Face Recognition – Part II

Rainer Stiefelhagen

November 23, 2009, Karlsruhe, Germany
Outline

- Recent Advances
  - Local Appearance-based Face Recognition
  - Face recognition across pose
    - Face normalization using AAMs
    - Video-based face recognition
  - Face Recognition using 3D Morphable Models
Local Appearance Based Face Recognition

- Modular Eigenspaces
- Local Principal Component Analysis (PCA)
- Local Discrete Cosine Transform (DCT)
Local vs. Holistic Approaches

- Local variations on the facial appearance, i.e. due to different expression, occlusion and lighting, lead to modifications on the entire representation in the holistic approaches, while in local approaches only the corresponding local region is effected.

- Face images contain different statistical illumination – high frequency at the edges, i.e. eyebrows, low frequency at smooth regions, i.e. cheeks. Easier to represent the varying statistics linearly by using local representation.

- Local approaches facilitate the weighting of each local region in terms of their effect on face recognition.
Modular Eigenspaces

- Does classification using fiducial regions instead of using entire face.

Modular Eigenspaces

Local Principal Component Analysis (Modular PCA)

- The face images are divided into N smaller sub-images.
- PCA is applied on each of these sub-images.

- Performed better than global PCA on large variations of illumination and expression
- No improvements under variation of pose

(Gottumukal & Asari, 2003)
Local Appearance-based Face Recognition Using Discrete Cosine Transform (DCT)

- **Objective:**

  To mitigate the effects of expression, illumination and occlusion variations by performing local analysis and by fusing the outputs of extracted local features at the feature or at the decision level.
Discrete Cosine Transform (DCT)

- Easy & Fast Implementation
- No need for (good) alignment in the training stage to construct proper basis -> data independent bases
Discrete Cosine Transformation

- **2D discrete cosine transform (DCT):** Transforms an image block \( f(x,y) \), where \( x, y = 0, 1, ..., N_P - 1 \), to \( N_P \times N_P \) matrix \( C(v,u) \) containing 2D DCT coefficients:

\[
C(v,u) = \alpha(v)\alpha(u) \sum_{y=0}^{N_P-1} \sum_{x=0}^{N_P-1} f(y,x) \beta(y,x,v,u)
\]

for \( v,u = 0,1,2,...,N_P - 1 \), where

\[
\alpha(v) = \begin{cases} \sqrt{\frac{1}{N_P}} & \text{for } v = 0 \\ \sqrt{\frac{2}{N_P}} & \text{for } v = 1,2,...,N_P - 1 \end{cases}
\]

and

\[
\beta(y,x,v,u) = \cos \left( \frac{(2y+1)v\pi}{2N_P} \right) \cos \left( \frac{(2x+1)u\pi}{2N_P} \right)
\]
Discrete Cosine Transform (DCT)

- Compact representation

```
<table>
<thead>
<tr>
<th>415</th>
<th>60</th>
<th>59</th>
<th>26</th>
<th>61</th>
<th>48</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>146</td>
<td>9</td>
<td>69</td>
<td>54</td>
<td>8</td>
<td>5</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>77</td>
<td>70</td>
<td>28</td>
<td>4</td>
<td>13</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>33</td>
<td>11</td>
<td>19</td>
<td>9</td>
<td>8</td>
<td>0</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>14</td>
<td>21</td>
<td>11</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>0</td>
<td>15</td>
<td>8</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
```
Local Appearance Based Face Representation

- Divide the input image to blocks of 8x8 pixels size.
- Perform DCT on each block.
- Order the remaining DCT coefficients using zig-zag scan.
- Remove the first DCT coefficients and from the remaining ones select the first M of them.

Figure: Zig-zag scan pattern
Feature Fusion
Decision Fusion
DCT vs. Principal Component Analysis (PCA)

- In PCA, a data-specific space is built which requires fine alignment to obtain proper bases.
- In DCT, one has data independent bases.

**Figure:** Eigenfaces bases computed from mis-aligned (top) and well-aligned (bottom) images
Local DCT-based Face Reco - Results
(Ekenel & Stiefelhagen, 2005)

<table>
<thead>
<tr>
<th>Method</th>
<th>Reco. Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA (20)</td>
<td>75.6%</td>
</tr>
<tr>
<td>LDA (14)</td>
<td>80.0%</td>
</tr>
<tr>
<td>ICA 1 (40)</td>
<td>77.8%</td>
</tr>
<tr>
<td>ICA 2 (40)</td>
<td>72.2%</td>
</tr>
<tr>
<td>Global DCT (64)</td>
<td>74.4%</td>
</tr>
<tr>
<td>Local DCT (18) + GMM (8) as in [12]</td>
<td>58.9%</td>
</tr>
<tr>
<td>Local DCT + Feature Fusion (192)</td>
<td>86.7%</td>
</tr>
<tr>
<td>Local DCT (10) + Decision Fusion (64)</td>
<td>98.9%</td>
</tr>
</tbody>
</table>

Results on Yale-DB
(15 individuals)

Results on CMUPIE-DB
(68 individuals)
Experimental Results

Face Recognition Grand Challenge (FRGC) database,
Controlled environment, 120 subjects,
Expression variations, time gap

98.5% (98.8%)

FRGC face database,
Uncontrolled environment,
120 subjects,
Expression variations, time gap

96.2% (80.6%)

AR face database,
110 subjects,
Occlusion

98.2% (97.5%) 97.3% (93.5%)

Results in (): best reported performance by other task-specific algorithms
Experimental Results: Illumination

CMU PIE face database, 68 subjects, Illumination variations

Extended Yale face db, 38 subjects, Illumination variations

100% (100%)

98.7% (99.2%)

98.9% (97.6%)
Some Applications

**Surveillance:** 92.5%, 41 subjects

**Smart Environments:**
Best system in the CLEAR evals
(96.4%, 28 subjects)
Some Applications
Local-Appearance based Face Recognition using DCT: Conclusions

- A generic approach that handles expression, illumination, occlusion, and uncontrolled conditions

- Works reliably under real-world conditions
Problem: Matching across face pose

- Problem: Different viewpoint / head orientation

- Recognition results degrade, when images of different head orientation have to be matched
Pose-Normalization

- Alignment using just eye-positions is not sufficient
- Idea:
  - Find several facial features (mesh)
  - Use complete mesh to normalize face
- Here: 2D Active Appearance Models
  - A texture and shape-based parametric model
  - Efficient fitting algorithm: Inverse compositional (IC) algorithm
  - (see specific lecture on January 8)
Model and Fitting

(→ see also specific lecture on January 8)

- Independent Shape and Appearance Model

\[ s = (x_1, y_1, x_2, y_2, \ldots, x_v, y_v)^T = s_0 + \sum_{i=1}^{n} p_i s_i \]

\[ A(x) = A_0(x) + \sum_{i=1}^{m} \lambda_i A_i(x) \quad \forall x \in s_0 \]

- Fitting Goal:

\[ \arg \min_{p, \lambda} \sum_{x \in s_0} \left[ A_0(x) + \sum_{i=1}^{m} \lambda_i A_i(x) - I(W(x; p)) \right]^2 \]
Instances of Shape and Texture

\( s_0 \), \( s_0 + p_1 s_1 \), \( s_0 + p_2 s_2 \), \( s_0 + p_3 s_3 \), \( A_0(x) + \lambda_1 A_1(x) \), \( A_0(x) + \lambda_2 A_2(x) \), \( A_0(x) + \lambda_3 A_3(x) \)
Alignment with AAMs
Fitting Examples

Fitted mesh

Mismatched mesh

Rainer Stiefelhagen
Pose normalization

- Fitted model can be used to warp image to frontal pose
  - E.g. using piecewise affine transformation of mesh triangles

- Faces with different poses from FERET data base and their pose-aligned images
Results (2)

- Much better results under pose variations compared to simple affine transform:

<table>
<thead>
<tr>
<th>Probe set</th>
<th>$bb$</th>
<th>$bc$</th>
<th>$bd$</th>
<th>$be$</th>
<th>$bf$</th>
<th>$bg$</th>
<th>$bh$</th>
<th>$bi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>With pose correction</td>
<td>44.0%</td>
<td>81.5%</td>
<td>93.0%</td>
<td>97.0%</td>
<td>98.5%</td>
<td>91.5%</td>
<td>78.5%</td>
<td>52.5%</td>
</tr>
<tr>
<td>Simple Affine Transf.</td>
<td>0.0%</td>
<td>5.5%</td>
<td>26.0%</td>
<td>62.5%</td>
<td>78.5%</td>
<td>26.5%</td>
<td>4.0%</td>
<td>1.0%</td>
</tr>
</tbody>
</table>

- Different warping functions can be used
  - Piecewise affine transformation worked best

- Approach works well with local-DCT-based approach
  - but not so well with holistic approaches, such as Eigenfaces (PCA)

(Gao, Ekenel, Stiefelhagen, ICB’09)

Rainer Stiefelhagen
Pose-normalization with AAMs

3D pose correction for pose invariant face recognition, DFFS assesses the quality of model fitting
Video-based face recognition

- Temporal information (face tracking) can help us to match faces under different poses
  - And problem with AAMs: need good resolution …

- Investigated in a person-retrieval scenario
  - Goal is to find shots with a specific actor

- Approach
  - Pre-segment shots via shot boundary detection
  - Track all faces in each shot -> “face tracks”
  - Find best matches between “face track” of selected face and all other face “tracklets”
Three different tracks of a person

Faces under different pose can be matched using the temporal association from the different tracks and the associations by face recognition (solid lines)
Actor Retrieval System: Screenshot

Selected Face

Initial (high confidence) query results

Interactive selection of additional query images / tracks

Rainer Stiefelhagen
Face Recognition Based on Fitting a 3D Morphable Model
(Blanz & Vetter, 2003)

- A method for face recognition across variations in pose and illumination.
- Simulates the process of image formation in 3D space.
- Estimates 3D shape and texture of faces from single images by fitting a statistical morphable model of 3D faces to images.
- Faces are represented by model parameters for 3D shape and texture.
A Morphable Model of 3D Faces

- The morphable face model is based on a vector space representation of faces that is constructed such that any combination of shape and texture vectors $S_i$ and $T_i$ describes a realistic human face:

$$S = \sum_{i=1}^{m} a_i S_i \quad \quad T = \sum_{i=1}^{m} b_i T_i$$
Face Recognition Based on Fitting a 3D Morphable Model

- Model-based Recognition

![Diagram showing the process of face recognition based on fitting a 3D morphable model]

- Database of 3D Scans
- Morphable Face Model
- Fitting
- Gallery images
- Probe image
- Identity determination

\[ \alpha_i, \beta_i \]
Database of 3D Laser Scans

- 3D scans of 100 males and 100 females were used to derive the morphable model.
- The scans represent face shape in cylindrical coordinates.
- The device measures radius $r$, and red, green, blue $\{R,G,B\}$ components of surface texture.

$$I(h, \phi) = (r(h, \phi), R(h, \phi), G(h, \phi), B(h, \phi))^T, \quad h, \phi \in \{0,\ldots,511\}.$$  

$h$: vertical steps, $\phi$: angular steps
Preprocessing / Alignment

- Some preprocessing is necessary
  - Filling holes
  - Some trimming along edges
  - Removal of back of the head and shoulders

- 3D Alignment of faces
  - Establish point-to-point correspondence between reference and new face
  - Based on optical flow
Face Vectors

- The definition of shape and texture vectors is based on a reference face $I_0$.
- The location of the vertices of the mesh in Cartesian coordinates is $(x_k, y_k, z_k)$ with colors $(R_k, G_k, B_k)$
- Reference shape and texture vectors are defined by:
  \[
  S_0 = (x_1, y_1, z_1, x_2, \ldots, x_n, y_n, z_n)^T \\
  T_0 = (R_1, G_1, B_1, R_2, \ldots, R_n, G_n, B_n)^T
  \]
- To encode a novel scan $I$, the flow field from $I_0$ to $I$ is computed.
Principal Component Analysis

- PCA is performed on the set of shape and texture vectors separately.

- Eigenvectors form an orthogonal basis:

\[ S = \overline{s} + \sum_{i=1}^{m-1} \alpha_i \cdot s_i, \quad T = \overline{t} + \sum_{i=1}^{m-1} \beta_i \cdot t_i \]
Fig. 4. The average and the first two principal components of a data set of 200 3D face scans, visualized by adding $\pm 3\sigma_{S_i}s_i$ and $\pm 3\sigma_{T_i}t_i$ to the average face.
Model-based Image Analysis

- The goal of the fitting process is to find shape and texture coefficients describing a 3D face model such that rendering produces an image $I_{\text{model}}$ that is as similar as possible to $I_{\text{input}}$.

- For initialization 7 facial feature points, such as the corners of the eyes or tip of the nose should be labelled manually.
Model Fitting

- Goal: Minimize
  \[ E_I = \sum_{x,y} \left\| I_{\text{input}}(x,y) - I_{\text{model}}(x,y) \right\|^2. \]

- Shape, texture, transformation, and illumination are optimized for the entire face and refined for each segment.
  - 99 coefficients \( \alpha_i, \beta_i \)
  - 22 rendering parameters (pose angles, translation, focal length, light intensities, illumination direction, …)

- Complex iterative optimization procedure
  - Processing time: 4.5 minutes per image (2 GHz, P4)
3D Face Reconstruction from a Single Image
Segments

- From a given set of examples, a larger variety of different faces can be generated if linear combinations of shape and texture are formed separately for different segments of the face:

*Eyes, nose, mouth and the surrounding area.*
Fitting Results

(a) Original Image

(b) Fitted Model

Novel views
Model-based Recognition

![Diagram of model-based recognition process]

- Database of 3D Scans
  - Morphable Face Model
  - Gallery
    - Fitting $\alpha_i, \beta_i$
    - Fitting $\alpha_i, \beta_i$
    - Fitting $\alpha_i, \beta_i$
  - Probe
    - Fitting $\alpha_i, \beta_i$
    - Identity
Results

- FERET: used 194 individuals, 10 poses, different illuminations conditions (only in one pose)
- CMU-PIE: used 68 individuals, three view-points (front, side, profile), 22 illuminations

<table>
<thead>
<tr>
<th>Database</th>
<th>$d_M$</th>
<th>$d_A$</th>
<th>$d_W$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMU-PIE</td>
<td>87.2%</td>
<td>94.2%</td>
<td>95.0%</td>
</tr>
<tr>
<td>FERET</td>
<td>80.3%</td>
<td>92.2%</td>
<td>95.9%</td>
</tr>
</tbody>
</table>

Results for different distance measures
Videos – Building a Morphable Model

Building a Morphable Model
Videos – Application to Images

Application to Images