D-LDA [5] algorithm.

The main idea of KDDA is the projection of data to a higher dimensional space (where the data are linearly separable) using the non-linear mapping function. The non-linear mapping function is learnt from the training data as follows (since the method is rather complex, most of the equations are omitted, reader interested in details is advised to read the paper [5]):

Let’s have a set of labeled training samples—aligned faces \( \{x_s^i\} \) were \( s \in S, S \) is set of \( M \) subjects, \( i = 1 \ldots N \), \( N \) is number of training images for subject \( s \).

1. Put all the images in matrix \( A = [x_1^1, x_2^1, \ldots, x_M^N]^T \). Compute the kernel function \( K \) for matrix \( A \), in our case we use Mahalanobis Kernel [8].

2. Compute the eigenvectors \( U \) for between class scatter matrix (the equation is given in [5] eq.7). Remove the vectors with zero eigenvalues. It means that the training images are projected to non-null space (transformation \( U \) does not contain vectors for zero eigenvalues), where the distance between the classes is increased.

3. Using the output (non-null space projection) of the previous step compute the eigenvectors \( P \) and eigenvalues \( \Lambda \) for within class scatter matrix (the equation is given in [5] eq.8,9). Remove the eigenvectors with high eigenvalues. In other words project the training images to a null space LDA, where the within class distance is minimal.

4. The final KDDA projection is given as \( \Gamma = U P \Lambda^{-1/2} \), where \( U \) represents eigenvectors computed at step 2, \( P \) and \( \Lambda \) represents eigenvectors (eigenvalues) computed at step 3.

5. Last step is computation of class means in the high-dimensional space. The class centers are computed as \( m^s = \Gamma \Phi(z) \), where \( \Phi(z) \) represents the mapping of original class mean through kernel (the equation is given in [5] eqs.11-14).

The output of the algorithm is the transformation matrix \( \Gamma \), kernel \( K \) and mean class \( m^s \). When the complex distributed data are mapped non-linearly onto the higher dimensional space, the data distribution becomes linear. The distance between the class and testing images can be measured by a standard metric like Euclidean distance. When the testing image is given, the KDDA algorithm proceeds as follows:

1. Project the testing image on high-dimensional space as \( x_j = \Gamma \Phi(z) \).

2. Compute the distance from class means \( m^s \) using the Euclidean distance, the smallest distance is the associated class, i.e. the recognized individual.

2.2. Null-space LDA Mixture Models

The major challenge in face association is the intrapersonal difference such as pose, illumination and expression changes. To properly distinguish between the subjects removing the intrapersonal dependency is a highly recommended strategy. The Null Space Linear Discriminant Analysis (LDA) learns the intrapersonal variations and subtracts this variation prior to face association.

The training is divided in two main parts. First the intrapersonal subspace is learned across training images using the Distance From Feature Space (DFFS) method [3]. Then the intrapersonal dependency is removed from data.
and Linear discriminative analysis is used to define the difference among the subjects. More detailed description of DFFS method is as follows:

Let’s have a set of labeled training samples—aligned faces $(x^i_s)$ were $s \in S$, $S$ is set of $M$ subjects, $i = 1 \ldots N$, $N$ is number of training images for subject $s$. The algorithm starts with computation of the mean value for each subject (class mean $m^s$) and subtracting the relevant class mean from the training images $\Delta^i_s = x^i_s - m^s$. The difference $\Delta_i$ is called intrapersonal difference sample. The information about the class assignment of training images $\Delta_i$ is not relevant because $\Delta_i$ truly represent the class independent variations. The goal of the DFFS method is to find a transformation that would subtract from each sample (face) the disturbing intrapersonal variations. Now we have training samples $\Delta_i$. The k-means algorithm prepares an initial cluster assignment for each type of intrapersonal variation. The number of clusters $k$ is chosen by user, in our case $k = 2$. Four steps DFFS iterative algorithm proceeds as follows:

1. Given training samples $\Delta_i$ with their cluster assignment compute the eigenvectors $U^k$, eigenvalues $\Lambda^k$ and cluster centers $\alpha^k$ for each cluster. The cluster $\alpha^k$ represents the intrapersonal variations.

2. Compute the Distance From Feature Space $DFFS = \| (I - U^kU^k^T)(\Delta_i - \alpha^k) \|^2$.

3. Assign $\Delta_i$ to cluster with minimal $DFFS$.

4. Stop if none of $\Delta_i$ has changed the cluster, otherwise return to step 1.

The outputs of the algorithm are the transformation matrices $U^k$ and cluster centers $\alpha^k$.

The Null LDA takes the cluster information and trains the projection that maximizes the distance among the classes while minimizing the distance inside the classes. Three steps Null LDA algorithm proceeds as follows:

1. Use intrapersonal clusters and project the class centers $m^s$ to null space $\beta^k_s = (I - U^kU^k^T)(m^s - \alpha^k)$. This actually means that a class mean is represented by $k$ representatives, not only one mean $m^s$.

2. Apply PCA on $\beta^k_s$ and compute eigenvectors $\Psi^k$.

3. Compute the null space LDA transformation for each intrapersonal cluster $u^k = (I - U^kU^k^T)\Psi^k$.

The performance of the method increases when large sample set for training is provided (i.e. training images contain various lighting and expressions).

The final face association runs in the following steps:

1. Subtract all class centers from testing image $\Delta_j = x_j - m^s$.

2. Compute the distance from feature space for all intrapersonal clusters $\|w^k_\Delta_j\|^2$, and choose the cluster with minimal distance.

3. For the chosen cluster $k$ compute the distances for all classes $\|w^k_\Delta_j\|$, the smallest distance is the associated class, i.e. the recognized individual.

### 2.3. Sequential Learnable Linear Predictor

Human face in image is detected by a face detector. Initial position of face center is quite precise, but the scale is estimated only approximately and face rotation is not estimated at all. Additional image rectification is necessary for more reliable face association. We decided to train the sequential learnable linear predictor (SLLiP) for estimation of 3 parameters: angle of in-plane rotation $\alpha$, scale change $\Delta s_x$ in $x$-dimension and scale change $\Delta s_y$ in $y$-dimension. An image point $(x, y)^T$ is transformed (aligned) to image point $(x', y')^T$ using transformation $A$ accordingly

$$
\begin{bmatrix}
x' \\
y'
\end{bmatrix} = A \begin{bmatrix}
x \\
y
\end{bmatrix},
$$

where

$$
A = \begin{bmatrix}
\cos \alpha & -\sin \alpha \\
\sin \alpha & \cos \alpha
\end{bmatrix} \begin{bmatrix}
1 + \Delta s_x & 0 \\
0 & 1 + \Delta s_y
\end{bmatrix}.
$$

A single learnable linear predictor (LLiP) estimates the transformation parameters directly from image intensities. The parameters are estimated as

$$
t = Hl,
$$

where $l$ is vector of intensity values and $t = [\alpha, \Delta s_x, \Delta s_y]^T$ is vector of parameters and $H$ is a linear regression matrix, which needs to be estimated from training data. Intensity values from image $I$ are collected in vector $l = I(X)$, where $X$ is a subset of image 2D coordinates spread over the face patch, which we call support set, see Fig. 1. In the training phase we need to compute the linear regression matrix, which forms the mapping between the space of image intensities and space of parameters. Matrix $H$ is trained according to equation

$$
H = TL^T (LL^T)^{-1} = TL^+,
$$

where matrix $L = [l_1, l_2, \ldots, l_m]$, contains intensity values of $m$ training examples and matrix $T$ contains columnwise stored ground truth parameters vectors (details about
the training set creation can be found in 3.2). The pair $(L, T)$ forms a training set for a single LLIp and the model $\theta$ is formed by the regression matrix and support set $\theta = \{H, X\}$. As it is shown in [1] the higher is the predictor complexity (number of pixels in support set $X$), the better is the prediction. Nevertheless increasing the complexity leads to diminishing returns. It is better [1] to train a sequence of LLIps in order to overcome this limitation. Predictors in sequence estimate the parameters one after each other (see equation 4), thus each improving the result of previous predictor estimation and lowering the error of estimation.

\[
t_1 = H_1 I(X_1) \\
t_2 = H_2 I(t_1 \circ X_2) \\
t_3 = H_3 I(t_2 \circ X_3) \\
\vdots \\
t = \bigcirc_{i=1,...,k} t_i,
\]

where operation $\circ$ means inverse transformation of support set points $A^{-1}X_i$ using parameters $t_{i-1}$, i.e. aligning the support set to fit the face using parameters estimated by the previous LLIp in sequence. Final result of the prediction is vector $t$ which combines results of all predictions in the sequence. Each predictor in the sequence has its own complexity and training set. The model $\theta_s$ for SLLiP is formed by the sequence of LLIps $\theta_s = \{[H_1, X_1], [H_2, X_2], \ldots, [H_k, X_k]\}$ for $k$ LLIps in sequence. More details about learning the SLLiP can be found in [2].

### 2.4. Active Shape Model

ASM [6] is composed from point distribution model (PDM) defining the shape with its variation and local appearance model describing the point’s surrounding. Fig. 1 shows the points of the PDM, the green line in the figure illustrates a whisker. Pixels along the whisker represents the local appearance model. The whisker should be orthogonal to the edge of shape. The angle of whisker is determined from the position of previous and next point (i.e. the whisker is orthogonal to the line between the previous and next point). The algorithm for training of PDM and local appearance model is as follows:

1. Align the training shapes. Take first shape as reference and align the remaining training shapes. When aligning a pair of shapes minimizes following criteria (Mean Square Error between all points)

\[
d = (X^1 - M(s, \Theta)X^m - t)^T \Phi (X^1 - M(s, \Theta)X^m) - t),
\]

where $\Phi$ is diagonal matrix representing the weights of each point, $M(s, \Theta)$ is transformation defined as

\[
M(s, \Theta) = \begin{bmatrix}
    s \cos \Theta & -s \sin \Theta & x^m \\
    s \sin \Theta & s \cos \Theta & y^m
\end{bmatrix}
\]

2. Apply PCA to aligned training shapes (get eigenvectors $u_k$ and eigenvalues $\lambda_k$). Compute the mean shape $\bar{X}$. The PDM model is represented as

\[
X = \bar{X} + \Phi b
\]

where $\Phi$ is transformation matrix of first $n$ eigenvectors, $b$ is vector deformation parameters. It is suitable to limit the vector $b$ to allow only deformation that were seen in the training set. $b \in (-3\sqrt{\lambda_k}, +3\sqrt{\lambda_k})$

3. Compute the local appearance model for each point separately. The intensities along the whisker $A^m_i$ for all training images are extracted and local appearance model of point is represented as difference between intensities normalized by sum of intensities difference

When new image is given, the algorithm of model fitting works as follows

1. Set all values of vector $b$ to zero, define the initial scale based on rectangle width from face detection. Set the model $X_{i-1} = \bar{X}$. Repeat step 2 to 6 until the convergence (defined in step 3) or maximal number of iteration.

2. Get the initial shape (for current iteration) as $X_i = X_{i-1}M(s, \Theta) + \Phi b$
3. Search new position of each PDM point. Compare the local appearance model with intensities around the actual whisker. If no point change the position than the model has converged.

4. Find transformation \( M(s, \Theta) \) needed to align the initial shape with new shape from previous point. Use the criteria as in training step 1.

5. Get actual shape \( \mathbf{X}_0 = \mathbf{X}_{i-1}M(s, \Theta) \) applying the transformation from previous points to initial shape.

6. Transform the actual shape to PCA subspace \( b = P/\mathbf{X} \).

7. Limit the actual value of \( b \) to maximal allowed deformation. Go to step 1.

### 2.5. Face Association Process

The face association process associates detected faces across images (cameras). Face detection and tracking is applied on given video segment (in case of LFW only face detection is used). The detector is based on Viola and Jones [7], it returns rectangle around face and information about the pose (frontal, left profile, right profile). The tracker is based on multiple independent appearance models and joint probabilistic data association filter is used to associate faces and to select the best hypothesis for image sequence. In order to establish the full track of the face, the tracker traces back the face starting from the frame where it was detected and going backwards in time up to the point where it can be reliably tracked.

For experiments (training and testing of face association) three types of data have been generated. First one is output of face detection and tracking, the face patches are cropped from images. Second one is output of SLLiP, the predicted transformation is used for face alignment and estimated scale changes are used for image cropping. The third one is output of ASM, the eye coordinates are used for face alignment and distance between eyes is used for image cropping. All extracted faces from face detection and two alignment methods are normalized to have same size. The result of association for face detection is used as reference. The result from the reference experiment is compared to results of experiments with association after the face alignment methods.

### 3. Experiments

This section provides a description of experiments: performance evaluation (3.1), settings of used algorithms (3.2, 3.3 and 3.4), database specification with results (3.5 and 3.6), and discussion on the results (3.7). All the experiments are focused on near frontal faces with various illumination and facial expressions (see Figs. 1, 2).

---

**Figure 2.** Sample from airport dataset. Snapshots are from a person track in corridor. Face resolution is: left 24 × 24, middle 30 × 30, and right 40 × 40

### 3.1. Performance Evaluation

For all experiments two metrics are used: recognition rate and ROC curve. All tables in this document represent recognition rate (REC). The REC value can be given as absolute or relative value. The absolute value of REC expresses the ratio of valid match versus number of testing images for given subject. For both methods the match is valid when the testing image has the lowest distance to its mean class (defined during training) for details about distance computation see subsection 2.1 and 2.2). The relative REC value expresses the performance as percentage relative to a reference alignment method. The reference method contains in all cells zero entries; typically its the first column in table. The relative value is defined as follows:

\[
REC_R = 100 \frac{REC_{\text{method}} - REC_{\text{referential}}}{REC_{\text{referential}}}
\]

where, \( REC_{\text{referential}} \) is recognition rate for face detection, \( REC_{\text{method}} \) is recognition rate for used face alignment method. The higher the \( REC_R \) value is, the better is the result (if the \( REC_R \) value is negative then the method is worse than reference method). Relative REC is another way how to represent the results, so it is easy to read the improvement or deterioration of methods.

The ROC curve illustrates the recognition rate versus false alarm rate. Each point on ROC curve is derived from similarity matrix and actual value of threshold. Similarity matrix expresses the distance between the testing images and training class means. Thresholds values represent chosen values of distances from similarity matrix. For ROC curve 100 threshold values were used (the smallest and biggest value of distance in similarity matrix divided by 100 steps). The performance represented in ROC curve is useful for cases when sequence of images is available. To classify the testing image the threshold is used for the distance to the class center (mean \( m^* \)) instead of taking the minimal distance to some class center. In some cases the image can be assigned to more than one class. Anyway when sequence of images is available such an information (image assigned
to more than one class) can be used for a voting system (e.g. having a sequence of 20 images, if more votes are given to class one the whole sequence is recognized as class one).

For each experiment setting (association method and alignment method), the experiment was repeated 15 times (using random selection of training data). The results in tables 1, 2 and figures 3, 4 represent mean value from 15 experiments.

3.2. SLLiP Setting

For the task of human face alignment we need to build a general predictor capable of working with any human face. Therefore we need a large training set in order to cover as many human face appearances as possible. 1 155 images from BioID dataset [9] were used for training. In the training phase we made random perturbations of the parameters \((\alpha, \Delta s_x, \Delta s_y)\) each on some predefined range (e.g. \(\pm \frac{\pi}{10}\) for rotation, etc.). Training set is made by transforming the images using these perturbed parameters. From 1 155 training images we got a total of 173 250 training examples. We tested several different models \(\theta_s\) and we selected the one with best performance. For SLLiP training we used at first branch and bound algorithm [2] which finds the optimal complexities and number of predictors in sequence for some fixed maximal error achieved on the training data. To train the rest of tested models we used more heuristic approach of training which is described in the rest of this section.

Our heuristics is following. The complexities of predictors were set manually and are growing with their order in sequence. With lower complexity we obtain lower precision of estimation, but better robustness. On the other hand with high complexity we obtain more precise estimation, with lower robustness. The idea is to start with more robust LLLiP which is able to give first rough parameters estimation and gradually improve the precision of estimation with more complex LLLiPs in the remaining sequence. The last predictor is less robust than all previous predictors, but this is not a problem, because the support set is already close to the right alignment and we expect only small but precise parameters change estimation from the last predictor. Models trained using this heuristics outperformed (in final association results) models trained using branch and bound algorithm.

To improve the robustness of predictors with lower complexity (and avoid the noise) we used approach similar to building image lower resolution pyramid. The predictors do not collect the image intensities directly but average intensity value of few neighboring pixels around each point of support set is computed and stored in vector \(l\). Averaging window size is bigger for LLLiPs with low complexity (e.g. \(7 \times 7\) pixels) and smaller for LLLiPs with high complexity (\(1 \times 1\) pixels for the last LLLiP in sequence). The averaging window sizes are on-line adapted according to the initial face patch size obtained by face detector.

3.3. ASM Setting

The ASM model used for face alignment is composed from 68 points, the training of model was done on Bio-ID dataset [9]. During training random noise was added to the training data. The model was built for near-frontal faces with small in-plane rotations.

3.4. Settings of Association Algorithms

The number of training images for both algorithms was on LFW database was 18, on airport data 25. The rest of images was used for testing. The images were resized for face association to \(64 \times 64\) pixels.

The number of clusters in the intrapersonal subspace is kept as 2 in our experiments in the Null-space LDA Mixture Models. For the KDDA there are several variables to be set, the number of features was set to 64, the scale value \(\sigma^2\) was set to 8, 6, the size of block is set to \(8 \times 8\) pixels.

3.5. The LFW Data

The Labeled Faces in the Wild is freely available database consisting of more than 13 000 images of 5 749 individuals. The number of images per subject varies between 1 to 530 samples. Most of data (4 069 subjects) has only one image per individual. Database is available at \(^2\), note that images have unconstrained environment. For the experiment at least 20 samples with frontal face are required. Only 24 subjects fulfilled the requirements. Four different results were obtained for LFW database. First one is association without face alignment, the second one is association with SLLiP alignment, the third one is association with ASM alignment, and the fourth is association with manually aligned face images.

<table>
<thead>
<tr>
<th>Method</th>
<th>FD</th>
<th>SLLiP</th>
<th>ASM</th>
<th>GT</th>
</tr>
</thead>
<tbody>
<tr>
<td>rel</td>
<td>KDDA</td>
<td>54.7</td>
<td>55</td>
<td>72.5</td>
</tr>
<tr>
<td>rel</td>
<td>NULL-LDA</td>
<td>50.8</td>
<td>51</td>
<td>69.7</td>
</tr>
<tr>
<td>abs</td>
<td>KDDA</td>
<td>0</td>
<td>0.4</td>
<td>32.3</td>
</tr>
<tr>
<td>abs</td>
<td>NULL-LDA</td>
<td>0</td>
<td>0.6</td>
<td>37</td>
</tr>
</tbody>
</table>

Table 1. Absolute and relative recognition rate [%] on LFW database. The columns represent results for different alignment methods, lines are the face association method used for experiment. FD - face detection, SLLiP linear predictor, ASM - shape model, GT - ground truth of eyes position. For the star point the recognition rate is 90% and the FAR is 20%.

Tab. 1 represents the absolute and relative recognition rate. Generally the results for KDDA are better than results for Null-Space LDA. The best results were obtained for KDDA association and alignment using the ground truth

\(^2\)http://vis-www.cs.umass.edu/lfw/
Figure 3. ROC curve for LFW dataset [4]. The reference curve is red (no alignment), results for alignment methods are green-SLLiP, blue - ASM, black - GT (ground truth). Solid line is for KDDA, and dashed line is for Null LDA association method. For the star point the recognition rate is 90% and the FAR is 20%.

(GT) information (manually labeled eyes position). The ASM alignment is slightly worse than GT (this signifies that eye location found by ASM is close to ground truth information). The results obtained for SLLiP are close to results from face detection, thus we can say that SLLiP does not have significant effect on high resolution data.

Fig. 3 visually compares the results for all experiments with LFW dataset. It is desirable to have the curve as close as possible to upper left corner (point 0,1). The point depicted in Fig. 3 as star (on the line GT-KDDA) illustrates the recognition rate for one chosen threshold value. That means that instead of using the minimal distance to class mean the distances to all class means are compared versus defined threshold (testing image can be assigned to more than one class). For one testing image such a number has no relevance, but having sequence of images the recognition rate can be significantly improved. Using a voting system with majority rule (i.e. compute the number of votes given to each class and assigning the testing sequence to the class with highest number of votes) may lead to much higher recognition rate than the one presented in Tab. 1. For instance in LFW database taking all testing images of one subject as sequence would yield to recognition rate of 100%.

3.6. The Airport Data

Data from an airport were acquired from six non-overlapping cameras. Three subjects are presented across all cameras. These subjects were playing various activities (meeting, following, luggage switch, etc.). Fig. 2. shows example of data. Prior to face association, face detection and tracking was applied. In total there are 22 person tracks used for experiment. Three subjects have a corresponding person track in all cameras. Four subjects have a person track in one of the cameras. Each image in track is accompanied with pose information (frontal, profile). The model is built over the whole track, except the non-frontal faces. The face size across the track in corridor vary from $10 \times 10$ pixels till $50 \times 50$. The face size in the hall is around $15 \times 15$ pixels. For the airport data the ground truth information (i.e. eye coordinates) is not available thus only results for alignment methods versus face detection are presented.

![Figure 4. ROC curve for Airport dataset [4]. The reference curve is red (no alignment), results for alignment methods are green-SLLiP, blue - ASM. Solid line is for KDDA, and dashed line is for Null LDA association method. For the star point the recognition rate is 60% and the FAR is 20%.](image)

<table>
<thead>
<tr>
<th></th>
<th>KDDA</th>
<th>NULL-LDA</th>
<th>ASM</th>
</tr>
</thead>
<tbody>
<tr>
<td>abs</td>
<td>50.4</td>
<td>52.3</td>
<td>42.7</td>
</tr>
<tr>
<td>rel</td>
<td>50.9</td>
<td>52</td>
<td>36.7</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>3.8</td>
<td>-15</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>2.1</td>
<td>-28</td>
</tr>
</tbody>
</table>

Table 2. Absolute and relative recognition rate [%] on airport database. The columns represent results for different alignment methods, lines are the face association method used for experiment. FD - face detection, SLLiP linear predictor, ASM - shape model.

Table 2 shows that the best performance was obtained in combination of KDDA face association and SLLiP face alignment methods. On the other hand the results obtained for ASM alignment deteriorated the performance of face association. The reason of bad performance was lower quality of the images, especially the indistinctive edges of faces (see Fig. 2).

Fig. 4 illustrates the results from experiments performed on airport data. Observing the ROC curves on Figs. 4, 3 reveals that for same performance the resulting false alarm is
much higher for airport data (e.g. REC=90% has 4 times bigger FAR). The reason of worse performance is lower quality of data making the face association more difficult. Using a voting system with majority rule for 22 tracks and given threshold (depicted as star) all subjects are recognized correctly. In real scenario this threshold is not known (since we don’t have the ground truth data - assignment of subject to class). To derive threshold, huge amount of referential data (ground truth - subject assignment to class) is needed. Still this threshold will be data dependent (i.e. we can use such a threshold for one type of data e.g. airport data, but we cannot use it for other type of data e.g. LFW).

3.7. Discussion

Essentially, the ASM alignment significantly helps with high resolution LFW data and SLLiP improves association in the low resolution airport data. Both the association methods are sensitive to the quality of alignment, the Null LDA a bit more than the KDDA. The good performance of ASM in aligning hi-res data was perhaps expectable. It finds the eye positions reliably which is the crucial factor in stable cropping face images. The SLLiP that offers a global alignment transformation does not really help here which is a bit disappointing. The alignment is not stable enough and even small variations spoil the learning of the association methods. The ASM completely fails in the main target application—face association of low resolution surveillance data. The improvement of the SLLiP may seem minor the data however are really pure and application of the SLLiP alignment costs virtually nothing, few milliseconds at most. Hence, using the SLLiP alignment for low resolution data is recommendable.

All tested methods were implemented in Matlab and their time efficiency on a standard PC are as following. KDDA associates one face to one of 24 clusters in 70 milliseconds, while DFFS does it in 1.1 seconds. Experiments showed that KDDA generally outperforms DFFS in both association rate and speed. In the alignment step, SLLiP is much faster than ASM it aligns one face in 14 and ASM in 500 milliseconds.

4. Conclusion

We experimented with two methods for face association and two face alignment methods. The experiments were motivated by an airport surveillance application. The low resolution data were accompanied by the standard LFW dataset. The ASM proved its usability in case of high resolution data but completely fails for the airport dataset. In contrary, the sequential linear predictor does not really help in aligning good data but improves the association performance for the airport cameras. Since the SLLiP runs extremely fast its use for low resolution data is recommended. For a good data ASM does much better than SLLiP.

Acknowledgements

The 1st author was supported by Czech Ministry of Education project 1M0567 and by EC project FP6-IST-027113 eTRIMS, the 2nd author was supported by Czech Ministry of Education project 1M0567 and by EC project FP6-IST-027787 DIRAC. Any opinions expressed in this paper do not necessarily reflect the views of the European Community. The Community is not liable for any use that may be made of the information contained herein.

References