Preprint of an article in press 2015 at
Journal of Wavelets, Multiresolution and Information Processing
http://www.worldscientific.com/worldscinet/ijwmip

RECONSTRUCTION OF IMAGES FROM GABOR GRAPHS WITH APPLICATIONS IN FACIAL IMAGE PROCESSING

Manuel Günther
Idiap Research Institute, Martigny, Switzerland
manuel.guenther@idiap.ch

Stefan Böhringer
Department of Medical Statistics and Bioinformatics,
Leiden University Medical Center, The Netherlands
and Institut für Humangenetik, Universitätsklinikum Essen, Germany
s.boehringer@lumc.nl

Dagmar Wieczorek
Institut für Humangenetik, Universitätsklinikum Essen, Germany
dagmar.wieczorek@uni-due.de

Rolf P. Würtz
Institute for Neural Computation, Ruhr-University Bochum, Germany
rolf.wuertz@ini.rub.de

Graphs labeled with complex-valued Gabor jets are one of the important data formats for face recognition and the classification of facial images into medically relevant classes like genetic syndromes. We here present an interpolation rule and an iterative algorithm for the reconstruction of images from these graphs. This is especially important if graphs have been manipulated for information processing. One such manipulation is averaging the graphs of a single syndrome, another one building a composite face from the features of various individuals. In reconstructions of averaged graphs of genetic syndromes, the patients' identities are suppressed, while the properties of the syndromes are emphasized. These reconstructions from average graphs have a much better quality than averaged images.

Keywords: face recognition; Gabor wavelets; reconstruction; caricature; average face; syndrome classification

AMS Subject Classification: 68T45, 94A08

1. Introduction

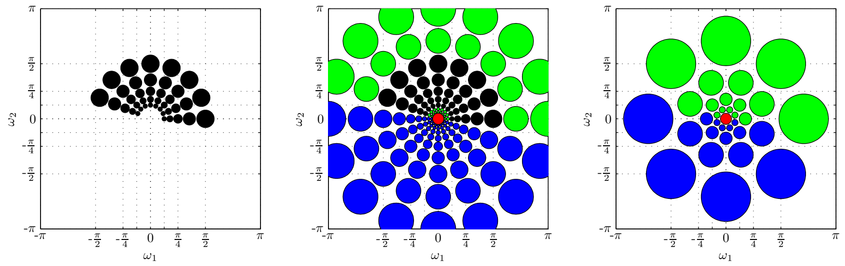
Graphs labeled with Gabor jets – so-called *Gabor graphs* – have proven very useful for face detection and recognition.[16,25] Recently, the data format has been applied successfully to the classification of different syndromes, which influence the facial appearance, from static facial images.[5,10,12,15,18,21,23] To better interpret which kind of features are contained in the Gabor graphs and for inspection by the clinician, it is important to visualize Gabor graphs by reconstructing images from them.

Daugman[6,7] used a neural network to approximate the expansion coefficients of a Gabor wavelet transform in full resolution. Attempts at reconstructing images from sparsely sampled Gabor wavelet responses include the ones by Wundrich[26–28] and Pötzsch.[19,20] Wundrich *et al.*[27] used an iterative algorithm to reconstruct images from densely sampled *amplitudes* of the Gabor wavelet responses. Furthermore, Wundrich[26] presented the framework for our reconstruction algorithm by using dual Gabor wavelets, which are presented and optimized in Section 3.1. On the other hand, Pötzsch[19] reconstructed single complex-valued Gabor jets and removed background information from them. He[20] also established an algorithm for visualizing Gabor graphs by reconstructing the Gabor jets locally and defining Voronoi areas around the nodes. This introduces artifacts like abrupt gray value changes at the borders of Voronoi cells.

In this paper, we extend the iterative reconstruction algorithm proposed by Wundrich[26–28] using complex-valued Gabor wavelet responses at sparse node positions of Gabor graphs. In Section 2, the Gabor wavelet transform is recapitulated and discrete Gabor wavelet families are introduced, which can be used for gray or color image transform and reconstruction. Section 3 presents an iterative reconstruction algorithm, which is able to reconstruct an image from a Gabor graph, and an approximative algorithm that interpolates Gabor wavelet responses from the landmark positions of the graph to yield better initial conditions for the iteration. Finally, Section 4 shows applications of the reconstruction algorithm by reconstructing composite face graphs or face graphs that average over different identities. As an example, face graphs of patients with the same genetic syndrome were averaged. The reconstructed images emphasize the characteristics of the genetic syndrome, while suppressing patient identity.

2. Gabor graphs

A Gabor graph $\mathcal{G} = (\underline{\mathcal{L}}, \underline{\mathcal{E}}, \underline{\mathcal{J}})$ consists of a certain number L of landmarks $\underline{\mathcal{L}}$, which are labeled with Gabor jets $\underline{\mathcal{J}}$ (see Section 2.1 for details) and linked by edges $\underline{\mathcal{E}}$. These graphs were shown to be useful for detection and recognition of objects and faces.[1,16,24,25] For object detection and recognition, usually grid graphs are employed, i. e., the landmarks are arranged in a grid structure and the edges link horizontal or vertical neighboring nodes. For frontal faces, a *face graph* is created. An example of a hand-labeled face graph including the underlying image is shown in Figure 3(a). The landmarks of the face graph are located at significant facial positions like nose tip and inner and outer eye corners. The edges of face graphs link neighboring landmarks and are used for landmark detection, face recognition, or syndrome classification, which are not discussed in this article. The reconstruction algorithm proposed in Section 3 is able to reconstruct images from any graph topology as long as the landmark sampling is dense enough. In this paper, only



(a) Common Gabor wavelet family Γ

(b) Extended Gabor wavelet family $\Gamma^{\{g\}}$ for gray channel

(c) Extended Gabor wavelet family $\Gamma^{\{c\}}$ for color channels

Figure 1. **Discrete families of Gabor wavelets.** *This figure displays the (extended) Gabor wavelet families in frequency domain that are used in this paper. Each circle displays one Gabor wavelet with standard deviation* $\frac{k_\zeta}{\sigma}$*. The black circles in (a) and (b) depict the common Gabor wavelet family. In (b) and (c), the green circles illustrate the additional levels of Gabor wavelets, the red circle marks the Gaussian* $\check{\psi}_0$*, and the blue circles show Gabor wavelets that not required due to symmetry.*

reconstructions from face graphs are shown, an object image reconstructed from a regularly shaped graph using this algorithm is shown elswhere.[9]

2.1. *Gabor jets*

The face graph $\mathcal{G}$ shown in Figure 3(a) is derived from the image $\mathcal{I}$ by the following procedure. First, landmarks are placed onto the face. *Elastic bunch graph matching*[25] (EBGM) is a well-established algorithm to locate landmark positions automatically, and also other detection processes[12,13] have shown up lately. Nonetheless, for most practical applications, the best results are obtained with hand-labeled graphs.

After placing landmarks, the Gabor jets are extracted at the landmark positions. To achieve this, image $\mathcal{I}$ is convolved with Gabor wavelets:[8,16]

$$\psi_{\underline{k}_j}(\underline{x}) = \frac{\underline{k}_j^2}{\sigma^2}\exp\left(-\frac{\underline{k}_j^2\underline{x}^2}{2\sigma^2}\right)\left[\exp\left(i\underline{k}_j^{\mathrm{T}}\underline{x}\right) - \exp\left(-\frac{\sigma^2}{2}\right)\right]. \tag{2.1}$$

The discrete Gabor wavelet family with parameters $\Gamma = (\zeta_{\max}, \nu_{\max}, k_{\max}, k_{\mathrm{fac}}, \sigma)$ has a range of center frequencies $\underline{k}_j$, which are commonly[16,25] calculated as:

$$\underline{k}_j = k_\zeta \begin{pmatrix} \cos(\varphi_\nu) \\ \sin(\varphi_\nu) \end{pmatrix} \tag{2.2}$$

with:

$$k_\zeta = k_{\max} \cdot k_{\mathrm{fac}}^{\zeta} \qquad\qquad \varphi_\nu = \frac{\nu \cdot \pi}{\nu_{\max}}\,, \tag{2.3}$$

where $1 < \nu \leq \nu_{\max}$ defines the rotation of the Gabor wavelet and $1 < \zeta \leq \zeta_{\max}$ its scale level. The index j enumerates all $J = \zeta_{\max} \cdot \nu_{\max}$ wave vector specializations of family Γ.

The commonly used[5,16,25] Gabor wavelet family includes $\zeta_{\max} = 5$ scale levels and $\nu_{\max} = 8$ directions. $k_{\text{fac}} = \frac{1}{\sqrt{2}}$ and $k_{\max} = \frac{\pi}{2}$ are adjusted such that the Gabor wavelet family evenly fills a sub-band in frequency domain, $k_{\max}$ specifies the highest investigated frequency. This family is displayed in Figure 1(a), where each circle depicts a Gabor wavelet in frequency domain:

$$\check{\psi}_{\underline{k}_j}(\underline{\omega}) = \exp\left(-\frac{\sigma^2(\underline{\omega} - \underline{k}_j)^2}{2\underline{k}_j^2}\right) - \exp\left(-\frac{\sigma^2(\underline{\omega}^2 + \underline{k}_j^2)}{2\underline{k}_j^2}\right), \tag{2.4}$$

with its center at the wave vector $\underline{k}_j$. The radius $\frac{\sigma}{k_\zeta}$ of the circle is one effective standard deviation of the Gaussian in frequency domain that is defining the Gabor wavelet, i. e., the first term of Equation (2.4). Finally, the setup of $\sigma = 2\pi$ implies that the wavelength of the Gabor wavelets is equal to one effective standard deviation of the enveloping Gaussian.

The Gabor jet, which encodes the local texture around landmark $\mathcal{L}_l$, is the aggregation of the complex responses of all Gabor wavelets $\psi_{\underline{k}_j}$:

$$(\mathcal{J}_l)_j = (\mathcal{I} * \psi_{\underline{k}_j})(\mathcal{L}_l)\,. \tag{2.5}$$

For detection, recognition, or classification, these complex values are usually used in their polar form: $(\mathcal{J})_j = a_j \cdot \exp(i\phi_j)$, but for reconstruction purpose the algebraic form, i. e., with real and imaginary values is more useful.

The convolution in Equation (2.5) is executed in frequency domain. For this purpose, image $\mathcal{I}$ is transformed using the *fast Fourier transform* (FFT) algorithm. The resulting image $\check{\mathcal{I}}$ has the same number of pixels $\check{\mathcal{I}}(\underline{y})$ as the original image $\mathcal{I}$.[14] To perform convolution, $\check{\mathcal{I}}$ is multiplied with each Gabor wavelet in frequency domain, the results are gathered in the *Gabor transformed image* $\check{\underline{\mathcal{T}}}$ with sub-bands $\check{\mathcal{T}}_{\underline{k}_j}$ in frequency domain:

$$\check{\mathcal{T}}_{\underline{k}_j}(\underline{y}) = \check{\mathcal{I}}(\underline{y}) \cdot \check{\psi}_{\underline{k}_j}(\underline{\omega})\,. \tag{2.6}$$

In Equation (2.6), $\underline{\omega} \in [-\pi, \pi)^2$ covers the 2-dimensional frequency domain, while $\underline{y}$ iterates over the image dimension $(N_1, N_2)^{\mathrm{T}}$. To be able to compute the multiplication pixel-wise, the coordinate transform:

$$\begin{aligned}\omega_1 &= \begin{cases} 2\pi\frac{y_1}{N_1} & \text{if } y_1 < \frac{N_1}{2} \\ 2\pi(\frac{y_1}{N_1} - 1) & \text{if } y_1 \geq \frac{N_1}{2}\end{cases} \\ \omega_2 &= \begin{cases} 2\pi\frac{y_2}{N_2} & \text{if } y_2 < \frac{N_2}{2} \\ 2\pi(\frac{y_2}{N_2} - 1) & \text{if } y_2 \geq \frac{N_2}{2}\end{cases}\end{aligned} \tag{2.7}$$

is applied. This transformation brings the frequency domain center $\underline{\omega}_0 = (0,0)^{\mathrm{T}}$ to the top-left pixel, as required by all common FFT implementations. Scaling is done non-isotropically, in the case of non-square images with $N_1 \neq N_2$, Gabor wavelets in frequency domain $\check{\psi}_{\underline{k}_j}(\underline{y})$ become ellipses in image coordinates, cf. Figure 1. Furthermore, the spatial extends of Gabor wavelets are independent of the current image resolution, see[12] for impacts of this. For the sake of legibility, the pixels $\underline{y}$ in frequency domain are neglected and only the frequencies $\underline{\omega}$ are used.

After creating a Gabor transformed image $\check{\underline{\mathcal{T}}}$ in frequency domain, the *inverse fast Fourier transform* (iFFT) is applied to its sub-bands $\check{\mathcal{T}}_{\underline{k}_j}$, resulting in the Gabor transformed image $\underline{\mathcal{T}}$ in spatial domain. The whole procedure is called the *Gabor wavelet transform* (GWT) from image $\mathcal{I}$ to the Gabor transformed image $\underline{\mathcal{T}}$. Gabor jet information can easily be collected from $\underline{\mathcal{T}}$:

$$(\mathcal{J}_l)_j = \mathcal{T}_{\underline{k}_j}(\mathcal{L}_l) \tag{2.8}$$

by converting the horizontal structure of $\underline{\mathcal{T}}$ to the vertical structure of $\mathcal{J}$.

2.2. *Extended Gabor jets*

Since the Gabor wavelet family Γ spans only a sub-band in frequency domain (see Figure 1(a)), only this information can be reconstructed. Especially, the low frequency information and the mean gray value at $\check{\mathcal{I}}(\underline{\omega}_0)$ are not included in the Gabor jets.

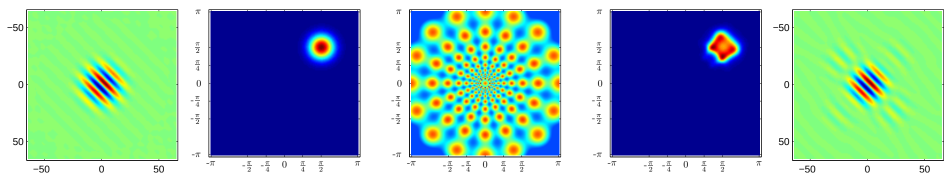
To be able to reconstruct full gray level or colored images, the missing information has to be embedded, too. This is achieved with the *extended discrete Gabor wavelet family.* $\Gamma^{\{g\}} = (\zeta^{\{g\}}_{\max}, \nu_{\max}, k^{\{g\}}_{\max}, k_{\text{fac}}, \sigma, \sigma_0)$ with $\zeta^{\{g\}}_{\max} = 9$ scales of Gabor wavelets, starting with the highest frequency $k^{\{g\}}_{\max} = \pi$. To cover the mean value and the lowest frequencies, a Gaussian:

$$\check{\psi}_0(\underline{\omega}) = \exp\left(-\frac{\sigma_0^2 \cdot \underline{\omega}^2}{2}\right) \tag{2.9}$$

with standard deviation $\sigma_0 = 2\pi$ centered at $\underline{\omega} = \underline{\omega}_0$ is added. Figure 1(b) displays the Gabor wavelets $\check{\psi}_{\underline{k}_j}$ and the Gaussian $\check{\psi}_0$, which are used to create extended Gabor jets $\mathcal{J}^{\{g\}}$ with full gray value information. The black circles are equal to Figure 1(a), the green circles depict the newly included Gabor wavelet scale levels, and the red circle displays the Gaussian. The second half of the frequency domain, which is filled with blue circles in Figure 1(b), is not needed for the Gabor jet creation since the responses can easily be calculated from the available ones (see Section 3).

Including color information is straightforward. The image has to be color transformed into YUV color space with the gray channel Y and two color channels U and V. The color channels are adjusted such that the neutral color, i. e., gray is centered at U- and V-value zero. Afterwards, the Y channel is handled by the family $\Gamma^{\{g\}}$ as shown in Figure 1(b), whereas the U and V channels are successively processed with the *discrete color Gabor wavelet family* $\Gamma^{\{c\}} = (\zeta^{\{c\}}_{\max}, \nu^{\{c\}}_{\max}, k^{\{c\}}_{\max}, k^{\{c\}}_{\text{fac}}, \sigma, \sigma_0)$. This family, which is shown in Figure 1(c), incorporates a sparser discretization of Gabor wavelets and a Gaussian, the parameters are:

$$\zeta^{\{c\}}_{\max} = 4 \qquad \nu^{\{c\}}_{\max} = 4 \qquad k^{\{c\}}_{\max} = \frac{\pi}{\sqrt{2}} \qquad k^{\{c\}}_{\text{fac}} = \frac{1}{2}\,. \tag{2.10}$$



(a) $\psi_{\underline{k}_j}(\underline{x})$ (b) $\check{\psi}_{\underline{k}_j}(\underline{\omega})$ (c) $\check{C}(\underline{\omega})$ of $\Gamma^{\{g\}}$ (d) $\check{\psi}^d_{\underline{k}_j}(\underline{\omega})$ (e) $\psi^d_{\underline{k}_j}(\underline{x})$

Figure 2. **Gabor wavelets and dual wavelets.** *This figure displays Gabor wavelets and their dual wavelets. In (a) and (e), the imaginary parts of $\psi_{\underline{k}_j}(\underline{x})$ and $\psi^d_{\underline{k}_j}(\underline{x})$ with $\underline{k}_j = (\frac{\pi}{8}, \frac{\pi}{8})^T$ are shown, while (b) and (d) present $\check{\psi}_{\underline{k}_j}(\underline{\omega})$ and $\check{\psi}^d_{\underline{k}_j}(\underline{\omega})$ for $\underline{k}_j = (\frac{\pi}{2}, \frac{\pi}{2})^T$. The denominator $\check{C}(\underline{\omega})$ for the dual wavelets is displayed in (c).*

Every other scale level is rotated by half a direction gap. Altogether, there are $J^{\{g\}} = 9 \cdot 8 + 1$ Gabor wavelets for the Y channel and each $4 \cdot 4 + 1$ Gabor wavelets for the U and V channels totaling to $J^{\{c\}} = 107$ complex-valued responses in a color Gabor jet $\mathcal{J}^{\{c\}}$.

3. Iterative Reconstruction

The reconstruction of an image $\mathcal{I}'$ from the extended Gabor graph $\mathcal{G}$ can be seen as an extension of the *inverse Gabor wavelet transform* (iGWT). Its calculation is made difficult by the following three problems.

(1) **Missing half of frequency space:** The reconstructed image $\mathcal{I}'$ in spatial domain is not real since only half of the frequency domain is covered by Gabor wavelets, i. e., the blue circles of Figures 1(b) and 1(c) are missing.
(2) **Low sampling in frequency space:** Gabor wavelets do not form a basis of the function space, because they are overcomplete. Such function systems are called *frames*.[17] In frequency space, the convolution with the wavelets is a multiplication. Therefore, image frequencies for which *all* wavelets have values close to zero (outside the black circles in Figures 1(b) and 1(c)) can not be recovered. This is not a problem if all relevant frequencies are well-covered with wavelets (green and black circles). Nevertheless, different frequencies have different weights (highest for the center frequencies $\underline{k}_j$. To account for this, reconstruction is best performed using *dual wavelets* (see below).
(3) **Low sampling in image space:** Most parts of the Gabor transformed image $\check{\underline{\mathcal{T}}}$ are undefined since information is only available at the node positions of $\mathcal{G}$.

```
Input: Gabor graph $\mathcal{G}$      Output: reconstructed image $\mathcal{I}'$
(1) initialize:
    (a) Reset all values of $\underline{\mathcal{T}}$: $\forall j, \forall \underline{x} : \mathcal{T}'_{\underline{k}_j}(\underline{x}) = 0 + 0i$
(2) iterate:
    (a) Fill Gabor graph into Gabor transformed image: $\mathcal{T}'_{\underline{k}_j}(\mathcal{L}_l) = (\mathcal{J}_l)_j$
    (b) Fourier transform $\underline{\mathcal{T}}'$ to $\check{\underline{\mathcal{T}}}'$
    (c) Reconstruct $\check{\underline{\mathcal{T}}}'$ to $\check{\mathcal{I}}'$
    (d) If number of iterations is reached: goto step 3
    (e) Forward Gabor transform: $\check{\mathcal{I}}' \Rightarrow \check{\underline{\mathcal{T}}}$
    (f) Inverse Fourier transform $\check{\underline{\mathcal{T}}}$ to $\underline{\mathcal{T}}$
(3) finalize:
    (a) Inverse Fourier transform $\check{\mathcal{I}}'$ to $\mathcal{I}'$
    (b) Correct pixel range to fit into, e. g., $[0, 255]$
```

Algorithm 1: **Iterative reconstruction.** *This iterative algorithm reconstructs the image $\mathcal{I}'$ from the Gabor graph $\mathcal{G}$.*

The first problem is solved by exploiting the symmetries of images, Gabor wavelets and Gabor transformed images:

$$\begin{aligned} \mathcal{I}(\underline{x}) &= \overline{\mathcal{I}(\underline{x})} & \check{\mathcal{I}}(\text{-}\underline{\omega}) &= \overline{\check{\mathcal{I}}(\underline{\omega})} \\ \psi_{\text{-}\underline{k}_j}(\underline{x}) &= \overline{\psi_{\underline{k}_j}(\underline{x})} & \check{\psi}_{\text{-}\underline{k}_j}(\underline{\omega}) &= \check{\psi}_{\underline{k}_j}(\text{-}\underline{\omega}) \\ \mathcal{T}_{\text{-}\underline{k}_j}(\underline{x}) &= \overline{\mathcal{T}_{\underline{k}_j}(\underline{x})} & \check{\mathcal{T}}_{\text{-}\underline{k}_j}(\underline{\omega}) &= \overline{\check{\mathcal{T}}_{\underline{k}_j}(\text{-}\underline{\omega})}\,. \end{aligned} \tag{3.1}$$

In Equation (3.1), $\overline{\cdot}$ refers to complex conjugation, and the term $\check{\psi}_{\text{-}\underline{k}_j}$ stands for Gabor wavelets $\check{\psi}_{\underline{k}_j}$ point reflected at $\underline{\omega}_0$, which are depicted by the blue circles in Figure 1. Similarly, $\mathcal{T}_{\text{-}\underline{k}_j}$ are the inexistent sub-bands of Gabor transformed image corresponding to the Gabor wavelets $\check{\psi}_{\text{-}\underline{k}_j}$.

3.1. *Dual wavelets*

The second problem is addressed by using *dual Gabor wavelets* $\psi^d_{\underline{k}_j}$ and a *dual Gaussian* ψ^d_0. The duals are reconstructing functions for the discretized wavelet transform and correct the different weighting of different image frequencies. In the limit of infinitely fine resolution in frequency space the dual wavelets converge to a multiple of the wavelets themselves. The inverse transform is obtained by multiplying the transformed image with the duals and adding up over all $\underline{k}_j$ and their counterparts in the other half of frequency domain:

$$\begin{aligned}\check{\mathcal{I}}'(\underline{\omega}) = \sum_j &\Big[\check{\mathcal{T}}_{\underline{k}_j}(\underline{\omega}) \cdot \check{\psi}^d_{\underline{k}_j}(\underline{\omega}) + \check{\mathcal{T}}_{\text{-}\underline{k}_j}(\underline{\omega}) \cdot \check{\psi}^d_{\text{-}\underline{k}_j}(\underline{\omega})\Big] \\ &+ \check{\mathcal{T}}_0(\underline{\omega}) \cdot \check{\psi}^d_0(\underline{\omega})\,. \end{aligned} \tag{3.2}$$

The duals themselves are calculated in frequency domain as:

$$\check{\psi}^d_{\underline{k}_j}(\underline{\omega}) = \frac{\check{\psi}_{\underline{k}_j}(\underline{\omega})}{\check{C}(\underline{\omega})} \qquad\qquad \check{\psi}^d_0(\underline{\omega}) = \frac{\check{\psi}_0(\underline{\omega})}{\check{C}(\underline{\omega})} \tag{3.3}$$

with:

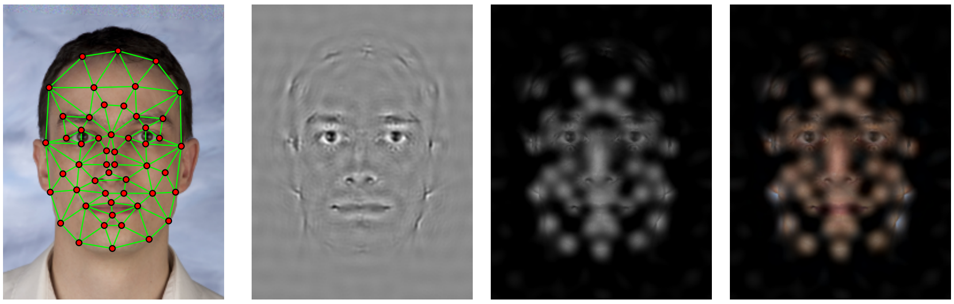
$$\check{C}(\underline{\omega}) = \max\left\{\check{C}_{\min}, \check{C}_{\psi}(\underline{\omega})\right\} \tag{3.4}$$

$$\check{C}_{\psi}(\underline{\omega}) = \sum_j \left[\check{\psi}_{\underline{k}_j}(\underline{\omega})^2 + \check{\psi}_{\text{-}\underline{k}_j}(\underline{\omega})^2\right] + \check{\psi}_0(\underline{\omega})^2\,. \tag{3.5}$$

The calculation of the denominator $\check{C}_{\psi}$[2,22] in Equation (3.5) can be deduced from wavelet theory.[12] The $\check{C}_{\min}$ parameter, which confines $\check{C}$ not to vanish, has to be lower than the sum of squared Gabor wavelets in frequency domain, when $\underline{\omega}$ is inside of the covered sub-band, but should not be too small. We choose $\check{C}_{\min} = 0.25$. If the family does not include a Gaussian ψ_0, e. g., as in Gabor wavelet family Γ shown in Figure 1(a), it can simply be left out of Equation (3.5).

Two pairs of Gabor wavelets and their corresponding dual Gabor wavelets are shown in Figure 2. Figures 2(a) and 2(e) present the imaginary parts of $\psi_{\underline{k}_j}$ and $\psi^d_{\underline{k}_j}$ in spatial domain, employing $\underline{k}_j = (\frac{\pi}{8}, \frac{\pi}{8})^{\mathrm{T}}$, while Figures 2(b) and 2(d) display $\check{\psi}_{\underline{k}_j}$ and $\check{\psi}^d_{\underline{k}_j}$ in frequency domain for $\underline{k}_j = (\frac{\pi}{2}, \frac{\pi}{2})^{\mathrm{T}}$. Finally, Figure 2(c) visualizes the denominator $C(\underline{\omega})$ for the extended Gabor wavelet family $\Gamma^{(g)}$. For the *continuous* Gabor wavelet family with infinite resolution in $\underline{k}_j$ and without the Gaussian ψ_0, the denominator $\check{C}_{\psi}(\underline{\omega})$ converges to the *admissibility constant* C_{ψ}.[12,17]

We now show that these functions yield a perfect reconstruction of those image frequencies for which $\check{C}(\underline{\omega}) > \check{C}_{\min}$ by inserting Equation (2.6) and Equation (3.3)



(a) Hand-labeled Face Graph

(b) Reconstruction with Γ

(c) Reconstruction with $\Gamma^{\{g\}}$

(d) Reconstruction with $\Gamma^{\{c\}}$

Figure 3. **Iterative reconstructions.** *This figure displays the iterative image reconstruction of the face graph shown in (a) after 200 iterations. (b) presents the result for Gabor wavelet family* Γ*, the gray values are auto-scaled between the lowest (black) and the highest (white) values. For (c) and (d), Gabor wavelet families* $\Gamma^{\{g\}}$ *and* $\Gamma^{\{c\}}$ *are employed and the resulting pixel values are cut into* $[0, 255]$.

into Equation (3.2):

$$\begin{aligned}
\check{\mathcal{I}}'(\underline{\omega}) &= \sum_j \left[\check{\mathcal{T}}_{\underline{k}_j}(\underline{\omega}) \cdot \check{\psi}^d_{\underline{k}_j}(\underline{\omega}) + \check{\mathcal{T}}_{\text{-}\underline{k}_j}(\underline{\omega}) \cdot \check{\psi}^d_{\text{-}\underline{k}_j}(\underline{\omega})\right] \\
&\quad + \check{\mathcal{T}}_0(\underline{\omega}) \cdot \check{\psi}^d_0(\underline{\omega}) \\
&= \sum_j \left[\check{\psi}_{\underline{k}_j}(\underline{\omega}) \cdot \check{\mathcal{I}}(\underline{\omega}) \cdot \check{\psi}^d_{\underline{k}_j}(\underline{\omega}) + \check{\psi}_{\text{-}\underline{k}_j}(\underline{\omega}) \cdot \check{\mathcal{I}}(\underline{\omega}) \cdot \check{\psi}^d_{\text{-}\underline{k}_j}(\underline{\omega})\right] \\
&\quad + \check{\psi}_0(\underline{\omega}) \cdot \check{\mathcal{I}}(\underline{\omega}) \cdot \check{\psi}^d_0(\underline{\omega}) \\
&= \check{\mathcal{I}}(\underline{\omega}) \left(\sum_j \left[\check{\psi}_{\underline{k}_j}(\underline{\omega}) \, \frac{\check{\psi}_{\underline{k}_j}(\underline{\omega})}{\check{C}_\psi(\underline{\omega})} + \check{\psi}_{\text{-}\underline{k}_j}(\underline{\omega}) \, \frac{\check{\psi}_{\text{-}\underline{k}_j}(\underline{\omega})}{\check{C}_\psi(\underline{\omega})}\right] \right. \\
&\quad \left. + \check{\psi}_0(\underline{\omega}) \, \frac{\check{\psi}_0(\underline{\omega})}{\check{C}_\psi(\underline{\omega})} \right) \\
&= \check{\mathcal{I}}(\underline{\omega}) \, \frac{\sum_j \left[\check{\psi}_{\underline{k}_j}(\underline{\omega})^2 + \check{\psi}_{\text{-}\underline{k}_j}(\underline{\omega})^2\right] + \check{\psi}_0(\underline{\omega})^2}{\check{C}_\psi(\underline{\omega})} \, . \qquad (3.6)
\end{aligned}$$

By comparison with Equation (3.5) the remaining fraction in Equation (3.6) is equal to one at all frequencies where $\check{C}(\underline{\omega})$ is larger than $\check{C}_{\min}$. All other frequencies, which are not covered well by the Gabor wavelet family, are more or less lost in the transform and, consequently, suppressed in the reconstruction.

3.2. *Iteration*

To address the third issue, we propose an iterative algorithm that slowly fills the Gabor transformed image $\underline{\mathcal{T}}$ with information from Gabor graph $\mathcal{G}$. This algorithm can be launched with any of the presented Gabor wavelet families and corresponding dual wavelets. The iterative reconstruction algorithm, which is in detail depicted in Algorithm 1, starts with a blank Gabor transformed image $\underline{\mathcal{T}}$ (cf. step 1a) that has J, $J^{\{g\}}$, or $J^{\{c\}}$ number of sub-bands.

The iteration is composed of six steps. First, the information contained in $\mathcal{G}$ is inserted into $\underline{\mathcal{T}}'$:

$$\mathcal{T}'_{\underline{k}_j}(\mathcal{L}_l) = (\mathcal{J}_l)_j \tag{3.7}$$

by pasting the complex-valued Gabor jet information $(\mathcal{J}_l)_j$ to the Gabor transformed image at the landmark positions $\mathcal{L}_l$ and leaving the remaining pixels unchanged. Successively, an FFT is applied to the Gabor transformed image (step 2b). In step 2c, the image $\check{\mathcal{I}}'$ is reconstructed from $\check{\underline{\mathcal{T}}}'$ using Equation (3.2). When continuing the iteration with step 2e, the resulting image $\check{\mathcal{I}}'$ is forward Gabor transformed (cf. Equation (2.6)). To be able to fill information of $\mathcal{G}$ into the $\underline{\mathcal{T}}$ during the next iteration, it needs to be in spatial domain and, thus, an iFFT from $\check{\underline{\mathcal{T}}}$ to $\underline{\mathcal{T}}$ needs to be performed as the last step of the iteration.

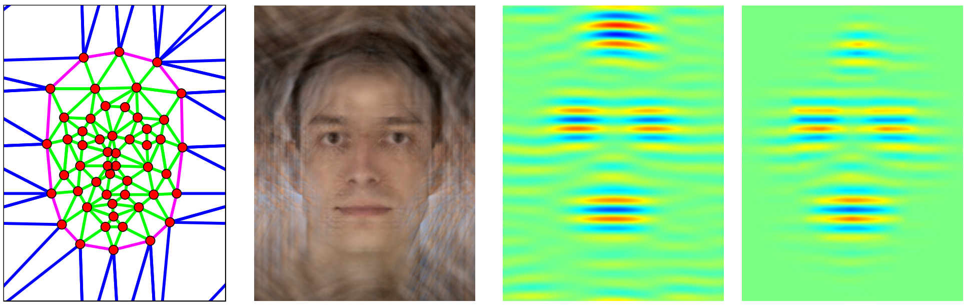
The number of iterations can be varied, but at least 100 iterations are needed for a reasonable result. Especially, if the spatial distribution of the landmarks is unbalanced, the number of iterations need to increase. After the last iteration, the image $\check{\mathcal{I}}'$ from step 2c is inverse Fourier transformed to spatial domain, and the pixel values are adjusted to fit into the pixel range of usual gray level or color images. When Gabor wavelet family Γ was employed or only few iterations were performed, this adjustment is done by linearly scaling pixel gray values, i. e., the lowest (usually negative) values become black, while the highest values turn white. For $\Gamma^{\{g\}}$ and $\Gamma^{\{c\}}$ and sufficient iterations, the gray or color values are simply cut to $[0, 255]$.

3.3. *Convergence of the iterative algorithm*

In order to understand why the described algorithm always converges to a solution we regard the functions from the Cartesian product of the image domain and the spatial frequency domain into the complex numbers as functions into a pair of real numbers. As both domains are discrete, these form a finite dimensional real vector space of dimension $2 \times N_1 \times N_2 \times \zeta_{\max} \times \nu_{\max}$, which we will call $\mathcal{W}$. The algorithm as described above and summarized in algorithm Algorithm 1 operates on elements of this vector space.

The operation in step 2a replaces any given element $\mathcal{T} \in \mathcal{W}$ to another one with the same components except the ones at the graph positions, where the components are set to those of the graph. The result is in

$$\mathcal{W}_{\mathcal{G}} = \left\{ \mathcal{T} \,|\, \mathcal{T}'_{\underline{k}_j}(\mathcal{L}_l) = (\mathcal{J}_l)_j \right\} .$$



(a) Triangulation

(b) Linear interpolation

(c) Original sub-band $\mathcal{T}_{\underline{k}_j}$

(d) Approximated sub-band $\mathcal{T}^*_{\underline{k}_j}$

Figure 4. **Interpolation of the Gabor transformed image.** *This figure displays the approximation process of the face graph shown in Figure 3(a). In (a), the triangulation result including the triangles that cross the image borders are shown. The reconstructed color image with full linear interpolation is shown in (b). In (c) and (d), the* $\underline{k}_j = (0, \frac{\pi}{8})^T$*-sub-band of the original and approximated Gabor transformed image (with limited weights) is visualized.*

$\mathcal{W}_\mathcal{G}$ is an affine subspace of $\mathcal{W}$. The operation itself is an orthogonal projection onto that subspace.

The operations 2b, 2c, 2e, and 2f are all linear. Their composition projects $\mathcal{T}$ orthogonally onto the space $\mathcal{W}_\mathcal{I}$ of all wavelet transforms of *some* image $\mathcal{I}$. This is a linear subspace of $\mathcal{W}$ of dimension at most $2 \times N_1 \times N_2$.

Therefore, the algorithm consists of alternating orthogonal projections onto affine subspaces. It is guaranteed to converge to a point on the intersection of the two spaces if the intersection is non-empty. If it is empty the procedure will end up alternatating between a pair of points on either set with minimal distance. In that case the result will be the inverse wavelet transform of the point on $\mathcal{W}_I$.

3.4. *Initial conditions from interpolation*

The iterative reconstruction algorithm has some drawbacks. The time needed for the iterations is rather high (see Section 3.6 for a detailed analysis), and pixels further away from the landmark positions keep nearly unchanged. This is an issue especially for reconstructions with $\Gamma^{\{g\}}$ or $\Gamma^{\{c\}}$, the reconstructed images looks blistered. Although the result of the iterative reconstruction using the common Gabor wavelet family Γ, an example of which is displayed in Figure 3(b), is recognizable quite well, the gray or color image reconstructions shown in Figures 3(c) and 3(d), respectively, are poor.

These problems can be partially solved by using better initial conditions for the iterative reconstruction algorithm, i. e., by interpolating the missing pixel positions of $\underline{\mathcal{T}}^*$ from the given graph $\mathcal{G}$.

3.4.1. *Linear weights*

The approximation of the Gabor transformed image $\underline{\mathcal{T}}^*$ is done pixel by pixel, $\underline{t} = (t_1, t_2)^{\mathrm{T}}$ names the currently investigated position. To find the landmarks surrounding $\underline{t}$ we use a Delaunay triangulation algorithm that tessellates the interior of the convex hull of the graphs nodes. We adjust this algorithm to triangulate also the exterior of the convex hull by transcending the borders of the image. The triangulation of the graph shown in Figure 3(a) can be found in Figure 4(a), where the red dots are the landmarks, the green lines show the interior triangulation, the purple lines indicate the convex hull, and the blue lines display the exterior triangulation result.

For each point $\underline{t}$, the tessellation defines three surrounding landmarks, say $\mathcal{L}_A = (A_1, A_2)$, $\mathcal{L}_B = (B_1, B_2)$, and $\mathcal{L}_C = (C_1, C_2)$. The weights w_A, w_B, and w_C for these landmarks are chosen to fulfill the linear equation:

$$\underline{t} = w_A \cdot \mathcal{L}_A + w_B \cdot \mathcal{L}_B + w_C \cdot \mathcal{L}_C \,. \tag{3.8}$$

Since the weights sum up to unity and the landmarks are not collinear, this equation has the unique solution:

$$\begin{pmatrix} w_A \\ w_B \\ w_C \end{pmatrix} = \begin{pmatrix} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \\ 1 & 1 & 1 \end{pmatrix}^{-1} \begin{pmatrix} t_1 \\ t_2 \\ 1 \end{pmatrix} . \tag{3.9}$$

The three computed weights are used to interpolate the value of the Gabor transformed image $\underline{\mathcal{T}}^*(\underline{t})$ from the Gabor jets $\mathcal{J}_A$, $\mathcal{J}_B$, and $\mathcal{J}_C$ at the three surrounding landmarks $\mathcal{L}_A$, $\mathcal{L}_B$, and $\mathcal{L}_C$, respectively. For the Gaussian sub-band (if existent), the approximated value is just the weighted average of the Gaussian responses in the Gabor jets:

$$\mathcal{T}_0(\underline{t}) = w_A \cdot (\mathcal{J}_A)_0 + w_B \cdot (\mathcal{J}_B)_0 + w_C \cdot (\mathcal{J}_C)_0 \,. \tag{3.10}$$

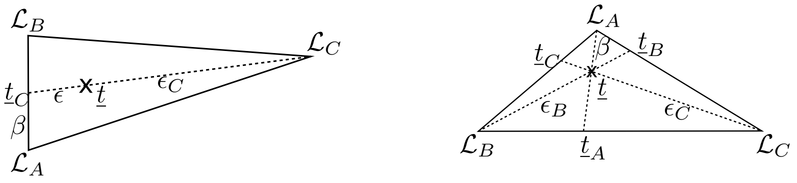
3.4.2. *Phase shift correction*

As the Gabor phase changes with location the responses of the Gabor wavelets need to be corrected in phase before being weighted. In a first approximation, the phase shift is proportional to the frequency $\underline{k}_j$ and to the distance $\underline{d}_X = \underline{t} - \mathcal{L}_X$ from $\underline{t}$ to landmark $\mathcal{L}_X$ (for $X \in \{A, B, C\}$):

$$(\mathcal{J}_X^*)_j = (\mathcal{J}_X)_j \cdot \exp\Big(i\, \underline{k}_j^{\mathrm{T}} \underline{d}_X\Big) \,. \tag{3.11}$$

This yields the following formula for the approximated value for the Gabor wavelet response at $\underline{t}$:

$$\mathcal{T}_{\underline{k}_j}^*(\underline{t}) = w_A \cdot (\mathcal{J}_A^*)_j + w_B \cdot (\mathcal{J}_B^*)_j + w_C \cdot (\mathcal{J}_C^*)_j \,. \tag{3.12}$$



(a) Triangle with one short edge $\mathcal{L}_A\mathcal{L}_B$

(b) Triangle with one long edge $\mathcal{L}_B\mathcal{L}_C$

Figure 5. **Limited linear interpolation.** *This figure shows points and factors used for the mixed linear/nonlinear interpolation model. The triangle in (a) has one linearly (short) and two nonlinearly (long) interpolated edges, whereas the triangle in (b) has two short and one long edge.*

3.4.3. *Limited linear weights*

A full linear interpolation of the responses of all Gabor wavelets may lead to wavy high frequency structures, especially in areas with sparsely placed nodes, but much high frequency information. One example of these waves can be seen in Figure 4(b), where specifically the background area is affected. To tackle this issue, the range of the linear interpolation is limited by calculating weights $w_{X;j}$ for each Gabor wavelet response[a] independently. The limiting distance is defined by the Gaussian envelope of the wavelet without the prefactor:

$$G_{\underline{k}_j}(\underline{x}) = \exp\left(-\frac{\underline{k}_j^2\underline{x}^2}{2\sigma^2}\right). \tag{3.13}$$

The corresponding weights $w_{A;j}$, $w_{B;j}$, and $w_{C;j}$ are assigned to:

$$w_{X;j} = G_{\underline{k}_j}(\mathcal{L}_X - \underline{t}) \qquad\qquad X \in \{A, B, C\}. \tag{3.14}$$

Since there need to be weights for three points $\mathcal{L}_A$, $\mathcal{L}_B$, and $\mathcal{L}_C$, there are four different possibilities:

(1) all three distances are short enough,
(2) all three distances are too long,
(3) one distance is short enough, but two are not,
(4) two distances are short enough, but one is not,

where "short enough" means that $G_{\underline{k}_j}(\mathcal{L} - \mathcal{L}')$ is below a threshold (here 10^{-4}). For possibility 1, the linear weights from Equation (3.9) are used, while possibility 2 uses weights as given in Equation (3.14).

The remaining possibilities 3 and 4 need to be handled by a mixture model that includes both linear and nonlinear weights. Figure 5(a) presents the calculation of the weights for case 3, the short edge is $\mathcal{L}_A\mathcal{L}_B$. The intersection $\underline{t}_C$ of this edge with

[a] It is sufficient to calculate the weights for each scale level of Gabor wavelets, but for legibility this differentiation is avoided here.

the ray from $\mathcal{L}_C$ through current position $\underline{t}$ is calculated and the linear factor β:

$$\beta = \frac{|\mathcal{L}_A - \underline{t}_C|}{|\mathcal{L}_B - \mathcal{L}_A|} \tag{3.15}$$

as well as the nonlinear weights ϵ and ϵ_C:

$$\epsilon = G_{\underline{k}_j}(\underline{t} - \underline{t}_C) \qquad\qquad \epsilon_C = G_{\underline{k}_j}(\mathcal{L}_C - \underline{t}) \tag{3.16}$$

are calculated and used to compute the final weights for the nodes:

$$\begin{aligned} w_{A;j} &= (1-\beta)\cdot\epsilon_0\cdot(1-\epsilon_C) \\ w_{B;j} &= \beta\cdot\epsilon_0\cdot(1-\epsilon_C) \\ w_{C;j} &= (1-\epsilon_0)\cdot(1-\epsilon_C)\,. \end{aligned} \tag{3.17}$$

The weights for case 4, which is depicted in Figure 5(b), require more effort. First, from each corner a ray is send through the current position $\underline{t}$ and the intersection point with the opposite sides are calculated. Assuming $\mathcal{L}_B\mathcal{L}_C$ to be the long edge, the linear weight β is calculated:

$$\beta = \max\left\{\frac{\delta_A}{\delta_A+\delta_B}, \frac{\delta_A}{\delta_A+\delta_C}\right\} \tag{3.18}$$

with:

$$\delta_X = \frac{|\underline{t}_X - \underline{t}|}{|\underline{t}_X - \mathcal{L}_X|} \qquad\qquad X \in \{A,B,C\}\,. \tag{3.19}$$

Then, the nonlinear weights ϵ_B and ϵ_C are computed as:

$$\begin{aligned} \epsilon_B &= G_{\underline{k}_j}(\mathcal{L}_B - \underline{t}_A)\cdot(1-\beta) \\ \epsilon_C &= G_{\underline{k}_j}(\mathcal{L}_C - \underline{t}_A)\cdot(1-\beta) \end{aligned} \tag{3.20}$$

The weights for the nodes are:

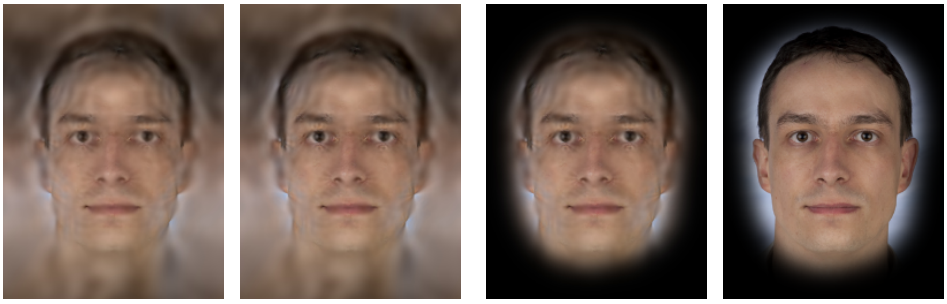
$$\begin{aligned} w_{A;j} &= w_A\cdot\beta \\ w_{B;j} &= w_B\cdot\beta + \epsilon_B\cdot(1-\epsilon_C) \\ w_{C;j} &= w_C\cdot\beta + \epsilon_C\cdot(1-\epsilon_B)\,, \end{aligned} \tag{3.21}$$

using the linear weights w_A, w_B, and w_C from Equation (3.9).

Finally, the interpolation of the Gabor transformed image from Equation (3.12) is rewritten:

$$\mathcal{T}^*_{\underline{k}_j}(\underline{t}) = w_{A;j}\cdot(\mathcal{J}^*_A)_j + w_{B;j}\cdot(\mathcal{J}^*_B)_j + w_{C;j}\cdot(\mathcal{J}^*_C)_j \tag{3.22}$$

using the sub-band dependent weights. The interpolation of the Gaussian sub-band of the Gabor transformed image $\mathcal{T}^*_0$ – if existent – is always linear, i. e., $w_{X;0} = w_X$ for $X \in \{A,B,C\}$. Figures 4(c) and 4(d) show an original sub-band $\mathcal{T}_{\underline{k}_j}$, i. e., the result of the Gabor wavelet transform with Gabor wavelet $\psi_{\underline{k}_j}$, and the approximated sub-band $\mathcal{T}^*_{\underline{k}_j}$ generated by this algorithm, both using $\underline{k}_j = (0, \frac{\pi}{8})^{\mathrm{T}}$. The responses at the border of Figure 4(c) need not be estimated properly since the nodes of the



(a) Limited Interpolation

(b) After 200 iterations

(c) Masked Reconstruction

(d) Masked Original Image

Figure 6. **Reconstruction.** *This figure presents the final results of reconstruction from the graph shown in Figure 3(a). In (a) and (b), reconstructions of the approximated Gabor transformed image after 0 and 200 iterations are shown. The result from (b) is masked with the background mask M, the result is displayed in (c). For direct comparison, the original image masked with the same mask is added in (d).*

graph are too far away from this background region and, thus, the Gabor jets do not contain that information.

One example of the reconstruction of the approximated Gabor transformed image can be obtained from Figure 6(a). It is nicely visible that, in comparison to the fully linear interpolated image shown in Figure 4(b), the high frequency waves are gone, while the overall face structure was kept stable. The reconstructed approximated Gabor transformed image $\underline{\mathcal{T}}^*$ can also be enhanced by performing some more iterations as described in Section 3.2. Figure 6(b) shows the reconstruction result similar to Figure 6(a), but with 200 additional iterations. The low frequency information is stable during the iterations, but the sharpness, which is induced by the high frequencies, increases.

3.5. *Background removal*

The final step of image reconstruction is background removal, a small cosmetic correction to the image. It applies a real-valued mask M to the pixel values of $\mathcal{I}'$:

$$\mathcal{I}'_M(\underline{x}) = \mathcal{I}'(\underline{x}) \cdot M(\underline{x})\,. \tag{3.23}$$

In case of a colored image, the mask is applied to all channels of the RGB color space independently. The mask M is defined by the landmarks of the convex hull of the graph, which are calculated by the triangulation algorithm in Section 3.4. To create mask M for $\mathcal{G}$, a closed cubic B-Spline S is fitted to the convex hull nodes of $\mathcal{G}$, resulting in a smooth curve. The values of the mask are then computed as:

$$M(\underline{x}) = \begin{cases} 1 & \text{if } \underline{x} \text{ is inside } S \\ \exp\left(-\frac{d(\underline{x},S)^2}{32\pi^2}\right) & \text{else} \end{cases}\,, \tag{3.24}$$

Figure	overall	approx.	MKL	LAPACK
3(b)	11.8 s	—	9.2 s	78.2 s
3(c)	21.9 s	—	16.5 s	144.7 s
3(d)	33.6 s	—	24.1 s	215.3 s
4(b)	6.4 s	4.2 s	0.1 s	0.6 s
6(a)	9.5 s	7.2 s	0.1 s	0.6 s
6(b)	40.6 s	7.2 s	23.9 s	215.0 s

Table 1. **Time Constraints.** *This table contains the run times that were required to reconstruct the images shown in Figures 4, 3, and 6. Fast Fourier transforms were executed with the Intel® Math Kernel Library (MKL). For comparison, the fast Fourier transform times needed by the Linear Algebra PACKage (LAPACK) are given in a separate column.*

where $d(\underline{x}, S)$ is the shortest Euclidean distance between point $\underline{x}$ and the Spline S. The Gaussian decay factor of $32\pi^2 = 2 \cdot (2\sigma_0)^2$ was chosen by hand and could be altered to include more or less of the background.

The final result of the reconstruction can be inspected in Figure 6(c), which is the masked result from Figure 6(b). For direct comparison, the original image masked with the same mask M is given in Figure 6(d).

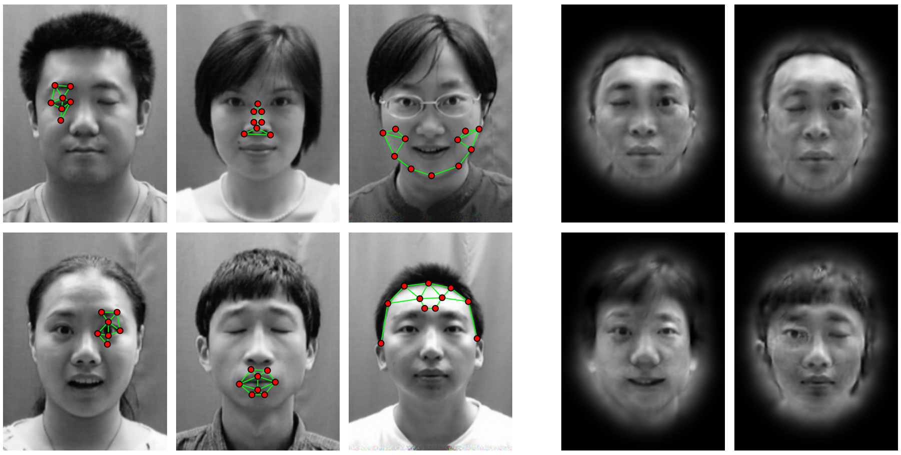
3.6. *Time Constraints*

Two important aspects of the reconstruction algorithm are the run time and the memory requirement. Of course, both are highly dependent on the resolution $N_1 \times N_2$ of the image to reconstruct and on the number of Gabor wavelets employed.

The required memory can be estimated by multiplying the image resolution $N = N_1 \cdot N_2$ with the number of sub-bands of $\mathcal{T}$, resulting in the number of complex-valued pixels of the Gabor transformed image $\underline{\mathcal{T}}$. Hence, relatively small color images of size 360×480 pixel need $N \cdot J^{\{c\}} \cdot 16$ Byte ≈ 308 MB of memory, only for the Gabor transformed image. Since $\underline{\mathcal{T}}^*$ is approximated in-place, the memory consumption is nearly identical for both initial conditions of the iteration, i. e., empty $\underline{\mathcal{T}}$ (cf. Section 3.2) or approximated $\underline{\mathcal{T}}^*$ as introduced in Section 3.4.

For practical applications, the duration of a single graph reconstruction should not take longer than a couple of seconds. Table 1 shows detailed measured times of the reconstruction algorithm reconstructing the images shown in Figures 4, 3, and 6. All experiments were executed on a Dell Precision 670 with a 3200 MHz Intel Xeon (64 bit) dual-core processor. For FFT and iFFT calculations, the commercial Intel® *Math Kernel Library* (MKL) and the open-source *Linear Algebra PACKage* (LAPACK) were employed. The resolution of the images is 192×256 pixel.

The iterative reconstruction schema as given in Algorithm 1 needs to execute an FFT and an iFFT of the whole transformed image $\underline{\mathcal{T}}$, i. e., one FFT and one



(a) Facial parts

(b) Composite faces

Figure 7. **Assembling faces.** *This figure displays exemplary assemblies of facial features from different faces. The source images including the subgraphs are displayed in (a), while (b) presents reconstructed composite graphs. In the first row of (b), combinations of the subgraphs from (a) in different spatial alignments are reconstructed. In the second row, composite graphs combining other subgraphs of the faces from (a) were reconstructed.*

iFFT per sub-band $\mathcal{T}_{\underline{k}_j}$ (cf. steps 2b and 2f of Algorithm 1). Unfortunately, the (i)FFT is the most time-consuming part of the iterative reconstruction algorithm (cf. Table 1), especially when using the LAPACK implementation.

In contrast, the approximation of the Gabor transformed image, which can be calculated in just a few seconds, already leads to a presentable image in one step. To generate and reconstruct this, less than 10 seconds are required. Nonetheless, adding several iterations increases the quality of the reconstructed image. Finally, the computation of mask M and its application took another 7 seconds, so that the image shown in Figure 6(c) was generated in 48 seconds (using MKL for (i)FFT).

4. Applications

The proposed reconstruction algorithm can be used for different applications. The reconstruction of a graph that is extracted from an image directly is not the only use case. Reconstructing modified face graphs provides a broad variety of applications, some of which are presented in this section. All images are created by reconstructing perturbed graphs using the iterative reconstruction algorithm depicted in Section 3.2 with 200 iterations, starting from the approximated Gabor transformed image introduced in Section 3.4.

4.1. *Composite faces*

The first application might be useful for forensic investigations to generate face composite drawings by integrating facial features from different persons into one image. To achieve this, subgraphs of the face graph have to be defined, each containing all nodes of one facial feature like an eye, the nose, or the mouth. Since the subgraphs can be aligned rather freely, a huge variety of differently assembled faces arises. As all subgraphs stem from natural facial images, the composite face graph might be used for face recognition, e. g., in a mug shot database search, which is not possible with commonly used sketch face composite drawings.

Figure 7 shows exemplary combinations of different facial images taken from the CAS-PEAL database.[11] Both images in the first row of Figure 7(b) are compositions of the facial parts displayed in Figure 7(a) with different spatial alignments of the subgraphs. Although the facial subparts are identical, clearly both images contain different faces. To indicate the range of images to generate, the second row of Figure 7(b) shows compositions of other parts, taken from the same original images.

4.2. *Average faces and morphing*

The second face graph modification is the computation of an average face graph. A number of face graphs $\mathcal{G}^{(p)}$ $(1 < p \leq P)$ are extracted from different source images $\mathcal{I}^{(p)}$, which must have approximately the same size and be upright. The *average face graph* $\hat{\mathcal{G}} = (\underline{\hat{\mathcal{L}}}, \underline{\mathcal{E}}, \underline{\hat{\mathcal{J}}})$ is computed as the weighted average of the landmarks and the extended Gabor jets:

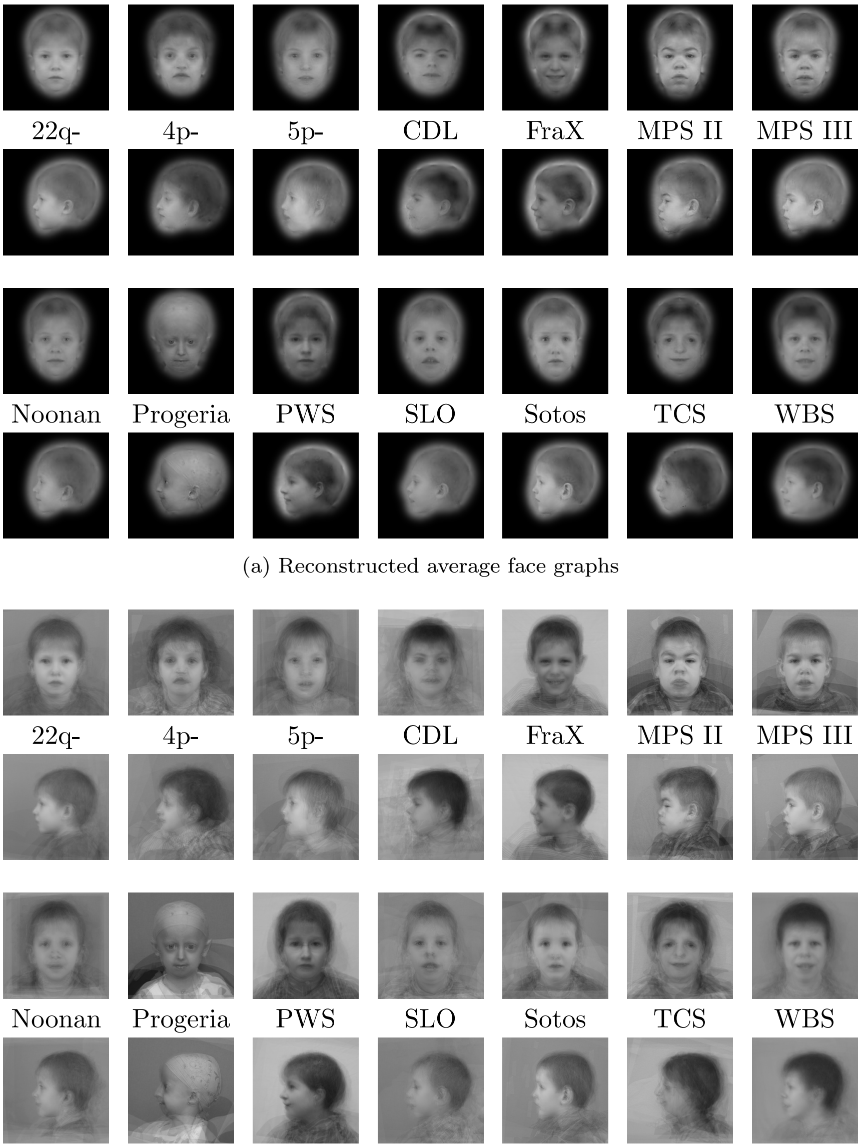
$$\begin{aligned} \hat{\mathcal{L}}_l &= \sum_p v^{(p)} \cdot \mathcal{L}_l^{(p)} \,, \\ \left(\hat{\mathcal{J}}_l\right)_j &= \sum_p v^{(p)} \cdot (\mathcal{J}_l)_j^{(p)} \,, \end{aligned} \tag{4.1}$$

where l iterates over the landmarks and j enumerates the entries of the Gabor jets.

The weights $v^{(p)}$ in Equation (4.1) should (but not necessarily have to) sum up to unity. Usually, they are identical: $v^{(p)} = \frac{1}{P}$. Averaging face graphs can be used for emphasizing common facial anomalies while suppressing differences due to facial expressions, illumination conditions, or noise. Exemplary facial properties are generated by human genome defects that have an impact on the face. The Human Genetics Institute in Essen, Germany, investigates several syndromes with a typical facial appearance. To classify the syndrome from static facial images (semi-)automatically, face graphs were hand-labeled on images of frontal and left profile views of the usually young patients. Permission to publish images was given by patients or their wardens.[4]

The list of syndromes and their abbreviations, as well as the number of averaged face graphs, is given in Table 2. As Figure 8(a) shows, averaging face graphs of one syndrome can emphasize prominent facial properties of the syndromes while suppressing facial expressions, illumination conditions, and patient identity. Shown

Preprint of an article in press 2015 at
Journal of Wavelets, Multiresolution and Information Processing
http://www.worldscientific.com/worldscinet/ijwmip



(a) Reconstructed average face graphs

(b) Average images

Figure 8. **Reconstruction from average graphs versus averaging gray values.** *This figure displays averaged images in frontal and profile view for 14 different syndromes from the study database.*[3] *While in (b) the gray values of the source images were averaged, (a) shows the reconstructions of the hand-labeled face graphs.*

to medical experts, the syndromes could be identified from these images clearly.[3] To show the advantage of using the reconstruction method to average images of syndromes, the same experiment was repeated, this time averaging pixel gray values

Abbrv.	Frontal	Profile	Full Name
22q-	26	26	Microdeletion 22q
4p-	12	12	Wolf-Hirschhorn syndrome
5p-	16	12	Cri-du-chat syndrome
CDL	17	17	Cornelia de Lange syndrome
FraX	11	12	Fragile X syndrome
MPS2	7	7	Mucopolysaccharidosis II
MPS3	8	8	Mucopolysaccharidosis III
Noonan	15	15	Noonan syndrome
PWS	13	13	Prader-Willi syndrome
Progeria	5	5	Progeria
SLO	17	17	Smith-Lemli-Opitz syndrome
Sotos	15	15	Sotos syndrome
TCS	12	12	Treacher Collins syndrome
WBS	44	44	Williams-Beuren syndrome

Table 2. **Number of images per syndrome.** *This table shows the subdivision of the study database*[3] *into the accordant syndromes and the number of images in frontal and left profile view.*

instead of Gabor jets and landmark positions:

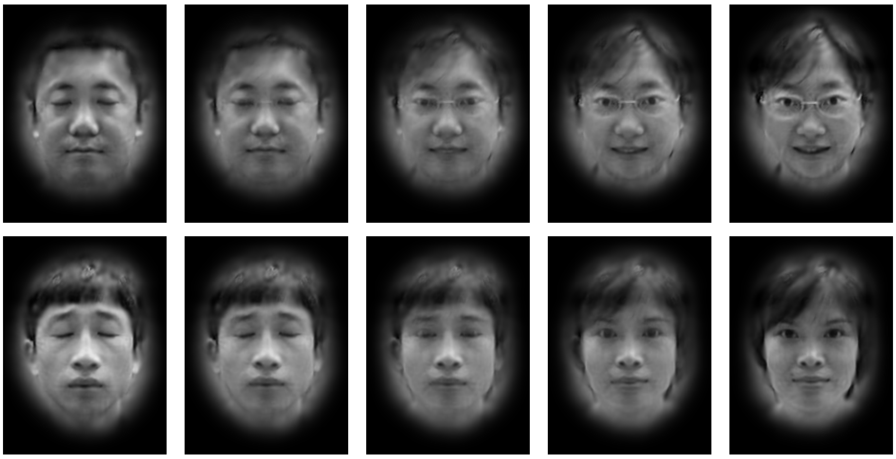
$$\hat{\mathcal{I}}(\underline{x}) = \sum_p v^{(p)} \cdot \mathcal{I}^{(p)}(\underline{x})\,. \tag{4.2}$$

Pixel averaging only works when images are aligned, e. g., to the hand-labeled eye positions in case of frontal view. The resulting average images for the syndromes are shown in Figure 8(b). In comparison, the reconstructions in Figure 8(a) are much clearer since Equation (4.1) averages both local texture (Gabor jets) and geometry (landmark positions), while Equation (4.2) averages only global texture (pixel gray values).

Graph averaging can also be used to morph one face into another one. To do a morph between two faces, the weights $v^{(1)}$ of the first graph slowly slides from zero to one, while the second weight calculates as: $v^{(2)} = 1 - v^{(1)}$. Two exemplary morphs are presented in Figure 9, morphing between the faces shown in Figure 9(a) and Figure 9(e).

4.3. *Caricatures*

Combining and extending face graph averaging and face graph morphing, *caricature graphs* can be created. A caricature is an image with exaggerated facial features.



(a) $v^{(1)} = 1$ $v^{(2)} = 0$

(b) $v^{(1)} = 0.75$ $v^{(2)} = 0.25$

(c) $v^{(1)} = 0.5$ $v^{(2)} = 0.5$

(d) $v^{(1)} = 0.25$ $v^{(2)} = 0.75$

(e) $v^{(1)} = 0$ $v^{(2)} = 1$

Figure 9. **Face morphing.** *This figure displays five steps of a morph from the face in (a) into the face in (e). The weights for the two graphs are given below the images.*

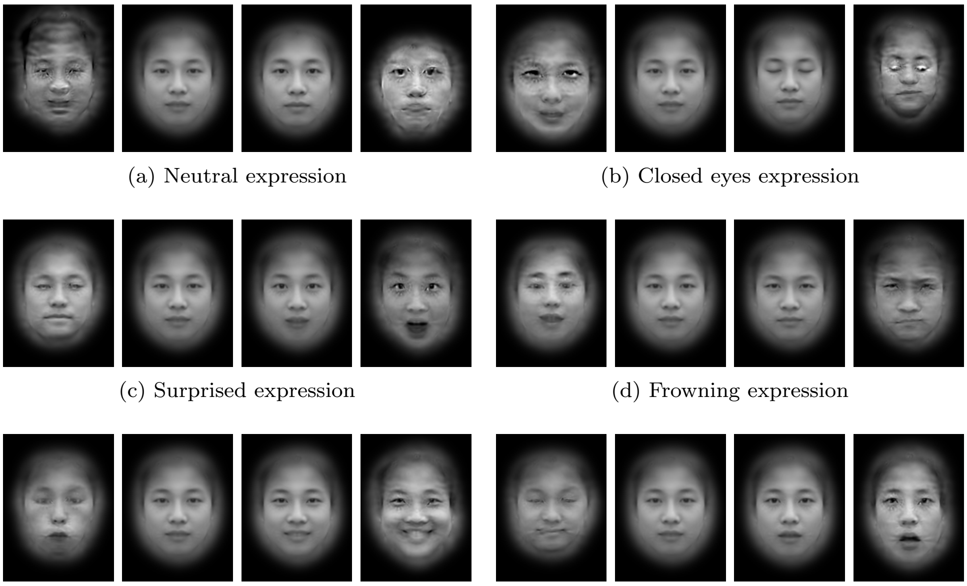
Similarly, caricature graphs are face graphs containing exaggerated texture and geometrical features. To generate a caricature graph $\mathcal{G}_{T+v}$, an average graph $\hat{\mathcal{G}}$ and a target graph $\mathcal{G}_T$ that includes the facial properties to be exaggerated are morphed:

$$\begin{aligned} \mathcal{G}_{T+v} &= v \cdot \mathcal{G}_T + (1 - v) \cdot \hat{\mathcal{G}} \\ &= \hat{\mathcal{G}} + v \cdot (\mathcal{G}_T - \hat{\mathcal{G}}) \,. \end{aligned} \tag{4.3}$$

For $v > 1$, caricatures in different magnitudes are generated, the higher the weight, the more the exaggeration. Weights $v = 1$ and $v = 0$ generate the target graph $\mathcal{G}_T$ and the average graph $\hat{\mathcal{G}}$, respectively. The result for $v < 0$ is an *anti-caricature graph*, where the facial features of the target graph are inverted. Hence, reconstructions of anti-caricature graphs show what the target graph does not look like.

As an example, caricatures and anti-caricatures of six facial expressions are displayed in Figure 10. The face graphs of 2261 images of 377 people from within the CAS-PEAL database[11] were detected semi-automatically employing the maximum likelihood landmark localization[12, 13] after the images were aligned according to their hand-labeled eye positions. The average graph shown in the second column of Figure 10 was generated as an average of all 2261 face graphs, while the target graphs displayed in the third column are averages of 377 face graphs containing the same facial expression.

The last column in Figure 10 presents caricatures of the facial expressions in magnitude $v = 4$. Especially, the laughing, frowning, and surprised caricatures visualize exaggerations of the corresponding facial expression excellently. Also, the opened mouth and surprised caricature are clearly distinguishable. Since some land-



(a) Neutral expression

(b) Closed eyes expression

(c) Surprised expression

(d) Frowning expression

(e) Laughing expression

(f) Opened mouth expression

Figure 10. **Facial expression caricatures.** *This figure shows reconstructed caricature and anti-caricature graphs of six different facial expressions. In each subfigure, the second image shows the averaged graph over all expressions, while the third image contains the average graph for the current expression. Image four shows the caricature in magnitude $v = 4$, while the first image visualizes the anti-caricature with $v = -3$.*

marks of the mouths in the surprised expression were misplaced during automatic landmark localization, the mouth region is blurry in the surprised caricatures. The first column of Figure 10 shows anti-caricatures of the expressions. Clearly, they show how facial expressions do not look like. Apparently, the laughing anti-caricature seems to show a sad face, although sad was not among the expressions contained in the CAS-PEAL database. Since the definition of an anti-neutral expression is ambiguous, the anti-caricature in Figure 10(a) does not show a clear image.

5. Outlook

In this paper, we introduced an iterative reconstruction algorithm that reconstructs images from graphs labeled with Gabor jets, optionally using extended discrete Gabor wavelet families. We further added a different initial condition for the iteration and showed that a comparably good image can be reconstructed in reasonable time. We showed different ways of incorporating several face graphs into one. These graph mixtures were reconstructed to show that the different ways of graph merging lead to valid graphs and, thus, could be used for several practical applications.

In future work, the iterative reconstruction algorithm might be improved by replacing step 2a of Algorithm 1 by an alternative with faster convergence. Also, the memory load could be decreased heavily by implementing the reconstruction sub-band by sub-band. To increase the number of applications, different, i. e., nonlinear graph combination algorithms can be investigated. With such methods, e. g., the aging process of people could be predicted.

Acknowledgments

We gratefully acknowledge funding from Deutsche Forschungsgemeinschaft (BO 1955/2-1, BO 1955/2-3, WU 314/2-1, WU 314/6-2). This work was part of the CRANIRARE Network funded through a grant from the German Ministry of Research and Education to D.W. (BMBF 01GM0802). 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.[11]

Bibliography

1. M. Becker, E. Kefalea, E. Maël, C. von der Malsburg, M. Pagel, J. Triesch, J. C. Vorbrüggen, R. P. Würtz and S. Zadel, GripSee: A gesture-controlled robot for object perception and manipulation, *Autonomous Robots* **6**(2) (1999) 203–221.
2. C. Blatter, *Wavelets - Eine Einführung*, 2. edn. (Vieweg, Wiesbaden, 2003).
3. S. Böhringer, M. Günther, S. Sinigerova, R. Würtz, B. Horsthemke and D. Wieczorek, Automated syndrome detection in a set of clinical facial photographs, *American Journal of Medical Genetics Part A* **155** (2011) 2161–2169.
4. S. Böhringer, F. van der Lijn, F. Liu, S. Sinigerova, S. Birnbaum, E. Mangold, S. Klein, W. Niessen, M. Breteler, R. Würtz, M. Noethen, B. Horsthemke, D. Wieczorek and M. Kayser, Genetic determination of the human facial shape: links between cleft-lips and normal variation, *European Journal of Human Genetics* **19** (2011) 1192–1197.
5. S. Böhringer, T. Vollmar, C. Tasse, R. Würtz, G. Gillessen-Kaesbach, B. Horsthemke and D. Wieczorek, Syndrome identification based on 2D analysis software, *European Journal of Human Genetics* **14**(10) (2006) 1082–1089.
6. J. G. Daugman, Complete Discrete 2-D Gabor Transforms by Neural Networks for Image Analysis and Compression, *IEEE Transactions on Acoustics, Speech, and Signal Processing* **36** (July 1988) 1169–1179.
7. J. G. Daugman, Non-orthogonal wavelet representations in relaxation networks: image encoding and analysis with biological visual primitives, *New Developments in Neural Computing*, eds. J. Taylor and C. Mannion (Institute of Physics Press, 1989), pp. 233–250.
8. J. G. Daugman, Quadrature-phase simple-cell pairs are appropriately described in complex analytic form, *Journal of the Optical Society of America* **10** (February 1993) 375–377.
9. G. S. Donatti, O. Lomp and R. P. Würtz, Evolutionary optimization of growing neural gas parameters for object categorization and recognition, *Proc. IJCNN*, (IEEE Computer Society, Los Alamitos, CA, 2010), pp. 1862–1869.
10. R. Frohner, R. Würtz, R. Kosilek and H. Schneider, Optimierung der Gesichtsklassifikation bei der Erkennung von Akromegalie, *Austrian Journal of Clinical Endocrinology and Metabolism* **6**(3) (2013) 20–24.
11. 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 Transactions on Systems, Man, and Cybernetics Part A* **38**(1) (2008) 149–161.

12. M. Günther, Statistical Gabor graph based techniques for the detection, recognition, classification, and visualization of human faces, PhD thesis, Fakultät für Informatik und Automatisierung, Technische Universität Ilmenau, Germany (July 2011).
13. M. Günther and R. P. Würtz, Face detection and recognition using maximum likelihood classifiers on Gabor graphs, *International Journal of Pattern Recognition and Artificial Intelligence* **23** (May 2009) 433–461.
14. K. D. Kammeyer and K. Kroschel, *Digitale Signalverarbeitung: Filterung und Spektralanalyse* (Teubner, Stuttgart, 1989).
15. R. P. Kosilek, J. Schopohl, M. Grünke, C. Dimopoulou, G. K. Stalla, A. Lammert, M. Reincke, M. Günther, R. P. Würtz and H. J. Schneider, Automatic face classification of Cushings syndrome in women – a novel screening approach, *Experimental and Clinical Endocrinology and Diabetes* **121**(09) (2013) 561–564.
16. 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 Transactions on Computers* **42**(3) (1993) 300–311.
17. T. S. Lee, Image representation using 2D Gabor wavelets, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(10) (1996) 959–971.
18. H. Loos, D. Wieczorek, R. Würtz., C. Malsburg and B. Horsthemke, Computer-based recognition of dysmorphic faces, *European Journal of Human Genetics* **11** (2003) 555–560.
19. M. Pötzsch, Die Behandlung der Wavelet-Transformation von Bildern in der Nähe von Objektkanten, Internal Report IRINI 94-04, Institut für Neuroinformatik, Ruhr-Universität Bochum, D-44780 Bochum, Germany (May 1994). Diplomarbeit.
20. M. Pötzsch, T. Maurer, L. Wiskott and C. Malsburg, Reconstruction from graphs labeled with responses of Gabor filters, *Proceedings of the ICANN 1996*, eds. C. v.d. Malsburg, W. v. Seelen, J. Vorbrüggen and B. Sendhoff *Springer Verlag, Berlin, Heidelberg, New York*, (Bochum, 1996), pp. 845–850.
21. H. J. Schneider, R. P. Kosilek, M. Günther, J. Schopohl, J. Römmler, G. K. Stalla, C. Sievers, M. Reincke and R. P. Würtz, A novel approach for the detection of acromegaly: accuracy of diagnosis by automatic face classification, *Journal of Clinical Endocrinology and Metabolism* **96**(7) (2011) 2074–2080.
22. Y. Sheng, Wavelet transform, *The Transforms and Applications Handbook, Second Edition*, ed. A. D. Poularikas (CRC press, 2000).
23. T. Vollmar, B. Maus, R. Würtz, G. Gillessen-Kaesbach, B. Horsthemke, D. Wieczorek and S. Böhringer, Impact of geometry and viewing angle on classification accuracy of 2D based analysis of dysmorphic faces., *European Journal of Medical Genetics* **51** (2008) 44–53.
24. J. Wieghardt, R. P. Würtz and C. von der Malsburg, Learning the topology of object views, *Computer Vision - ECCV 2002*, eds. A. Heyden, G. Sparr, M. Nielsen and P. Johansen *LNCS*(2353), (Springer, 2002), pp. IV–747–IV–760.
25. L. Wiskott, J.-M. Fellous, N. Krüger and C. von der Malsburg, Face recognition by elastic bunch graph matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(7) (1997) 775–779.
26. I. Wundrich, Untersuchungen zur Rekonstruierbarkeit lokaler Bildmerkmale aus der Gaborwavelettransformierten, Master's thesis, Institut für Neuroinformatik, Ruhr-Universität Bochum, Germany (October 1998).
27. I. J. Wundrich, C. von der Malsburg and R. P. Würtz, Image reconstruction from Gabor magnitudes, *BMCV '02: Proceedings of the Second International Workshop on*

Biologically Motivated Computer Vision, (Springer-Verlag, London, UK, 2002), pp. 117–126.

28. I. J. Wundrich, C. von der Malsburg and R. P. Würtz, Image representation by complex cell responses, *Neural Computation* **16**(12) (2004) 2563–2575.