International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 23, No. 3 (2009) 433–461
© World Scientific Publishing Company
World Scientific
www.worldscientific.com

FACE DETECTION AND RECOGNITION USING MAXIMUM LIKELIHOOD CLASSIFIERS ON GABOR GRAPHS

MANUEL GÜNTHER and ROLF P. WÜRTZ

Institut für Neuroinformatik
Ruhr-Universität Bochum
D–44780 Bochum, Germany
manuel.guenther@neuroinformatik.ruhr-uni-bochum.de

We present an integrated face recognition system that combines a Maximum Likelihood (ML) estimator with Gabor graphs for face detection under varying scale and in-plane rotation and matching as well as a Bayesian intrapersonal/extrapersonal classifier (BIC) on graph similarities for face recognition. We have tested a variety of similarity functions and achieved verification rates (at FAR 0.1%) of 90.5% on expression-variation and 95.8% on size-varying frontal images within the CAS-PEAL database. Performing Experiment 1 of FRGC ver2.0, the method achieved a verification rate of 72%.

Keywords: Face recognition; Maximum Likelihood estimators; Gabor graphs.

1. Introduction

To be useful for practical applications face recognition systems must solve a variety of tasks. *Detection* procedures must select single faces from images, excluding the background. *Matching* or *alignment* estimates pairs of corresponding points in the face image to be analyzed (subsequently called *probe image* for brevity) and stored faces (called *models*), *comparison* calculates a similarity between probe image and models, and a *statistical decision* about the identity is made on the basis of similarities. Finally, for efficiency reasons the *memory organization* of the models is of high relevance.

Elastic Bunch Graph Matching (EBGM)[15] is one of the successful base techniques for face recognition. It provides an integrated method for detection, matching, comparison and decision. One of its strengths is the memory organization in bunch graphs, which integrates the results of off-line matching between all models for efficient alignment to the probe image and rapid comparison. Preprocessing with Gabor wavelets yields basic invariance under small deformations and also illumination changes.

On the other hand, there are statistical learning approaches, which exploit the statistics of many models to come to an optimal decision about identity. They go back to the work by Moghaddam and Pentland,[7,8] who applied them to so-called *eigenfaces*, i.e. the principal components of facial images.

In this article, we present a method that combines the advantages of Gabor-labeled graphs with statistical decision making into an integrated method. We also improve the performance by scale and rotation correction of the Gabor features and by the use of different distance measures for feature comparison. In Sec. 2, we will define Maximum Likelihood estimators, and in Sec. 3, the novel method of Gabor graph scaling and rotation is introduced. Sections 4 and 5 will present the face detection and matching steps, while Sec. 6 describes decision making. Finally, Secs. 7 and 8 show the results of either procedure on different databases. Section 9 gives a summary and outlook to future work.

2. Maximum Likelihood Estimators

2.1. *Previous work*

Maximum Likelihood (ML) classifiers for face detection and recognition have been introduced by Moghaddam *et al.*[7,9] They defined ML classifiers on eigenfaces for face detection, and maximum a posteriori (MAP) classifiers in a PCA subspace of image differences for face recognition. For face detection,[7] they transformed image patches $\mathbf{x}$ of different sizes and from different positions in the input image to eigenspace and estimated the Maximum Likelihood probability

$$\hat{P}(\mathbf{x}) = \frac{\exp\left(-\frac{1}{2}\sum_{i=1}^{M}\frac{y_i^2}{\lambda_i}\right)}{(2\pi)^{\frac{M}{2}}\prod_{i=1}^{M}\lambda_i^{\frac{1}{2}}} \cdot \epsilon(\mathbf{x}) \ , \tag{1}$$

to find the patch that best covers the face. In Eqs. (1) and (2), $\mathbf{y}$ names the input $\mathbf{x}$ transformed to eigenspace, while λ_i are the corresponding eigenvalues, M is the number of eigenfaces used for estimation, and ϵ approximates the distance from feature space (DFFS).[7]

For face recognition, Moghaddam *et al.*[9] implemented the Bayesian intrapersonal/extrapersonal classifier (BIC) that classifies differences $\mathbf{x}$ between two facial images into *intrapersonal* if it was created using two images of the same person and *extrapersonal* for images of different persons. They defined the MAP probability of being intrapersonal using Bayes rule on the Maximum Likelihood estimates $P(\mathbf{x}|\Omega)$ of the two classes Ω_I and Ω_E. Equation (1) is applied to approximate the likelihoods $P(\mathbf{x}|\Omega_I)$ and $P(\mathbf{x}|\Omega_E)$, and one PCA subspace is trained for each class. Teixeira[13] simplified the ML probability of Eq. (1) by using its logarithm as a similarity measure and showed that the MAP probability can be transformed to the much simpler form

$$S_T(\mathbf{x}) = -\left[\sum_{i=1}^{M_I}\frac{y_{I;i}^2}{\lambda_{I;i}} + \epsilon_I(\mathbf{x})\right] + \left[\sum_{i=1}^{M_E}\frac{y_{E;i}^2}{\lambda_{E;i}} + \epsilon_E(\mathbf{x})\right] , \tag{2}$$

where the subscripts I and E specify the parameters of intrapersonal and extrapersonal PCA subspaces, respectively. The S_T similarity of Eq. (2) defines the MAP probabilities by the dimensionless Mahalanobis length of $\mathbf{x}$ in the corresponding PCA subspaces, which is independent of the prior probabilities of Ω_I and Ω_E.

2.2. *Refinement*

We now present a method that estimates the ML probability without calculating a PCA, but addresses the components x_i independently. It uses ML estimations of the mean μ_i and variance σ_i^2 for each dimension $i = 1, \ldots, N$ of $\mathbf{x}$ on its own. Therefore, this algorithm can deal with less training images than dimensions $(P < N)$ since the correlations between components x_i and x_j need not be estimated.

The ML probability can be either used for face detection in the one-class-fashion

$$S_{\mathrm{ML}}(\mathbf{x}) = \sum_{i=1}^{N} - \frac{(x_i - \mu_i)^2}{\sigma_i^2} \tag{3}$$

or on image similarities to classify image differences:

$$S_{\mathrm{BIC}}(\mathbf{x}) = \sum_{i=1}^{N} \left[- \frac{(x_i - \mu_{I;i})^2}{\sigma_{I;i}^2} + \frac{(x_i - \mu_{E;i})^2}{\sigma_{E;i}^2} \right] . \tag{4}$$

In the training stage mean μ_i and variance σ_i^2 are evaluated statistically on the basis of a training set $\{\mathbf{x}^{(p)} | p = 1, \ldots, P\}$ by assuming a Gaussian distribution:

$$\mu_i = \frac{1}{P} \sum_{p=1}^{P} x_i^{(p)} \quad i = 1, \ldots, N \tag{5}$$

$$\sigma_i^2 = \frac{1}{P-1} \sum_{p=1}^{P} (x_i^{(p)} - \mu_i)^2 \quad i = 1, \ldots, N. \tag{6}$$

For the S_{BIC} classification of image differences the parameters of the two classes Ω_I and Ω_E have to be estimated using Eqs. (5) and (6). For this purpose two distinct sets $\{\mathbf{x}_I^{(p)} | p = 1, \ldots, P_I\}$ and $\{\mathbf{x}_E^{(p)} | p = 1, \ldots, P_E\}$ are required. Each vector $\mathbf{x}_I^{(p)}$ of the first set is generated as the result of comparing two images of the same person, the second set contains vectors $\mathbf{x}_E^{(p)}$ created from different persons. Since no prior probabilities are needed, the sizes P_I and P_E of the training sets can differ without any problems.

As we apply the estimators to full feature vectors, no residual error ϵ needs to be estimated. Equations (3) and (4) can be computed very fast because no transformation mixing input data from different images is required. Thus, time complexity $O(N)$ is reached, in comparison to $O(N \cdot M)$ when using PCA. Another nice fact is that we have not lost the benefit of the Mahalanobis distance measure, the resulting similarity value still relies on dimensionless data. Beyond that, S_{BIC} has the ability to detect outliers if both intrapersonal and extrapersonal distances are high, but this property will not be discussed further in this paper.

Compared to Moghaddam *et al.*,[7,9] who used images or image differences for face detection and face recognition, respectively, we will use Gabor graphs and

Gabor graph similarities, which will be introduced in the next section, as input vectors $\mathbf{x}$.

3. Gabor Graphs

Mean-free Gabor wavelets are very useful for both face detection and face recognition.[5,15] With the center frequency $\mathbf{k}$ as parameter, their form in image and frequency domain is:

$$\psi_{\mathbf{k}}(\boldsymbol{\xi}_0) = \frac{\mathbf{k}^2}{\beta^2} e^{-\frac{\mathbf{k}^2\boldsymbol{\xi}_0^2}{2\beta^2}} \left[e^{i\mathbf{k}\boldsymbol{\xi}_0} - e^{-\frac{\beta^2}{2}} \right] \tag{7}$$

$$\check{\psi}_{\mathbf{k}}(\boldsymbol{\omega}) = e^{-\frac{\beta^2(\boldsymbol{\omega}-\mathbf{k})^2}{2\mathbf{k}^2}} - e^{-\frac{\beta^2\left(\boldsymbol{\omega}^2+\mathbf{k}^2\right)}{2\mathbf{k}^2}}. \tag{8}$$

3.1. *Gabor jets*

A typical parametrization for face recognition employs a family of $J = 40$ Gabor wavelets $\psi_{\mathbf{k}_j}$ $(j = 1, \dots, J)$ at five scale levels and eight directions. This family is visualized in frequency domain in Fig. 1. As input images are real-valued only half the frequency domain needs to be covered.

The convolution of an image with a Gabor wavelet $\psi_{\mathbf{k}_j}$ at image position $\boldsymbol{\xi}_0$ results in a complex-valued response $a_j \cdot e^{i\phi_j}$. The responses from all J Gabor wavelets taken at the same position $\boldsymbol{\xi}_0$ in the image are called a *Gabor jet* $\mathcal{J}$, which codes the texture information around the offset point. It is common to normalize Gabor jets according to Eq. (9) to unit length, which leads to local contrast normalization.

$$a_j = \frac{a_j}{\sqrt{\sum_{j'=1}^{J} a_{j'}^2}} \qquad j = 1, \dots, J. \tag{9}$$

3.2. *Model graphs*

Gabor jets $\mathcal{J}_l$ $(l = 1, \dots, L)$ are taken at different landmark positions $\mathcal{L}_l$ in the image and combined into a face graph $G = (\boldsymbol{\mathcal{J}}, \boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{E}})$, where $\boldsymbol{\mathcal{E}}$ denotes the links between neighboring nodes. An example for hand-labeled face graphs with 52 nodes, which are called *model graph*,[15] is shown in Fig. 3 in Sec. 4.

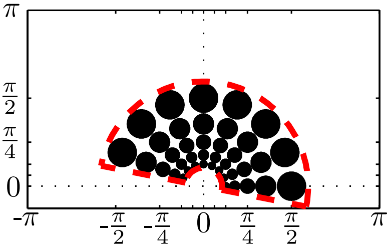


Fig. 1. The usual family of Gabor wavelets in frequency domain. Each circle depicts one Gabor wavelet, i.e. a Gaussian with radius $\beta/|\mathbf{k}|$. The dotted red line encircles the Gabor wavelet family.

Model graphs incorporate associational information on the one hand, giving the possibility to distinguish between faces and non-faces, and distinctive information that can be used for identity recognition on the other hand. The Graph similarity function:

$$S_{\mathcal{G}}(G, G') = \sum_{l=1}^{L} S_a(\mathcal{J}_l, \mathcal{J}'_l) \tag{10}$$

with[a]:

$$S_a(\mathcal{J}, \mathcal{J}') = \frac{\sum_{j=1}^{J} a_j \cdot a'_j}{\sqrt{(\sum_{j=1}^{J} a_j{}^2) \cdot (\sum_{j=1}^{J} a'_j{}^2)}} , \tag{11}$$

which originated from Lades *et al.*[5] and is used for both face detection and face recognition, disregards this fact. Beyond that, it neglects the landmark position and edge information completely. With a ML estimator from Sec. 2, it is easily possible to combine node position information $\boldsymbol{\mathcal{L}}$ or edge information $\boldsymbol{\mathcal{E}}$ with jet information $\boldsymbol{\mathcal{J}}$ in the landmark positioning and in the recognition algorithm, which are presented in Secs. 5 and 6, respectively. Furthermore, we are able to filter out the incorporated information, i.e. we use only the consistent parts of the model graphs to locate the face in the image.

3.3. *Model graph scaling and rotation*

To be able to detect faces at various scales and rotated in-plane we need training model graphs with different scales and angles, see Sec. 4 for details. To scale and rotate a model graph G first the node positions $\boldsymbol{\mathcal{L}}$ are transformed linearly using the center of gravity of the graph nodes as transformation center. The second step is to update the Gabor jets $\boldsymbol{\mathcal{J}}$. There are two ways of rotating and scaling Gabor jets. The first way — optimal, but slow — is to rotate and scale the image, again using the center of gravity of the graphs nodes as transformation center, and afterwards extract the new Gabor jets $\mathcal{J}_l$ from the scaled image at each of the L node positions $\mathcal{L}_l$ of the transformed graph.

The second way is to interpolate the values from the available complex-valued Gabor wavelet responses. For scaling factors below 1, the whole subband in frequency domain that is covered by the Gabor wavelets (cf. Fig. 1) is radially moved to higher frequencies, for factors above 1, the subband is radially moved in the direction of the frequency domain center $\boldsymbol{\omega}_0 = (0, 0)^T$. To accomplish rotation, the subbands have to be rotated around $\boldsymbol{\omega}_0$.

To achieve rotation and scaling of an unnormalized Gabor jet more than the usual 5 levels of Gabor wavelets are needed. The wavelets have to fill up the entire frequency domain, besides the center $\boldsymbol{\omega}_0$, i.e. the mean gray value[b] and the highest frequency parts. In Fig. 2, scaling and rotation of a long Gabor jet $\underline{\mathcal{J}}$ of size $\underline{J} = 9 \cdot 8$

[a] The denominator of Eq. (11) vanishes when using normalized Gabor jets with Eq. (9).

[b] The mean gray value cannot be reached because all Gabor wavelets are mean-free.

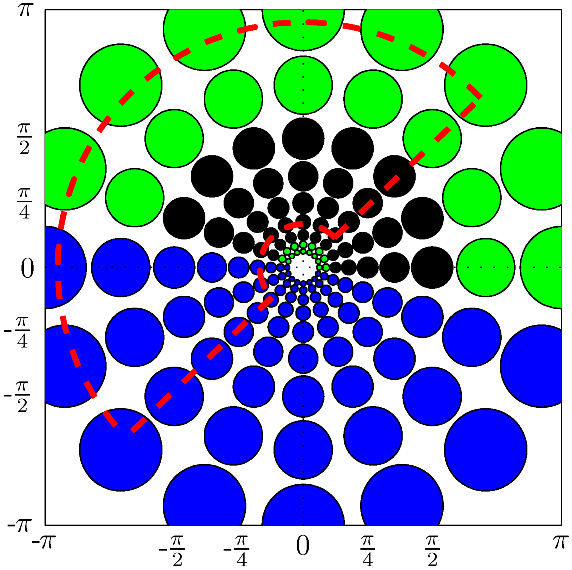


Fig. 2. Rotating and scaling of a Gabor jet. This figure visualizes the rotation by −55° and scaling with factor 1.6 of a long Gabor jet $\underline{\mathcal{J}}$ to a usual Gabor jet $\mathcal{J}$. The *black circles* display the responses of the usual family of Gabor wavelets, the *green circles* indicate the supplemental levels of Gabor wavelets. The *blue circles* show the second half of the frequency domain, which could be calculated from the other Gabor wavelet responses. Finally the *red border* marks the rotated and scaled Gabor jet $\mathcal{J}$.

to a usual Gabor jet $\mathcal{J}$ of size $J = 5 \cdot 8$ is visualized. The black circles adumbrate the usual set of 5 levels and 8 directions of Gabor wavelets, the green circles stand for the new Gabor wavelet levels and the blue ones indicate the second half of the frequency domain, which could be calculated from the other half. The partition outlined in red indicates the position of the rotated and scaled Gabor jet $\mathcal{J}$, which again consists of $5 \cdot 8$ approximated complex-valued Gabor wavelet responses. Each of these values is estimated as an interpolation of the surrounding 4 Gabor wavelet responses, the interpolation is done linearly in angular and logarithmically in radial direction and totalizes the real and imaginary parts of the complex responses.

The scaling factor is bounded below by 0.5 because in spatial domain Gabor kernels keep their size, independent of the size of the input image. At scaling factor 0.5, the responses of Gabor wavelets with the highest possible frequency π, i.e. wavelength 2 pixels are used. An upper bound for the scaling factor of the jets does not exist, but the size of the image limits the possibilities of scaling the graphs nodes $\boldsymbol{\mathcal{L}}$. Additionally, the Gabor wavelets in frequency domain get very small near $\boldsymbol{\omega}_0$.

4. Face Detection

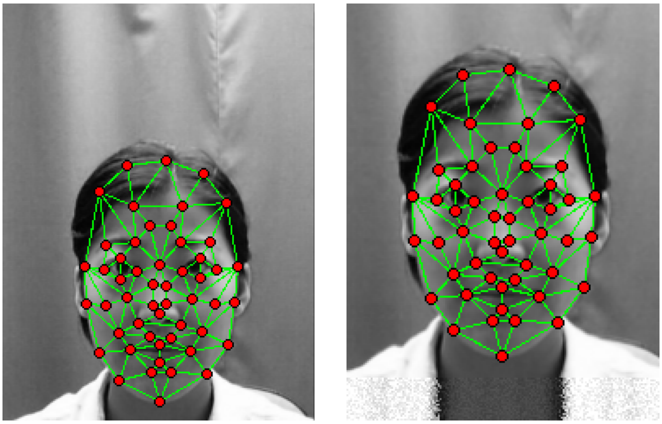
In this section, we define an estimator for face detection that is built upon the ML estimator from Sec. 2 on model graphs as representations for faces. For face detection only the absolute values of the Gabor jets are taken into account and the phases are discarded. The absolute values of Gabor wavelet responses are stable a few pixels around the landmark, but the phases are varying fast. It is crucial that the training images, especially the training model graphs, have about the same size because Gabor jets taken at the same landmark in two different images are only comparable if the underlying landmark texture patches are of comparable size.

4.1. *Training image preprocessing*

In the training set the scales and in-plane rotations of two facial images of the same person may differ. Therefore, we implemented an image preprocessing step that converts colored images to grayscale if needed and normalizes size and angle of the images and the corresponding hand-labeled graphs. Rotation normalization is done using the outer corners of the eyes so that the vertical positions of the eyes become equal. The scale factor s is calculated according to Eq. (12), where <original graph area> specifies the number of pixels covered by the graph in the original image and <preprocessed image size> is the number of pixels in the preprocessed image. Our choice for resolution of the preprocessed image is 168 × 224 pixel^2 and the graph is chosen to cover 45% of it.

$$s = \sqrt{\frac{< \text{preprocessed image size} > \cdot 0.45}{< \text{original graph area} >}}. \tag{12}$$

After the image is scaled, rotated and cut to the new size — padding missing pixels with noise — the analogously transformed graph is placed in the center of the preprocessed image. An example for the preprocessing step can be found in Fig. 3.



(a) original hand-labeled

(b) preprocessed hand-labeled

Fig. 3. Image and graph preprocessing. This figure shows a hand-labeled training graph in the original image, cf. Fig. 3(a), and the result of the preprocessing, see Fig. 3(b). In the lower part of the preprocessed image noise-padding was performed.

4.2. *Training*

In the training stage, the P preprocessed training images are convolved with $\underline{J} = 72$ Gabor wavelets in the usual 8 directions, but to accommodate scale changes the number of scale levels is increased to 9. At each hand-labeled landmark position $\mathcal{L}_l$ the $\underline{J}$ Gabor kernel responses are taken and combined into a training model graph $\underline{G}^{(p)}$, which consists of long Gabor jets $\underline{\mathcal{J}}_l^{(p)}$. With this family of $\underline{J}$ kernels, scaling factors between 0.5 and 2 and arbitrary rotation angles can be achieved, when Gabor jets $\mathcal{J}_l$ of the common dimensionality $J = 40$ are created with the graph transformation defined in Sec. 3.3.

For the detection of faces in all possible scales and angles a ML estimator (see Sec. 2) has to be trained for each single transformation. Additionally, different horizontal and vertical scaling factors s_x and s_y are introduced to be able to detect slender as well as broad faces. Thus, each ML estimator is responsible for a pair of scaling factors (s_x, s_y) and one rotation angle α and consisting of $L = 52$ Gabor jets $\mathcal{J}_l$ with each $J = 40$ absolute values of Gabor kernel responses. In total $L \cdot J = 2080$ means and variances are estimated for every (s_x, s_y, α) triplet.

To train the estimator with scaling factors s_x and s_y and rotation angle α, training databases are generated by rotating and scaling the Gabor jets of the training model graphs $\underline{G}^{(p)}$. This is only done for the scale factor $s = \sqrt{s_x \cdot s_y}$. For each training model graph $\underline{G}^{(p)}$, the Gabor jets of all nodes are interpolated according to s and α, resulting in L jets $\mathcal{J}_l$ with each J components, which are then normalized to unit Euclidean length using Eq. (9).

The training database of size $P \cdot L \cdot J$ is now fed into the ML estimator training. Mean $\boldsymbol{\mu}_l$ and variance $\boldsymbol{\sigma}_l^2$ are calculated for each node $l = 1, \ldots, L$ and each jet component $j = 1, \ldots, J$:

$$\mu_{l,j} = \frac{1}{P} \sum_{p=1}^{P} a_{l,j}^{(p)} \,, \tag{13}$$

$$\sigma_{l,j}^2 = \frac{1}{P-1} \sum_{p=1}^{P} (a_{l,j}^{(p)} - \mu_{l,j})^2. \tag{14}$$

The second part of the training stage is the approximation of the node positions $\boldsymbol{\mathcal{M}}$ of a mean model graph using Eq. (15). At the node positions of $\boldsymbol{\mathcal{M}}$, the Gabor jets will be extracted in the estimation stage. To calculate $\boldsymbol{\mathcal{M}}$, the center of gravity $C^{(p)}$ [cf. Eq. (16)] of the node positions of model graph $G^{(p)}$ is subtracted from each node position, such that the center of gravity of $\boldsymbol{\mathcal{M}}$ vanishes.

$$\mathcal{M}_l = \frac{1}{P} \sum_{p=1}^{P} (\mathcal{L}_l^{(p)} - C^{(p)}) \qquad l = 1, \ldots, L \tag{15}$$

$$C^{(p)} = \frac{1}{L} \sum_{l=1}^{L} \mathcal{L}_l^{(p)}. \tag{16}$$

4.3. *Estimation of scale, angle and location of a face*

In the evaluation stage a probe image is searched for a face with specific position, scale and in-plane rotation. It is first Gabor transformed, i.e. the responses of all wavelets are calculated at every position in the image by multiplying its Fourier transform with each Gabor kernel $\check{\psi}_{\mathbf{k}_j}$ out of the common family of 40 Gabor wavelets and applying an inverse Fourier transform to the results.

The goal is to find the best fitting parameter tuple $(\boldsymbol{\xi}^*, \mathbf{s}^*, \alpha^*)$ out of a finite set. This tuple must maximize the Maximum Likelihood similarity value S_{ML} over each position and each trained estimator:

$$(\boldsymbol{\xi}^*, \mathbf{s}^*, \alpha^*) = \arg \max_{(\boldsymbol{\xi}, \mathbf{s}, \alpha)} \{S^{[\mathbf{s},\alpha]}(\boldsymbol{\mathcal{J}}_{\boldsymbol{\xi},\mathbf{s},\alpha})\}, \tag{17}$$

where $\boldsymbol{\mathcal{J}}_{\boldsymbol{\xi},\mathbf{s},\alpha}$ are the Gabor jets taken at the positions $\boldsymbol{\mathcal{L}}_{\boldsymbol{\xi},\mathbf{s},\alpha}$ with

$$\mathcal{L}_l = \boldsymbol{\xi} + \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \cdot \begin{pmatrix} \mathcal{M}_{l;x} \cdot s_x \\ \mathcal{M}_{l;y} \cdot s_y \end{pmatrix}. \tag{18}$$

Thus, the landmark positions of the mean model graph $\boldsymbol{\mathcal{M}}$ are rotated and scaled according to $(\mathbf{s}, \alpha)$ and moved to the current position $\boldsymbol{\xi}$ in the image. $S^{[\mathbf{s},\alpha]}$ $(\boldsymbol{\mathcal{J}})$ depicts the S_{ML} value for the estimator that is responsible for $(\mathbf{s}, \alpha)$. It is calculated according to Eq. (19), which is similar to the ML estimator similarity measure of Eq. (3).

$$S^{[\mathbf{s},\alpha]}(\boldsymbol{\mathcal{J}}) = -\sum_{l=1}^{L}\sum_{j=1}^{J} \frac{(a_{l,j} - \mu_{l,j})^2}{\sigma_{l,j}^2}. \tag{19}$$

We implemented a three step search algorithm for the optimal parameter tuple $(\boldsymbol{\xi}^*, \mathbf{s}^*, \alpha^*)$. The first step tries to locate the face in the image using a global search with only very few scales and angles at sparse grid positions. The second and third steps refine the parameters found in the previous step with a local search, i.e. only the neighborhood of the previously found position is considered. The best found parameter tuple, i.e. the tuple with the highest estimated S_{ML} value is now used for landmark positioning, which is described in the next section.

5. Landmark Finding

After detection of the face the probe image is rotated and scaled according to the optimal parameters α^* and $s^* = \sqrt{s_x^* \cdot s_y^*}$, respectively, and cropped to the size of the preprocessed training images. Then an accordingly rescaled version of $\boldsymbol{\mathcal{M}}$ is placed into the center of the rotated and scaled images. An example for this automatic image processing step can be found in Fig. 4(b). This image is again Gabor transformed using Gabor kernels with, in turn, the common 5 levels and 8 directions, resulting in one Gabor jet at each pixel. To avoid the time-consuming recalculation of the Gabor transformation, it is also possible to use the found graph in the original image [cf. Fig. 4(a)] directly, but this should be considered only

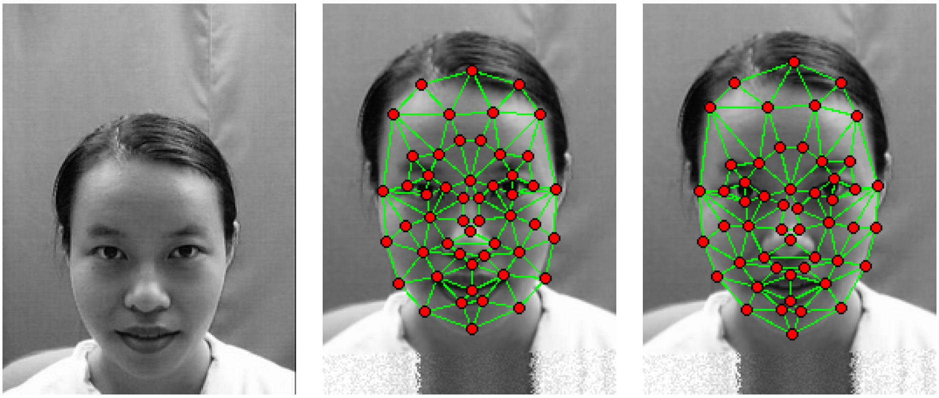

(a) original image (b) detected face (c) placed landmarks

Fig. 4. Face detection. This figure displays an example for the two steps of face detection and landmark localization. (a) The original image and (b) the detected face and the starting model graph for landmark localization. Finally, (c) presents the automatically placed landmarks.

when $s \approx 1$ and $\alpha \approx 0$ for all images in the database. These Gabor jets are normalized according to Eq. (9) and used for landmark localization by the ML estimator described in Sec. 5.1. The second estimator, which will be introduced in Sec. 5.2, uses the edge information $\boldsymbol{\mathcal{E}}$ of the graph to calculate a score that estimates the plausibility of the current graph structure.

5.1. *Local Gabor jet estimator*

The first landmark placement ML estimator is quite similar to the ML estimator used for face detection, but additionally to the absolute values a_j of the normalized Gabor jets, also the phases ϕ_j are taken into account.

In the training stage the mean absolute values $\boldsymbol{\mu}_a$ and the mean phases $\boldsymbol{\mu}_\phi$, as well as the variances of the absolute values $\boldsymbol{\sigma}_a^2$ and the phases $\boldsymbol{\sigma}_\phi^2$ over all training Gabor jets are calculated. These four vectors are evaluated separately for each landmark l, but the landmark index is discarded for legibility. The mean $\boldsymbol{\mu}_a$ and the variance $\boldsymbol{\sigma}_a^2$ of the absolute values are calculated in the same manner as for the face detection ML estimators, see Eqs. (13) and (14). The estimation of the mean phase $\boldsymbol{\mu}_{\phi;j}$ is more difficult. It is not calculated as the mean over all training phases, but as the phase of the mean complex value:

$$\mu_{\phi;j} = \arg\left(\frac{1}{P}\sum_{p=1}^{P} \mathcal{J}_j^{(p)}\right). \tag{20}$$

Also the phase variance $\sigma_{\phi;j}^2$ is calculated differently:

$$\sigma_{\phi;j}^2 = \frac{\sum_{p=1}^{P}(d_\phi(\phi_j^{(p)} - \mu_{\phi;j}))^2 \cdot a_j^{(p)}}{(P-1)\cdot \mu_{a;j}}, \tag{21}$$

with d_ϕ being the phase difference normalization function, i.e. $(\phi - \mu_\phi)$ is reduced modulo 2π into $[-\pi, \pi]$. The square of this phase difference is multiplied with the absolute value because phases are very unreliable when corresponding absolute values are low.

The estimation stage is again similar to the ML estimation of the face position in Eq. (19). The estimated likelihood score S_ϕ of the local jet estimator of Eq. (22) is a modification of Eq. (3), where again the phases of low absolute values are depreciated. Additionally, the value is normalized by the number of elements in the Gabor jet to ensure dimensionlessness and, thus, comparability with the results of the graph structure estimator that is described in the next section.

$$S_\phi\left(\mathcal{J}\right) = -\frac{1}{2J}\sum_{j=1}^{J}\left[\frac{\left(a_j - \mu_{a;j}\right)^2}{\sigma_{a;j}^2} + \frac{\left(\phi_j - \mu_{\phi;j}\right)^2 \cdot a_j}{\sigma_{\phi;j}^2 \cdot \mu_{a;j}}\right]. \tag{22}$$

5.2. *Graph structure estimator*

The second landmark placement estimator does not deal with the Gabor jets, but uses the links $\boldsymbol{\mathcal{E}}$ between the landmark positions $\boldsymbol{\mathcal{L}}$ of the graphs (cf. Fig. 3) to estimate the correctness of a node position in correlation to its linked nodes. Each landmark $\mathcal{L}_l$ is connected to a couple of neighboring landmarks $\mathcal{L}_{l_e}$, where E_l denotes the count of links for node $\mathcal{L}_l$ and $e = 1, \ldots, E_l$. The graph-theoretic order E_l may vary between different nodes, but is usually three or higher.

The graph structure estimator consists of two parts, namely, the horizontal and vertical components of the distance vectors $\boldsymbol{\delta}_{l,l_e} = \mathcal{L}_l - \mathcal{L}_{l_e}$. Again, the means $\mu_{x;l,l_e}, \mu_{y;l,l_e}$ and variances $\sigma_{x;l,l_e}^2, \sigma_{y;l,l_e}^2$ of the variables $\delta_{x;l,l_e}, \delta_{y;l,l_e}$ are estimated from the training data set. This is done for all landmarks l and their links l_e independently. This is better than considering only the Euclidean distance $\|\boldsymbol{\delta}_{l,l_e}\|$ between the two linked nodes because landmarks can have very different tendencies for horizontal and vertical movements, respectively.

In the estimation step, the S_δ similarity value is calculated as another modification of Eq. (3):

$$S_\delta\left(\mathcal{L}_l\right) = -\frac{1}{2E_l}\sum_{e=1}^{E_l}\left[\frac{\left(\delta_{x;l,l_e} - \mu_{x;l,l_e}\right)^2}{\sigma_{x;l,l_e}^2} + \frac{\left(\delta_{y;l,l_e} - \mu_{y;l,l_e}\right)^2}{\sigma_{y;l,l_e}^2}\right]. \tag{23}$$

5.3. *Landmark localization*

Exact landmark localization is done by local search for each node $l = 1, \ldots, L$ independently. In a small region $\mathcal{R}$ around the current landmark position $\mathcal{L}_l$ the best displacement vector $\mathbf{d}_l^*$ of maximal similarity score is computed according to Eq. (24), the constant c is defined below.

$$\mathbf{d}_l^* = \arg\max_{\mathbf{d}\in\mathcal{R}}\{S_\phi(\mathcal{J}_{\mathcal{L}_l+\mathbf{d}}) + c \cdot S_\delta(\mathcal{L}_l + \mathbf{d})\}. \tag{24}$$

The node positions are updated after all displacements are calculated. This update is done off-line because changing node positions on-line would interfere with the

S_δ score and the order of position updates would become important. We use an R-step iterative local search, where the influence of the graph structure, i.e. $c = \frac{r}{R}$ $(r = 0, \ldots, R-1)$ is increasing in each iteration. For the first round $c = 0$ is chosen so that, e.g. all nodes of one eye can move freely. In the last round c is near 1, so solitary outliers are recaptured. Finally, the Gabor jets at the detected landmark positions are extracted to a model graph G. An example for a resulting model graph is shown in Fig. 4(c).

6. Face Recognition

The model graphs that were found with the face detection and landmark localization algorithms are now recognized. The similarity between two graphs is calculated using the S_{BIC} estimator introduced in Sec. 2.2 based on a comparison between two graphs. We implemented several methods to compare two model graphs. Each of these comparison functions $f(\cdot)$ takes graphs G and G' as input and returns a vector $\mathbf{x}^{[f]}$. The dimensionalities of these vectors differ between the proposed methods, but this does not disturb the BIC classification. The components of $\mathbf{x}^{[f]}$ can be the results of any kind of measure between G and G', in our case we use similarity measures $S(\cdot)$ as well as distance metrics $D(\cdot)$.

In this section, we first present some comparison functions, namely, S_a, D_C and D_ψ, which use the absolute values of the normalized Gabor jet information $(\boldsymbol{\mathcal{J}}, \boldsymbol{\mathcal{J}}')$ of the two graphs, and later we introduce the metrics D_L and D_E, which deal with landmark positions $(\boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{L}}')$ and edges $(\boldsymbol{\mathcal{E}}, \boldsymbol{\mathcal{E}}')$, respectively. Finally, the $S_{\mathrm{BIC}}^{[f]}$ similarity measure will be constituted and the combination of different comparison functions is displayed.

6.1. *Jet similarity*

The first comparison function S_a is the Gabor jet comparison function defined in Eq. (11). The dimensionality $N^{[S_a]}$ of $\mathbf{x}^{[S_a]}$ is the number of nodes L and the coefficients are $x_l^{[S_a]} = S_a(\mathcal{J}_l, \mathcal{J}_l')$. The distribution of these jet similarities cannot be precisely Gaussian because they are confined between 0 and 1, which would violate one assumption of the Maximum Likelihood estimator training. Empirically however, the distribution of the similarity values is Gaussian shaped. An example for the intrapersonal and extrapersonal distribution of the S_a-measure can be found in Fig. 5(a).

6.2. *Gabor kernel response distance*

The second metric D_ψ goes one step further and uses all entries $a_{l,j}$ of the jets independently, resulting in a vector $\mathbf{x}^{[D_\psi]}$ of dimensionality $N^{[D_\psi]} = L \cdot J$. The coefficients are calculated according to Eq. (25):

$$x_i^{[D_\psi]} = D_\psi(a_{l,j}, a'_{l,j}) \qquad i = (l-1)\cdot J + j \tag{25}$$

with

$$D_\psi(a, a') = \begin{cases} \dfrac{|a - a'|}{|a| + |a'|}, & \text{if } |a| + |a'| > 10^{-6} \\ 0, & \text{else} \end{cases}. \tag{26}$$

Again, these distance values lie between 0 and 1, Fig. 5(c) shows a sample of their distribution. Apparently, this is quite different from all other distributions from this section because the overlap of intrapersonal and extrapersonal distribution is large and the maximum of both distributions is similarly near 0.

6.3. *Canberra jet distance*

The last comparison function dealing with Gabor jet information is the Canberra distance metric D_C shown in Eq. (27), which is a mixture of the two functions above. It uses the D_ψ distance from Eq. (26) totalized over all jet components to calculate $x_l^{[D_C]} = D_C(\mathcal{J}_l, \mathcal{J}'_l)$, a sample distribution can be seen in Fig. 5(b).

$$D_C(\mathcal{J}, \mathcal{J}') = \sum_{j=1}^{J} D_\psi(a_j, a'_j). \tag{27}$$

6.4. *Landmark position distance*

The landmark position distance D_L uses the landmark positions $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{\mathcal{L}}'$ of the two graphs to define a distance measure. According to Eq. (28), the distance of the nodes of corresponding landmarks $x_l^{[D_L]} = D_L(\mathcal{L}_l, \mathcal{L}'_l)$ is calculated, the determination of the mean movement R between $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{\mathcal{L}}'$ is given in Eq. (29). Figure 5(d) displays a sample distribution of D_L.

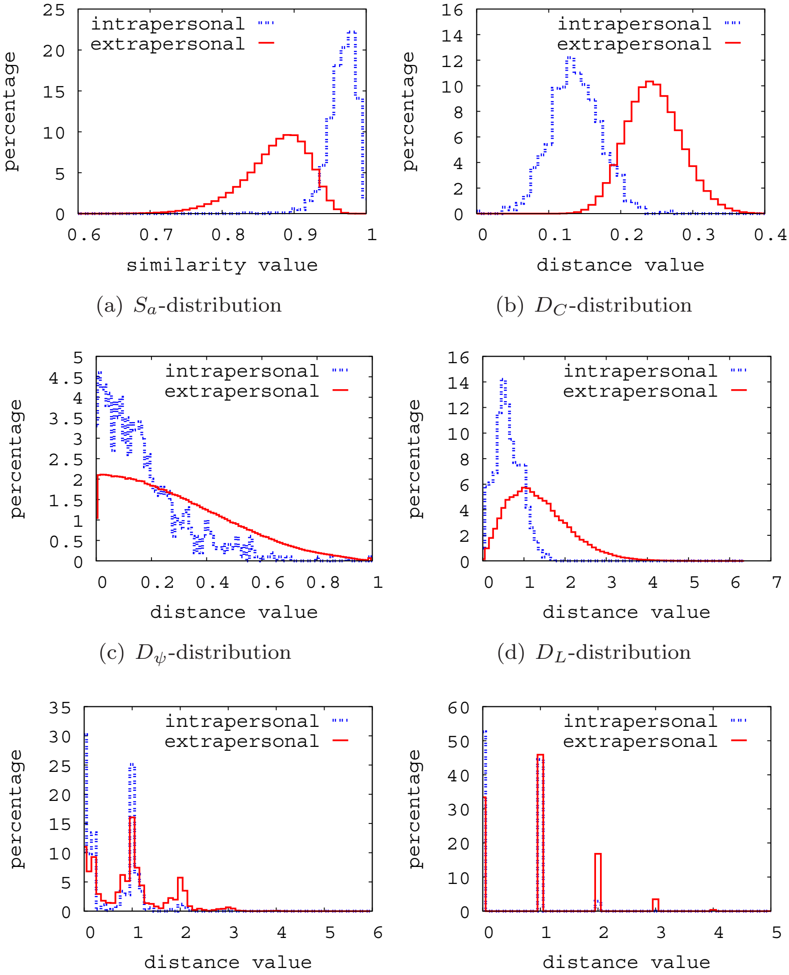
$$D_L(\mathcal{L}, \mathcal{L}') = \|\mathcal{L} - \mathcal{L}' + R\| \tag{28}$$

$$R = \frac{1}{L}\sum_{l=1}^{L}(\mathcal{L}'_l - \mathcal{L}_l). \tag{29}$$

Another approach uses — comparably to the graph structure ML estimator from Sec. 5.2 — the distance of $\mathcal{L}_l$ and $\mathcal{L}'_l$ in horizontal and vertical directions independently, denoted as $D_{L:h/v}$. This accounts for the fact that some node position changes are typically horizontal or vertical, e.g. opening the mouth will usually shift the lower lip node in vertical direction only.

6.5. *Edge length distance*

The last function D_E does not use the landmark positions directly, but calculates the length difference of the edges $\boldsymbol{\mathcal{E}}$ that links neighboring nodes. The components



(a) S_a-distribution

(b) D_C-distribution

(c) D_ψ-distribution

(d) D_L-distribution

(e) D_E-distribution

(f) $D_{E:h/v}$-distribution

Fig. 5. Sample distributions for comparison functions. This figure displays histograms of sample distributions for the introduced comparison functions generated from the ground truth positions of the FaceGen database (cf. Sec. 7.1) with 1000 intra- and 1998, 2000 extrapersonal training pairs. Histograms (a)–(d) were generated using right mouth corner landmarks (and the responses of a vertical Gabor wavelet at this position in case of the D_ψ metric), (e) and (f) show the distribution of the edge from nose-tip to the right alar wing of the nose. The bin-widths of the histograms are set to 0.01 for the functions S_a, D_C and D_ψ, and to 0.1 for D_L, D_E and $D_{E:h/v}$.

$x_e^{[D_E]} = D_E(\mathcal{E}_e, \mathcal{E}'_e)$ with $e = 1, \ldots, N^{[D_E]}$ and $N^{[D_E]} = E$ are calculated according to Eq. (30). One advantage of this measure is the independence of graph rotation.

$$D_E(\mathcal{E}, \mathcal{E}') = \big| \|\mathcal{E}\| - \|\mathcal{E}'\| \big|. \tag{30}$$

Like in the D_L measure we use the edge length in horizontal and vertical directions independently, denoted by $D_{E:h/v}$. Figure 5(e) shows the distribution of D_E, while

Fig. 5(f) displays the distribution of $D_{E:h/v}$ in horizontal direction only. One disadvantage of the D_E comparison function is that $x_e^{[D_E]}$ can represent only a few distinct values, $x_e^{[D_{E:h/v}]}$ is even limited to integer values. Thus, the Gaussian distribution precondition is not met at all.

6.6. *Recognition*

Each of these comparison functions $f(\cdot)$ is now used to calculate the $S_{\mathrm{BIC}}^{[f]}$ similarity value for two given graphs G and G'. Therefore, the $\mathbf{x}^{[f]}$ is determined as presented in the previous sections and the S_{BIC} similarity value in Eq. (31) is calculated as a variant of Eq. (4).

$$S_{\mathrm{BIC}}^{[f]}(\mathbf{x}^{[f]}) = \frac{1}{N^{[f]}} \sum_{i=1}^{N^{[f]}} \left[-\frac{(x_i^{[f]} - \mu_{I;i}^{[f]})^2}{\sigma_{I;i}^{[f]\,2}} + \frac{(x_i^{[f]} - \mu_{E;i}^{[f]})^2}{\sigma_{E;i}^{[f]\,2}} \right]. \tag{31}$$

Since the S_{BIC} similarity values are dimensionless, it is easily possible to combine the results of different comparison functions. Especially, the combination of Gabor jet dependent and graph structure dependent similarity values might enhance the recognition. Equation (32) exemplarily shows the combination of the Canberra distance D_C and the landmark position distance D_L.

$$S_{\mathrm{total}} = S_{\mathrm{BIC}}^{[D_C]}(G, G') + S_{\mathrm{BIC}}^{[D_L]}(G, G'). \tag{32}$$

7. Face Detection Experiments

In this section, we present the results of our face detection and landmark positioning experiments and compare them with the results of Elastic Bunch Graph Matching (EBGM).[15] To ensure comparable results, we used the same training graphs for the bunch graph and for the training of our maximum likelihood estimators. For the node positioning step of the EBGM algorithm the so-called *disparity move*, which uses the phases of the Gabor jets to estimate the displacement, has been replaced by an iterative local search because the latter generated better results.

7.1. *FaceGen*

The first database is the FaceGen database of artificial images that was also used by Müller[10] because it provides exact *ground truth* (GT) node position information. The database consists of 1000 identities each with two frontal images and medium random pose changes, the image size is 128 × 128 pixel^2, throughout. Some of the images including ground truth model graphs are shown in Fig. 6. The face detection estimators are trained with the first 100 identities,[c] which also make up the bunch graph used for the EBGM algorithm. The images of the remaining 900 identities are used for testing both algorithms.

[c] The preprocessing step (cf. Sec. 4.1) was left out because the data set is not much scaled or rotated in-plane.

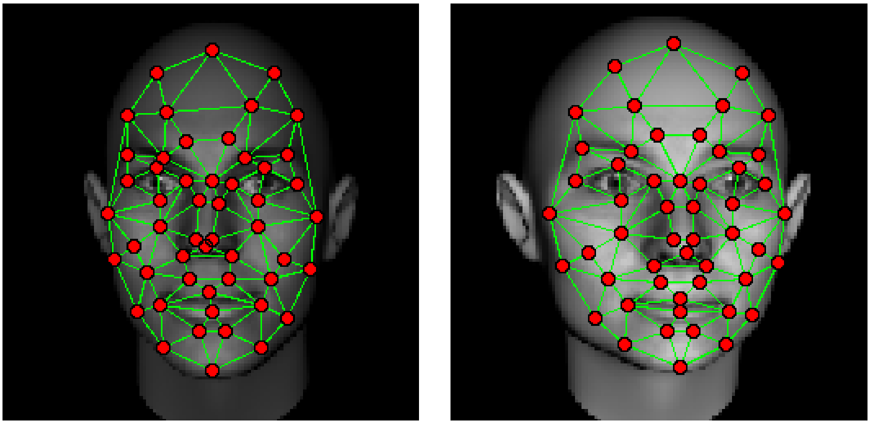
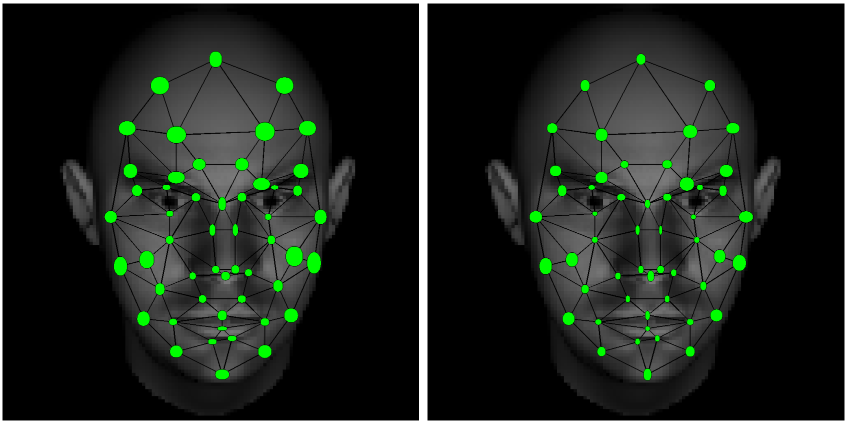


Fig. 6. FaceGen images. This figure shows exemplary images of two FaceGen identities including the graphs with ground truth (GT) node position information.



(a) EBGM

(b) ML

Fig. 7. Node positioning errors. In this figure the average node positioning errors on the FaceGen database can be inspected. (a) shows the results for the EBGM node positioning, (b) those of the Maximum Likelihood estimators.

Figure 7 shows the mean positioning errors, i.e. the average distance between found and GT node positions in horizontal and vertical directions for the two tested algorithms. One can see that the positioning errors for the ML approach [cf. Fig. 7(b)] are in general smaller than the positioning errors of the EBGM algorithm shown in Fig. 7(a). The nodes on landmarks with high structure information, e.g. the eye nodes or the mouth nodes are found more precisely than the others. Especially, the nodes on the outer rim are hard to localize, the EBGM algorithm seems to have even more problems with finding these landmarks.

The execution time of the face detection algorithms also differed a lot. While the ML detection algorithm — reusing the original images in the landmark positioning step — needed only 21 min to detect all 1800 test faces of the FaceGen database, the EBGM algorithm calculated 7.5 h on a Intel Xeon 3.2 GHz desktop PC with 2 GB of memory. Taking into account that the Gabor transformation, which was executed in both algorithms, took about 6 min of that time, the ML algorithm is about 30 times faster than the EBGM algorithm. Furthermore including that ML — in opposition to EBGM — also detected scale and in-plane rotation, the speedup factor is in the order of the number of training items, i.e. 200.

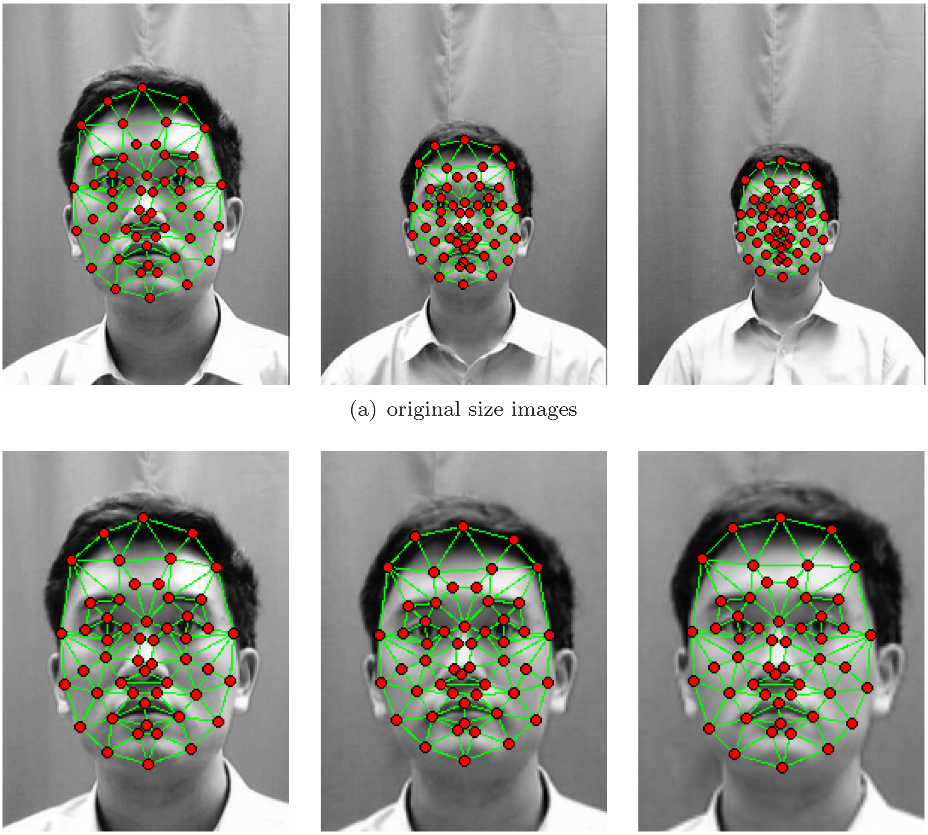
7.2. *CAS-PEAL*

The second inspected database is the CAS-PEAL database,[4] which consists of a large number of facial images of Chinese people. The images from the database were downscaled from their original size of 360×480 pixel^2 to 192×256 pixel^2 to reduce computation time. The database provides different kinds of image variations, in this study we used the sets with expression variation and size differences. The expression set **EXPR** provides frontal facial images with three different expressions per person: a neutral and a surprised expression and one with closed eyes. The second set **SIZE** contains frontal facial images in up to three different sizes, all showing a neutral expression.

The face detection training for the **EXPR** set was performed on 18 hand-labeled graphs of six people, three men and three women. The remaining 1113 images of 371 persons have been used for face finding experiments. The time required by the ML detection algorithm was 53 min, split up into 35 min needed for the Gabor transformation,[d] 14 min for the face detection and 4 min for landmark positioning. The needed time for the EBGM algorithm was 45 min and, thus, less, but the distribution is different: 12 min for Gabor transformation and 33 min for detection and landmark positioning. Since the CAS-PEAL database[4] does not provide ground truth information for all landmarks, there is no direct way to measure the correctness of the found positions. Therefore, it is only possible to compare the recognition results on this database with the recognition results of another algorithm — the EBGM algorithm — on the same set. This is done in Sec. 8.2.

The training of the face detection and landmark positioning ML estimators for the **SIZE** set was also performed on 18 hand-labeled graphs on preprocessed images, but unlike the **EXPR** set each image contained a different person in frontal pose with neutral expression. The remaining 524 images of 254 persons — none of them in the training set — were used for face detection. To display the results of the face detection and landmark positioning on the **SIZE** set all three found graphs of one person are shown in Fig. 8. In Fig. 8(a), the found graphs were placed into the original images, Fig. 8(b) contains the same results that were automatically scaled, rotated and centered using the scale, angle, and position obtained in the face

[d] The difference in Gabor transformation time is due to the additional transformation of the automatically scaled images after the face detection step, cf. Sec. 5.



(a) original size images

(b) automatically scaled and rotated images

Fig. 8. Face detection results. This figure shows exemplary results of the face detection and landmark positioning ML algorithm on the **SIZE** set of the CAS-PEAL database, where (a) presents the found graphs in the original images, and (b) displays the results in the automatically transformed images.

detection step, cf. Secs. 4 and 5. Except for minor scale and angle variations all faces were detected correctly. Since the landmark positioning step is executed on images normalized for scale and rotation [as shown in Fig. 8(b)], the node placement errors are — apart from possible misdetections and image scaling artifacts — independent of the size of the face in the original image.

7.3. *FRGC*

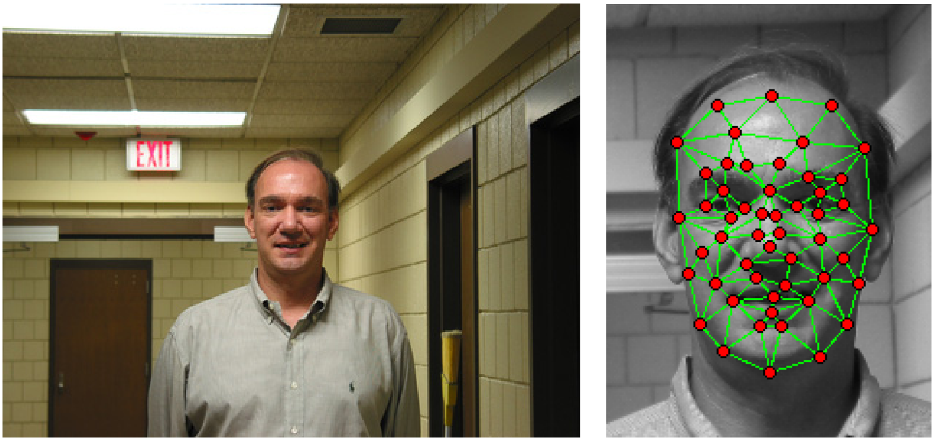
The last inspected database is the Face Recognition Grand Challenge (FRGC) database[11] in the version FRGC ver2.0, which provides a huge collection of high resolution facial images — sizes of the images are 1200 × 1600 pixel^2 or 1704 × 2272 pixel^2 — including hand-labeled eye, nose and mouth positions as well as standardized training, gallery and probe sets for different experiments. We here process Experiment 1, comparing each two images with controlled lighting conditions, i.e. studio portrait images, and Experiment 4, where the gallery images are controlled, but the probe images are taken under uncontrolled lighting conditions in landscape format. The FRGC ver2.0 database furthermore includes a baseline algorithm based on an eigenface approach and the recognition results of this algorithm are supplied.

The 12,776 colored training images of Experiments 1 and 4 of the FRGC ver2.0 data set[11] include facial images of 222 identities with both controlled and uncontrolled lighting conditions. For training purpose, the faces in the images were preprocessed as described in Sec. 4.1 using the provided eye positions, i.e. the faces are fitted into 168 × 224 pixel^2 images. The faces in the preprocessed images have a height of about 180 pixel. The model graphs of the training subset are generated automatically using our face detection and landmark positioning algorithm[e] with little scale variations. Since there are no hand-labeled graphs for the FRGC database, the face detection ML estimators were trained on hand-labeled graphs of the CAS-PEAL database. Figure 9 shows an example for the automatic training image preprocessing and the detected face graph. As the ML estimators were trained on Chinese people only, some node placement errors occurred.

The gallery subset of Experiments 1 and 4 consists of 16,028 controlled images. These images are down-scaled to 246 × 328 pixel^2 without cutting away the background, the faces shown there are about 150 pixel high. Experiment 4 furthermore uses 8014 uncontrolled images, which are downscaled to 560 × 420 pixel^2, as probe subset. For gallery and probe subsets, the face detection ML estimators are retrained on the previously found graphs of the FRGC ver2.0 training subset. In the detection step the global move is enabled, the inspected scales are in the range $[0.5, 1]$ and the tested angles are between $-10°$ and $10°$. Hence, in opposition to the baseline algorithm,[2] we do not use the provided positions, but our only assumption is that there is exactly one face with height between circa 90 and 180 pixel. After face detection the original size images are scaled and rotated — according to the detected graphs scale and angle — to image size 168×224, cf. Sec. 5. Two examples of the fully automatic detection, scale and angle normalization and landmark positioning are shown in Fig. 10. Although the ML estimators were trained on partly poorly labeled landmark positions, the node placement errors seem to be small, besides a couple of misdetected faced in the uncontrolled images.

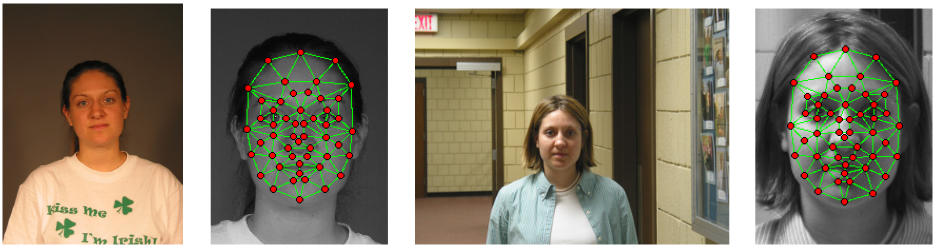
The execution time of the ML face detection for one of the controlled images is about 10 s, which are split up into 4.6 s of Gabor transformation, 4.1 s for face detection and finally 0.75 s for landmark positioning. The training of all needed ML estimators, which had to be done only once, with all 12,776 training graphs took about 4 h.

[e] To avoid misdetections in the training subset the global move step was not used, but the center of the image was taken as the center of the face.



(a) original training image (b) training image with graph

Fig. 9. FRGC training image preprocessing. This figure shows the semi-automatic preprocessing of the FRGC training images. In (a) one training image with uncontrolled lighting conditions is presented, (b) displays the preprocessed image with the semi-automatically found face graph.



(a) gallery image (b) gallery graph (c) probe image (d) probe graph

Fig. 10. FRGC gallery and probe image processing. This figure shows the fully automatic processing of FRGC gallery and probe images. (a) and (b) contain a controlled gallery image and the automatically found, rotated and scaled face graph, while (c) and (d) display an uncontrolled probe image and the detected graph.

8. Face Recognition Experiments

In this section we present the recognition results of our S_{BIC} similarity function when using the comparison functions $f(\cdot)$ introduced in Sec. 6. These results are compared with the recognition results that could be achieved by the model graph similarity function $S_{\mathcal{G}}$ from Eq. (10), which is adapted to f, depicted by $S_{\mathcal{G}}^{[f]}$. The adaptation[f] of $S_{\mathcal{G}}$ to the Canberra distance comparison function D_C is given in Eq. (33). It is easy to see that $S_{\mathcal{G}}^{[D_C]}$ and $S_{\mathcal{G}}^{[D_\psi]}$ refer to the same similarity function:

$$S_{\mathcal{G}}^{[D_C]}(G, G') = -\sum_{l=1}^{L} D_C(\mathcal{J}_l, \mathcal{J}'_l) = -\sum_{l=1}^{L}\sum_{j=1}^{J} D_\psi(a_{l,j}, a'_{l,j}) \,. \tag{33}$$

[f] The negative sign is included to convert the distance metric to a similarity measure.

To determine the performance of the algorithms the commonly used Receiver Operating Characterics (ROC) is calculated by computing the Verification Rate[g] (VR) over the False Acceptance Rate (FAR). To provide a single number for easy comparison the VR at FAR 0.1% is used.

8.1. *FaceGen*

We first tested the S_{BIC} similarity function on the FaceGen database using the test set from the landmark detection step, which contains 1800 graphs of 900 identities. This set is again split up into three distinct subsets: training, gallery and probe. 100 FaceGen identities, which offer 100 intrapersonal and 19800 extrapersonal pairs, are used for training. With these pairs, the BIC estimators for the different comparison functions are trained, i.e. mean and variance vectors for the intrapersonal and extrapersonal classes are calculated. For the remaining 800 identities one graph per identity is used as gallery and one as probe.

The recognition results of the different comparison functions are presented in Table 1. The first three columns display the Gabor jet comparison functions, whereas the second block contains the functions that only compare graph structures. Table 1(a) shows the verification rates for the adapted graph similarity function, where $S_{\mathcal{G}}^{[S_a]}$ refers to the original graph similarity function from Eq. (10). One can see that the Gabor jet-dependent comparison functions are not reliant on precise node positioning since the verification rates are about the same for the three investigated node position algorithms GT, EBGM and ML. The ground truth node positions GT give — as suggested — the best verification. For the graph structure comparison functions, the correct landmark placement is much more important. While GT can achieve verification rates around 45% — besides the malfunctioning $D_{L:h/v}$ measure — the automatically found positions of ML and EBGM lead to about one-third of these rates.

Table 1(b) contains the results for the trained S_{BIC} function. Nearly all rates increased. Again, the choice of the node positioning algorithm does not change the Gabor jet-dependent verification rates remarkably, the slightly better algorithm for S_a and D_C is ML. The advantage of the D_ψ comparison function over D_C and S_a is most probably due to the nature of the FaceGen data because the texture of the images is very smooth and there is no noise in it. The verification rates of the graph structure comparison functions of the ML detection algorithm are comparable with the results of the EBGM algorithm when using the untrained $S_{\mathcal{G}}$ function [see Table 1(a)]. In Table 1(b), this changes the ML positions outperform the EBGM positions. Seemingly, the landmark detection errors of the ML algorithm are more

[g] The Verification Rate is calculated as 100% — False Rejection Rate (FRR) and is equal to the Correct Acceptance Rate (CAR).

Table 1. FaceGen verification results. This table shows the verification rates at FAR 0.1% for the recognition experiments on the FaceGen database when using comparison functions $f(\cdot)$ and different landmark positioning algorithms.

	S_a	D_C	D_ψ	D_L	$D_{L:h/v}$	D_E	$D_{E:h/v}$
(a) Adapted graph similarity function $S_{\mathcal{G}}^{[f]}$							
GT	49.3%	66.9%	66.9%	44.1%	16.3%	45.8%	46.9%
EBGM	48.3%	63.4%	63.4%	17.4%	6.6%	13.6%	14.5%
ML	48.4%	66.2%	66.2%	18.7%	6.0%	13.6%	15.5%
(b) BIC similarity function $S_{BIC}^{[f]}$							
GT	85.4%	88.8%	92.3%	82.9%	50.3%	75.0%	79.8%
EBGM	85.4%	88.5%	90.4%	29.0%	5.0%	15.1%	18.9%
ML	86.4%	88.9%	92.0%	37.6%	9.5%	23.5%	25.4%

Table 2. Combination of comparison functions. In this table, the verification rates for the combination of the D_C comparison function with the node position comparison functions D_L, $D_{L:h/v}$, D_E and $D_{L:h/v}$ are displayed. The verification results were calculated using graphs of the three different node positioning algorithms GT, EBGM and ML. The last column repeats the results for the single D_C comparison function.

	D_L	$D_{L:h/v}$	D_E	$D_{E:h/v}$	None
GT	92.1%	89.3%	90.0%	89.8%	88.8%
EBGM	86.4%	86.8%	86.4%	86.9%	88.5%
ML	86.8%	85.8%	87.0%	87.5%	88.9%

predictable.[h] Accessorily, the 82.9% verification rate of $S_{\text{BIC}}^{[D_L]}$ on the GT data shows that the landmark positions hold nearly as much identity determining information as the texture does.

The verification rate of the S_a and D_C comparison functions can be increased when they are combined with graph structure information. This can easily be achieved by summing up the S_{BIC} similarity values from different comparison functions, as shown in Eq. (32) in Sec. 6. Table 2 shows the recognition results of the combination of the D_C function with each graph structure comparison function. When using the GT landmark positions, the combination of D_C with D_E, D_L or one of their h/v modifications outperforms the verification rates of the single D_C comparison function. The node positions of both positioning algorithms EBGM and ML are, however, not precise enough to enhance the recognition rate of D_C.

[h] The predictability of the error is a possible cause of the fact that the recognition rate on the ML graphs is higher than on the ground truth node positions for $S_{\text{BIC}}^{[S_a]}$. The dependency of the recognition on systematic positioning errors will be investigated in future work.

8.2. *CAS-PEAL*

For the **EXPR** set of the CAS-PEAL database the recognition results on the graphs found with ML are compared to the recognition results attained with the EBGM algorithm, too. To assure a fair comparison the graphs from the ML algorithm that are placed in the original images are taken. Similar to the FaceGen experiments, the two similarity functions $S_{\mathcal{G}}$ and S_{BIC} are tested using the comparison functions defined in Sec. 6. The test subset from the face detection step, which contains the automatically found graphs of the **EXPR** set, is split up into training, gallery and probe subsets. The training set contains all graphs of 100 random persons. For the remaining 271 persons, the graph of one *random* expression is put into the gallery set, the graphs of the other two expressions are accumulated in the probe subset.

The verification rates — averaged over 10 recognition experiments with different random training/gallery/probe-subsets[i] — are presented in Table 3. The Gabor jet comparison functions show the same trend as in the experiments with artificial images. The Canberra distance D_C is able to outperform S_a in all tests, the graphs generated with ML are better suited for face recognition than the EBGM graphs. The S_{BIC}-training further improves the verification rate, but D_ψ is no longer able to exceed D_C. The automatically found node positions, however, are not good enough since the verification rates only reach 15%.

The graphs in the **SIZE** subset of the CAS-PEAL database are easier to recognize when they were found correctly since they all enclose the same neutral expression. The EBGM algorithm is not designed to find faces on different scales. Therefore, it is not reasonable to compare our ML algorithm with it. Thus, in Table 4 only, the recognition results of the ML approach are shown, again using the different comparison functions and the two similarity measures $S_{\mathcal{G}}$ and S_{BIC}. As done for the **EXPR** subset, training, gallery and probe subset for face recognition were created from the face detection test set by putting random 100 identities into training set. For the remaining identities one random graph per identity was put into the gallery and the other graphs were used as probes. The recognition experiments were repeated ten times with different random training/gallery/probe subsets.

Table 4(a) displays the average verification rate generated from the found graphs taken from the unpreprocessed images, examples are shown in Fig. 8(a). All comparison functions are not designed to compare graphs of different sizes. Thus, the verification is very poor for the untrained $S_{\mathcal{G}}$ function. Nevertheless, the intrapersonal/extrapersonal training is able to enhance the verification rate to 43% for the D_ψ comparison function. The results from Table 4(b) are generated with the found graphs that were automatically transformed to fit into 168×224 pixel^2 images, see Fig. 8(b) for some examples. The achieved verification rates for the Gabor jet comparison functions are much higher now, $S_{\mathcal{G}}^{[S_a]}$ is the best function. Also the

[i] The same ten random subset splits were used in all experiments.

Table 3. CAS-PEAL **EXPR** verification results. This table shows the verification rates for the experiments on the **EXPR** set of the CAS-PEAL database when using comparison functions $f(\cdot)$ with $S_{\mathcal{G}}$ [cf. Table 3(a)] and S_{BIC} [see Table 3(b)] and the two landmark positioning algorithms EBGM and ML.

	S_a	D_C	D_ψ	D_L	$D_{L:h/v}$	D_E	$D_{E:h/v}$
			(a) Verification rates for the adapted graph similarity function $S_{\mathcal{G}}^{[f]}$				
EBGM	58.2%	69.0%	69.0%	15.7%	12.5%	12.1%	12.9%
ML	63.2%	77.0%	77.0%	7.9%	5.9%	8.8%	10.9%
			(b) Verification rates for BIC similarity function $S_{BIC}^{[f]}$				
EBGM	66.9%	86.4%	85.1%	14.6%	14.3%	12.1%	14.2%
ML	77.9%	90.5%	89.9%	8.6%	10.3%	6.4%	9.2%

Table 4. CAS-PEAL **SIZE** verification results I. This table shows the verification rates from the **SIZE** subset of the CAS-PEAL database using the graphs found with the ML algorithm on the original images [see Fig. 8(a)] and on the automatically scaled and rotated images [cf. Fig. 8(b)].

	S_a	D_C	D_ψ	D_L	$D_{L:h/v}$	D_E	$D_{E:h/v}$
			(a) original size images				
$S_{\mathcal{G}}$	1.5%	2.7%	2.7%	0.0%	0.0%	0.0%	0.0%
S_{BIC}	32.4%	38.0%	43.0%	2.5%	2.3%	1.8%	3.3%
			(b) automatically scaled images				
$S_{\mathcal{G}}$	89.3%	86.7%	86.7%	20.8%	13.7%	19.0%	21.1%
S_{BIC}	87.1%	86.0%	91.0%	17.4%	15.6%	20.6%	23.2%

verification rates on the graph structure increased, but the False Rejection Rate is always above 75% for FAR = 0.1%.

The size of the images is an important factor for recognition since Gabor wavelets cover a fixed region that does not scale with the image. Thus, for different image sizes different patches of texture information are coded in the Gabor jets. In Table 5, the recognition results for different resolutions are presented[j]. Astonishingly, the S_a comparison function is pretty stable for all investigated image resolutions, whereas the D_C comparison function declines with increasing resolution. Nevertheless, D_C is the better comparison function since the verification rate of 95.8% for resolution 120×160 pixel^2 is higher than the 92.5% of the S_a comparison function. The impact of image processing artifacts and node position rounding errors, which were induced in the scaling step, on the recognition remains to be investigated. The trained S_{BIC} similarity is not able to beat the verification rate of the $S_{\mathcal{G}}$ function, besides $S_{\mathrm{BIC}}^{[D_C]}$ and $S_{\mathrm{BIC}}^{[D_\psi]}$ for higher resolutions. If the variations in the image are too small, i.e.

[j] The landmark and edge dependent comparison functions are left out because the verification rates can only get worse when the graphs are scaled.

Table 5. CAS-PEAL **SIZE** verification results II. This table displays the verification rates from the **SIZE** subset of the CAS-PEAL database, using the graphs found with the ML algorithm on the automatically scaled and rotated images of different resolutions.

	$S_{\mathcal{G}}$		S_{BIC}		
Resolution	S_a	D_C	S_a	D_C	D_ψ
90 × 120	89.8%	94.8%	87.2%	94.4%	91.8%
120 × 160	92.5%	95.8%	90.6%	94.6%	92.0%
168 × 224	89.3%	86.7%	87.2%	86.0%	91.0%
240 × 320	89.0%	69.4%	87.3%	70.5%	93.2%
300 × 400	85.2%	51.3%	84.6%	52.5%	92.2%

Table 6. CAS-PEAL recognition rates. For comparison with Refs. 3 and 4, we also give the recognition rates for parts of the conditions of the **EXPR** set from Table 3 in Table 6(a). The recognition rates of the **SIZE** set [see Table 6(b)] employed the resolution 120 × 160 pixel^2 (cf. Table 5). Table 6(c) shows the recognition rates from Ref. 4 without illumination normalization, the value for LGBPHS has been estimated from Fig. 15 in Ref. 3.

	S_a	D_C	D_ψ
(a) ML graphs on **EXPR**			
$S_{\mathcal{G}}$	81.9%	88.6%	—
S_{BIC}	90.1%	95.9%	95.7%

	S_a	D_C	D_ψ
(b) ML graphs on **SIZE**			
$S_{\mathcal{G}}$	98.2%	99.5%	—
S_{BIC}	97.6%	99.4%	98.2%

	PCA	PCA+LDA	Gabor PCA+LDA	LGBPHS
(c) Recognition rates for other algorithms				
EXPR	53.7%	71.3%	90.6%	95%
SIZE	74.2%	93.5%	100.0%	99%

the variances of Gabor jet similarities are low, the BIC training does not seem to be profitable.

The CAS-PEAL database[3,4] is well suited for comparison of our method with other classical methods, namely eigenfaces (PCA),[14] fisherfaces (LDA),[1] the combination of LDA with Gabor coefficients,[6] and the novel Local Gabor Binary Pattern Histogram Sequences (LGBPHS).[16] Table 6(c) shows the recognition results of these algorithms on the **EXPR** and **SIZE** sets as reported in Refs. 3 and 4. Using our ML algorithm on the **EXPR** set [cf. Table 6(a)] with the S_{BIC} similarity function and either D_C or D_ψ yields significantly higher recognition rates than all baseline algorithms, only LGBPHS shows a comparable performance. Our results on the **SIZE** set are as shown in Table 6(b), below the very high recognition rates in Refs. 3 and 4.

All those results are only partly comparable because our experiments had more difficult conditions than the ones in Ref. 3 and 4 in the following respects:

- Geometric normalization was fully automatic.
- No masking has been performed.

- Identities used for training have not been used for gallery or probe set.
- For the **EXPR** set the expression used for the gallery was chosen randomly for each identity.

Thus, the system presented here performs better or equal with fewer manual interactions.

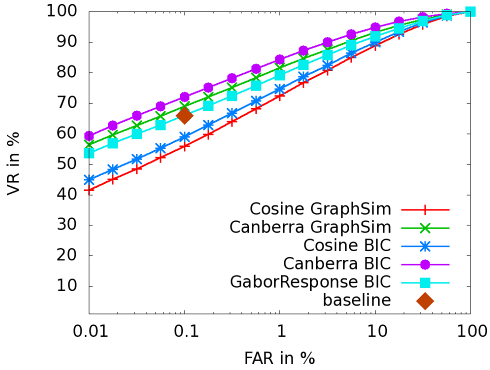
8.3. *FRGC*

The recognition experiments on the FRGC ver2.0 database[11] are executed using the common experimental setup provided by the database. To train the BIC classifiers we redetected the training graphs by training the face detection ML estimators with the previously found training graphs and redo the face detection from Sec. 7.3 on the preprocessed training images. This step was included to assure that training and gallery/probe graphs were detected similarly. The BIC training, finally, was applied on all pairs of training graphs generated from controlled and uncontrolled images for Experiment 1, whereas for Experiment 4 the BIC training used only pairs where a controlled graph is compared to an uncontrolled. Hence, the training was performed on 371,052 intrapersonal and 81,235,648 extrapersonal pairs for Experiment 1 and 188,719 intrapersonal and 40,617,824 extrapersonal pairs for Experiment 4.

For executing Experiment 1 all controlled gallery graphs are compared to each other, resulting in a 16,028 × 16,028 similarity matrix. Experiment 4 has different gallery and probe sets generating 16,028 × 8014 similarity scores. Parts of these similarity matrices are selected to create the Receiver Operating Characterics ROC III, which are defined by the FRGC ver2.0 database.[11] The performance, i.e. the verification rate for FAR 0.1% of the baseline PCA algorithm on ROC III is about 66% for Experiment 1 and 12% for Experiment 4.

Figure 11 shows the ROC III curves of the different comparison functions for Experiment 1. With 72%, D_C is outperforming the S_a measure and the baseline performance of 66% is outrivaled by the D_C comparison function, too. The graph structure comparison functions are — due to the indirect landmark detection training — nearly worthless. The execution time of the BIC classifier are — compared to the number of calculated similarities — quite low. The BIC training — using over 81 million graph pairs — took 26 min for S_a, 55 min. for D_C and 1.1 h for D_ψ, whereas the 256 million similarity scores were calculated in 1.4 h, 3.1 h and 3.8 h, respectively. The calculation of the untrained $S_\mathcal{G}$ similarities needed about the same time: 1.4 h for S_a and 3.1 h for D_C.

The results of Experiment 4 on ROC III are as shown in Fig. 12, worse and the baseline performance of 12% cannot be reached. There are a couple of possible reasons for that, e.g. misdetections of some probe graphs. Nonetheless, the BIC training is able to boost the verification rate for S_a, D_C and D_ψ. The D_ψ measure is obviously most appropriate to learn uncontrolled lighting changes since it processes the (normalized) responses of each Gabor kernel independently.

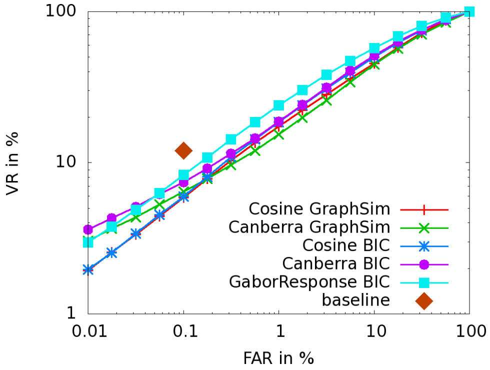


(a) ROC curve

Algorithm	Performance
baseline	66.0%
$S_{\mathcal{G}}^{[S_a]}$	55.9%
$S_{\mathcal{G}}^{[D_C]}$	68.9%
$S_{BIC}^{[S_a]}$	59.0%
$S_{BIC}^{[D_C]}$	72.0%
$S_{BIC}^{[D_\psi]}$	66.1%
$S_{BIC}^{[D_E]}$	5.5%
$S_{BIC}^{[D_L]}$	5.7%

(b) VR at FAR 0.1%

Fig. 11. FRGC Experiment 1. This figure encloses the verification results for Experiment 1 of the FRGC ver2.0 database, generated on ROC III. (a) shows the ROC curves on logarithmical FAR axis for the Gabor jet dependent functions, while (b) holds the verification rates for FAR 0.1%.



(a) ROC curve

Algorithm	Performance
baseline	12.0%
$S_{\mathcal{G}}^{[S_a]}$	5.9%
$S_{\mathcal{G}}^{[D_C]}$	6.4%
$S_{BIC}^{[S_a]}$	6.0%
$S_{BIC}^{[D_C]}$	7.4%
$S_{BIC}^{[D\ \ ]}$	8.3%
$S_{BIC}^{[D_E]}$	1.4%
$S_{BIC}^{[D_L]}$	1.5%

(b) VR at FAR 0.1%

Fig. 12. FRGC Experiment 4. This figure shows the verification results for Experiment 4 of the FRGC ver2.0 database, generated on ROC III. (a) holds the ROC curves with logarithmical FAR and logarithmical VR axis for the Gabor jet comparison functions, while (b) displays the verification rates for FAR 0.1%.

9. Summary and Outlook

We have presented fully automatic face detection and face recognition algorithms based on Maximum Likelihood estimators and Bayesian intrapersonal/extrapersonal classifiers that work with Gabor graphs. We have shown that the face detection algorithm is able to detect faces with different scales and in-plane rotation angles. We have tested this algorithm on artificial facial images as well as on the CAS-PEAL and the FRGC databases. We implemented several comparison functions on Gabor graphs and showed that the trained intrapersonal/extrapersonal classifier outperforms the common[15] and the adapted graph similarity function as well as the eigenface baseline algorithm on Experiment 1 of FRGC ver2.0.[11] The verification rates of commercial systems as reported in Ref. 12 could not be reached. On the expression data of CAS-PEAL our algorithm clearly performs better than all algorithms tested in Ref. 4 and is comparable to LGBPHS.[16]

We showed that combining Gabor jet and node position information is able to beat the verification rates of the Gabor jet when ground truth node position information is available. We also tested the recognition algorithm on different image resolutions.

In future work, we will investigate the dependence of the BIC algorithm on correct scale and angle detection as well as on correctly placed nodes. Another study will further determine the best image size for face recognition when using the Bayesian intrapersonal/extrapersonal classifier on different Gabor jet dependent comparison functions.

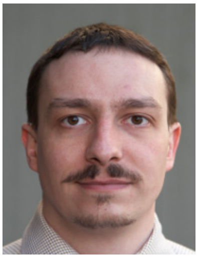
Acknowledgments

We gratefully acknowledge funding from Deutsche Forschungsgemeinschaft (WU 314/2-2 and WU 314/5-2). Portions of the research in this paper use the CAS-PEAL face database collected under the sponsorship of the Chinese National Hi-Tech Program and ISVISION Tech. Co. Ltd.

References

1. P. Belhumeur, J. Hespanha and D. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, *IEEE Trans. Patt. Anal. Mach. Intell.* **19**(7) (1997) 711–720.
2. J. R. Beveridge, D. S. Bolme, B. A. Draper and M. Teixeira, The CSU face identification evaluation system, *Mach. Vis. Appl.* **16**(2) (2005) 128–138.
3. W. Gao, B. Cao, S. Shan, X. Chen, D. Zhou, X. Zhang and D. Zhao, The CAS-PEAL large-scale Chinese face database and baseline evaluations, *IEEE Trans. Syst. Man Cybern. Part A* **38**(1) (2008) 149–161.
4. W. Gao, B. Cao, S. Shan, D. Zhou, X. Zhang and D. Zhao, The CAS-PEAL large-scale Chinese face database and baseline evaluations, Technical Report JDL-TR-04-FR-001, Joint Research & Development Laboratory for Face Recognition, Chinese Academy of Sciences (2004).
5. M. Lades, J. C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Würtz and W. Konen, Distortion invariant object recognition in the dynamic link architecture, *IEEE Trans. Comput.* **42**(3) (1993) 300–311.
6. C. Liu and H. Wechsler, Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition, *IEEE Trans. Imag. Process.* **11**(4) (2002) 467–476.
7. B. Moghaddam and A. Pentland, Probabilistic visual learning for object detection, *Int. Conf. Computer Vision (ICCV'95)*, Cambridge, USA (June 1995), pp. 786–793.
8. B. Moghaddam and A. Pentland, Probabilistic visual learning for object representation, *IEEE Trans. PAMI* **19**(7) (1997) 696–710.

9. B. Moghaddam, W. Wahid and A. Pentland, Beyond eigenfaces: probabilistic matching for face recognition, *IEEE Int. Conf. Automatic Face and Gesture Recognition* (1998), pp. 30–35.
10. M. Müller, Finden von Punktkorrespondenzen zwischen Gesichtsbildern, Internal Report IRINI 2004-01, Institut für Neuroinformatik, Ruhr-Universität Bochum, D-44780 Bochum, Germany, (March 2004).
11. P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer and W. Worek, Preliminary face recognition grand challenge results, *Proc. 7th Int. Conf. Automatic Face and Gesture Recognition* (IEEE Computer Society, 2006) pp. 15–24.
12. P. Phillips, T. Scruggs, A. J. O'Toole, P. J. Flynn, K. W. Bowyer, C. L. Schott and M. Sharpe, FRVT 2006 and ICE 2006 large-scale results, Technical Report NISTIR 7408, http://www.frvt.org/ (2007).
13. M. L. Teixeira, *The Bayesian Intrapersonal/Extrapersonal Classifier*, Ph.D. thesis, Computer Science, Colorado State University (July 2003).
14. M. Turk and A. Pentland, Eigenfaces for recognition, *J. Cogn. Neurosci.* **3**(1) (1991) 71–86.
15. L. Wiskott, J.-M. Fellous, N. Krüger and C. von der Malsburg, Face recognition by elastic bunch graph matching, *IEEE Trans. Patt. Anal. Mach. Intell.* **19**(7) (1997) 775–779.
16. W. Zhang, S. Shan, W. Gao, X. Chen and H. Zhang, Local Gabor binary pattern histogram sequence (LGBPHS): a novel non-statistical model for face representation and recognition, *Proc. ICCV* (2005), pp. 786–791.



Manuel Günther studied computer science and mathematics at the Technical University of Ilmenau, Germany, starting in 2000. After receiving the diploma in 2004, he started his Ph.D. research at the Institute for Neural Computation of the Ruhr University of Bochum, Germany.

He is investigating face detection and recognition systems built upon Gabor wavelets as models of the visual system in humans. He focuses on improving the detection and recognition accuracies by exploiting statistics of faces and their occurrences. Further interests are in the field of applied learning theories such as artificial neural networks and evolutionary algorithms.

Rolf P. Würtz obtained his diploma in mathematics from the University of Heidelberg, Germany in 1986. After that, he was a research assistant at the Max-Planck-Institute for Brain Research in Frankfurt, Germany. In 1990, he joined the Institute for Neural Computation at the University of Bochum, Germany, where he received his Ph.D. from the Physics department in 1994. Until 1997, he was a postdoctoral researcher at the Department of Computing Science at the University of Groningen, The Netherlands. He is currently a faculty member at the Institute for Neural Computation and the International Graduate School for Neuroscience in Bochum.

His research interests include neuronal models and efficient algorithms for object recognition, hand-eye coordination, integration of visual and tactile information, and links to higher cognition. Computational organization problems and solution methods inspired by living systems are studied under the framework of "Organic Computing".