Statistical Gabor Graph Based Techniques for the Detection, Recognition, Classification, and Visualization of Human Faces

Dissertation
Zur Erlangung des akademischen Grades
Doktoringenieur (Dr.-Ing.)

vorgelegt der Fakultät für Informatik und Automatisierung
der Technischen Universität Ilmenau

von	**Dipl.-Inf. Manuel Günther**
geboren am	**3. Dezember 1979 in Leinefelde**

vorgelegt am	**30. Juni 2011**
Gutachter:	**1.) Prof. Dr. Horst-Michael Groß**
	2.) PD. Dr. Rolf P. Würtz
	3.) Prof. Dr. Thomas Zielke

Abstract

In this work, I focus in a simple parameter-free statistical model that requires few training data and can be trained fast. I show that the model is well suited for face detection, person identification, and classification of facial properties.

For face detection, the well known elastic bunch graph matching algorithm is adapted to learn appearance probabilities of facial features. Furthermore, texture features are transformed to be used for the detection of faces in different sizes and in-plane rotation angles. In order to place facial landmarks more reliably and to increase face recognition accuracy, images are automatically standardized according to the found scale and angle of the face. It is shown that both extensions of the elastic bunch graph matching algorithm work well with only few hand-labeled training examples and that the face detection can be accelerated.

After applying small changes to the model, it can be employed for identifying a person that is shown in an image. In opposition to other state-of-the-art identification algorithms, the model learns how two facial images can be compared most reasonably. For both the intrapersonal and the extrapersonal class, each one statistical model is approximated. The intrapersonal class consists of comparisons of images showing the same person, while the extrapersonal class contains image comparisons of different identities. Utilizing face graphs, it is shown empirically that the statistical model is able to reliably recognize faces in different sizes and with different facial expressions. Identification under illumination variation is still a tough problem, but it is illustrated that the proposed model is indeed able to outperform state-of-the-art face recognition approaches. This is also reached by exploiting parts of the texture descriptors that are ignored by most current algorithms.

The very same model is employed for classification of facial expression and illumination condition by simply exchanging training image pairs in the intrapersonal and extrapersonal class. Another current classification challenge is to diagnose several genetic syndromes, which have an impact on the facial appearance, by processing facial images. The statistical model is modified slightly in order to implement this classification automatically. It is shown that the proposed parameter-free model is able to classify genetic syndromes better than highly specialized classification methods with carefully chosen parameters. To be applicable in clinical practice, a Java program was developed, which allows medical personnel to administrate a database of

facial images, automatically detect the faces in the images, manually adjust landmark positions if necessary, and diagnose a syndrome according to these images.

The visualization of the used texture features is solved by a combination of a solid mathematical foundation with an engineer's approach. Using this reconstruction method, the possibility of visualizing modified texture features is shown. As an example, texture features of different patients with the same genetic syndrome are combined into one texture feature, in whose visualization medical experts could positively identify the corresponding syndrome.

Zusammenfassung

In dieser Doktorarbeit befasse ich mich mit einem einfachen, parameterfreien statistischen Modell, das nur wenige Trainingsdaten benötigt und schnell zu trainieren ist. Am Beispiel der Gesichtsdetektion, der Identifikation und der Klassifikation von Gesichtseigenschaften zeige ich einige Einsatzgebiete dieses statistischen Modells auf.

Zur Detektion wird das weit bekannte Verfahren des Elastic Bunch Graph Matching adaptiert, indem das statistische Modell zum Lernen von Auftretenswahrscheinlichkeiten von Gesichtsmerkmalen eingesetzt wird. Weiterhin werden Texturmerkmale derart transformiert, dass sie zur Detektion von Gesichtern in unterschiedlichen Größen und mit unterschiedlichen Rotationswinkeln in der Bild-Ebene verwendet werden können. Um Landmarken im Gesicht besser platzieren zu können und um bessere Identifikationsergebnisse zu erzielen, wird das Bild anhand der gefundenen Gesichtsgröße und -rotation automatisch standardisiert. Es wird gezeigt, dass beide Erweiterungen des Elastic Bunch Graph Matching auch mit wenigen hand-gelabelten Trainingsdaten robust funktionieren, und dass die Detektion beschleunigt werden kann.

Durch eine kleine Veränderung am Modell wird dieses auch zur Identifikation der Person, die in einem Bild zu sehen ist, eingesetzt. Im Gegensatz zu den meisten aktuellen Identifikationsverfahren lernt das Modell, auf welche Art und Weise zwei Gesichtsbilder am sinnvollsten miteinander zu vergleichen sind. Dazu wird jeweils ein statistisches Modell für die intrapersonale und für die extrapersonale Klasse approximiert. Die intrapersonale Klasse besteht aus Vergleichen von Bildern der gleichen Person, während in der extrapersonalen Klasse Vergleiche unterschiedlicher Personen enthalten sind. Unter Verwendung der Gesichtsgraphen wird empirisch belegt, dass das statische Modell in der Lage ist, verlässlich Gesichter in unterschiedlichen Größen und mit unterschiedlichen Gesichtsausdrücken wiederzuerkennen. Die Identifikation unter veränderter Beleuchtung ist nach wie vor ein schweres Problem, es wird jedoch gezeigt, dass das einfache, parameterfreie statistische Modell andere aktuelle Ansätze zur Identifikation auszustechen vermag. Dies wird auch erreicht, indem Elemente der Texturdeskriptoren verwendet werden, die von den meisten aktuellen Algorithmen ignoriert werden.

Das gleiche Modell wird auch zur Klassifikation von Gesichtsausdrücken und Beleuchtungsbedingungen verwendet. Dies ist möglich, indem die verwendeten Trainingspaare in der intrapersonalen und der extrapersonalen

Klasse ausgetauscht werden. Ein weiteres aktuelles Klassifikationsproblem versucht, anhand von Gesichtsbildern von Personen mit genetischen Syndromen ebendiese Syndrome zu diagnostizieren. Das statistische Modell wird leicht verändert eingesetzt, um diese Klassifikation automatisch durchführen zu können. Es wird gezeigt, dass dieses parameterfreie Modell in der Lage ist, die Syndrome besser zu klassifizieren, als aktuelle hochspezialisierte Klassifikationsmethoden mit wohlgewählten Parametern. Zur Anwendung in der klinischen Praxis wurde ein Java-Programm entwickelt, das dem medizinischen Personal erlaubt, Bilder von Patienten zu verwalten, die Gesichter in den Bildern automatisch zu detektieren und gegebenenfalls Landmarken manuelle zu korrigieren, und ein Syndrom anhand dieser Bilder zu diagnostizieren.

Durch die Kombination einer soliden mathematischen Grundlage mit einer ingenieurstechnischen Herangehensweise ist es gelungen, eine gute Visualisierung der verwendeten Texturmerkmale zu erreichen. Es wird gezeigt, dass mit Hilfe dieser Rekonstruktionsmethode auch abgewandelte Texturmerkmale visualisiert werden können. Unter Anderem wird aus Texturmerkmalen unterschiedlicher Patienten mit dem gleichen genetischen Syndrom ein mittleres Texturmerkmal kombiniert, in dessen Visualisierung medizinische Experten das entsprechende Syndrom eindeutig erkennen konnten.

Danksagung

Bevor ich tiefer in die Materie dieser Arbeit eintauche, möchte ich es nicht versäumen, denjenigen Personen zu danken, die zum Gelingen dieser Doktorarbeit beigetragen haben.

Der größte Dank gebührt zweifelsfrei meinem Betreuer am Institut für Neuroinformatik, PD. Dr. Rolf Würtz, der mir dieses sehr spannende Thema angeboten und mir stets mit Rat und Tat zur Seite gestanden hat. Insbesondere danke ich ihm dafür, dass er mir die Freiheit gestattet hat, meine eigenen Ideen und Vorstellungen in die Tat umzusetzen, während er die angefallenen administrativen Aufgaben klaglos übernommen hat.

Ein sehr großer Dank geht natürlich auch an meinen Betreuer an der TU Ilmenau, Prof. Dr. Horst-Michael Groß, der mich schon während meines Studiums auf vielen Wegen begleitet hat. Durch ihn wurde ich auf das Thema der Gesichtserkennung aufmerksam, und auf seinen Rat hin bin ich nach Bochum und zuletzt auch an das Institut für Neuroinformatik der Ruhr-Universität gekommen.

Danken möchte ich auch den Professoren des Institutes, namentlich Prof. Dr. Gregor Schöner, Prof. Dr. Laurenz Wiskott und Prof. Dr. Christoph von der Malsburg. Durch ihre interdisziplinäre Arbeit haben sie mir die Möglichkeit eröffnet, die Themen dieser Arbeit aus einem weiteren Blickwinkel zu betrachten. Weiterhin möchte ich der Administration des Institutes, Arno Berg, Michael Neef (in stillem Gedenken) und Michael Ziesmer danken, die stets für einen funktionsfähigen Arbeitsplatz gesorgt haben. Auch den Sekretärinnen des Instituts gebührt ein Dank. Selbiges gilt auch für die Kollegen des Instituts, die ich unmöglich alle aufzählen kann. Stellvertretend danke ich:

- Dr. Marco Müller für die Einführung in das Elastic Bunch Graph Matching und die Erstellung der FaceGen-Datenbank.

- Guillermo Sebastián Donatti für viele fruchtbare Diskussionen, nicht zuletzt über das Software-Design unserer gemeinsamen C++-Bibliothek *pragma*.

- Dr. Günter Westphal für die Erstellung von pragma.

- Dr. Susanne Winter, Markus Lessmann, Thomas Walter, Andreas Wille und Oliver Lomp sowie oben aufgeführte Kollegen für die hilfreichen Kommentare zu dieser Arbeit.

- Dr. Maximilian Krüger, Dr. Andreas Tewes, Dr. Wilfried Horn, Dr. Achim Schäfer, Dr. Marek Barwinski, Dr. Agnieska Grabska-Barwin-

ska, Dr. Ellen Otte und vielen, vielen mehr, die mich auf meiner wissenschaftlichen Reise begleitet haben.

Weiterhin möchte ich den Studenten Dennis Haufe, Norbert Neuser und Benedikt Stratmann danken, die in meinem Auftrag im Rahmen von Bachelor- und Master-Arbeiten Teilaspekte meiner Arbeit näher beleuchtet haben.

Einen Dank möchte ich auch unseren Kooperationspartnern Prof. Dr. Bernhard Horsthemke, Prof. Dr. Dagmar Wieczorek, Dr. Stefan Böhringer, PD. Dr. Harald Schneider und Robert Kosilek für ihre fruchtbare Zusammenarbeit auf dem Gebiet der Klassifikation von genetischen und endokrinologischen Syndromen aussprechen.

Dank gilt auch Prof. Dr. Thomas Zielke, unter dessen Betreuung ich meine Diplomarbeit verfassen durfte. Ohne ihn wäre ich wohl kaum in Bochum gelandet.

Ein ganz spezieller Dank geht auch an meine Eltern Peter und Agnes Günther, die mich während meines gesamten Studiums und auch während meiner Zeit hier in Bochum tatkräftig unterstützt haben, obwohl sie in dieser Zeit selbst schwere Schicksalsschläge hinnehmen mussten. Auch meine Schwestern Christa, Ramona und Gabi sowie ihren Familien sollen vom Dank nicht ausgeschlossen sein.

Contents

Chapter 1

Introduction

Automatic face recognition is a hot topic during the last 20 years. The developed use cases for face recognition are numerous, but they can be classified into four different applications: authentication, identification, surveillance, and classification. Authentication proves the identity of a person, identification finds out which identity out of a set of persons is present, surveillance tries to monitor public places and alert on a set of interesting persons, and finally classification estimates properties of the face.

All of these tasks require an image from either a still camera or a video camera that includes at least one face. Given such an image, the first step is to detect the face in it. Several methods for face detection were proposed in the last years, beginning with Turk and Pentlands *eigenfaces* [89] and the *elastic bunch graph matching* from Wiskott *et al.* [96], continuing with the probabilistic eigenface template search from Moghaddam and Pentland [53], and several artificial neural network applications [74, 75]. The last great invention was the Viola-Jones face detector [90], which is in most cases very fast and reliable. Despite of such achievements, applying *hierarchical slow feature analysis* [100] to face detection already showed [54] that it can outperform the Viola-Jones detector.

After detecting the face, image features must be extracted in order to recognize its identity. Most often, one of the two main types of features are used for recognition. The first and most popular one is the eigenface feature from Sirovich and Kirby [82], which is a pixel-based engineer's approach that extracts global facial features. The second feature type, namely the *face graph* from Wiskott [96] and Würtz [105], which assembles local texture features, is more of a scientist's approach that mimics human image feature extraction. Two completely new approaches came up in the last years. In 1999, Lowe [46] introduced the *scale invariant feature transformation* that generates features, which are scale, rotation, and translation invariant. Lately, the *local binary pattern* [1] and its extension to the *local Gabor binary pattern histogram sequence* [110] were invented to overcome some issues all other algorithms suffer from, but still there is a lot to be researched in this direction.

The recognition step usually relies on a comparison of features extracted from two different images. How these features are compared highly depends

on the feature type. Eigenfaces might be matched directly by computing a distance measure between two eigenface features [89], or can be further processed, e. g., using *linear discriminant analysis* [19, 112] or *independent component analysis* [3, 15]. Face graphs, which comprise local texture and geometry information, are usually compared by computing texture similarity, while latest approaches define graph-theoretical measures on the geometrical relation of the facial parts [72]. Scale invariant feature transformation features are matched with distance or similarity functions, while local binary pattern histogram sequences usually are compared using a histogram intersection measure.

One common issue of these algorithms is that they use enhanced features, eventually by learning properties of these features, but the feature comparison itself employs invariable measures. One approach into learning how to compare two features was done by Moghaddam and Pentland [51, 50]. They introduced the *Bayesian intrapersonal/extrapersonal classifier*, which statistically learns the difference in comparing two images of the same person in opposition to the comparison of images of two different persons. Still, their approach relies on performing *principal component analysis* of image differences, which requires lots of training data and has parameters that have to be set up carefully. Another approach was introduced in my diploma thesis [28], where I used evolutionary algorithms to discover those parts of the texture features and those locations that are best suited for the comparison of two face graphs.

To test how well face recognition algorithms work and which algorithm performs best, *face recognition vendor tests* are run periodically. The recognition accuracies enormously increased during the last two decades, i. e., the *verification rate* rose from 21% in the test from 1993 via 80% in 2002 to 99% in the face recognition vendor test 2006 [66]. The latest breakthrough was the usage of high resolution colored images in combination with a large training set. Still, recognizing faces under uncontrolled illumination conditions is not yet solved satisfactorily [66, 11].

Classification of properties like gender, age, facial expression, pose, or lighting conditions of facial images also aroused scientists' interests. Knowledge of all of these properties can help in the face recognition task, e. g., by trying to process images such that poses or facial expressions are eliminated [86] or by training recognition systems specifically for the estimated facial property. Often, classification is done by *support vector machines*, e. g., Graf and Wichmann [25] and Ji and Lu [35] used support vector machines for gender classification, while Martin *et al.* [48], Ren [73], and Fischer [21] classified facial expressions with support vector machines. For the estimation of facial expressions and poses, *active appearance models* combining shape and texture

information are widely used [48, 73, 40, 76]. Tewes [87] extended these to a *flexible object model* by combining face graph and active appearance model approaches.

In general, the features that are extracted from facial images are usually high dimensional and cannot be interpreted easily. An appropriate way to see, what kind of information is embedded in the extracted features, is the reconstruction of an image from these features. For the kind of features used in this work, some reconstruction procedures were presented [71, 103], but the results are not yet accurate.

1.1 Applications

Face recognition systems are already employed for authentication in access control systems, where illumination conditions can be controlled and the participants cooperate by looking into the camera and showing a neutral facial expression. As soon as environment conditions can be controlled, face recognition is practically solved [66]. Unfortunately, controlling the environment is impossible in most cases, and uncontrolled illumination conditions basically prevents any face recognition system to work properly [66, 11].

Since 2005, the German passport includes a computer chip containing biometric information like a digital photo and a fingerprint, an iris scan is planned to be included as well. On passport control, these biometric features can be checked and, hence, the document is copy-proof. In general, the stored biometric facial image needs to show a neutral expression since face recognition systems are said to be prone to facial expressions. Some tests support this opinion [23], while state-of-the-art recognition systems seem to be able to deal successfully with uncontrolled facial expressions [66].

A novel application of face recognition is in the field of social human-robot-interactions. Amongst others, autonomous service robots are built by the Neuroinformatics and Cognitive Robotics Lab [27, 26] in the Technical University of Ilmenau. In a first study [26], these robots were positioned into home improvement stores and led customers to the products of their choice. To communicate with a person, the robot should decide whether or not the person is already known and maybe continue ongoing tours. Additionally, it should automatically perceive the emotional state of its counterpart by classifying his or her facial expression and adapt its dialog accordingly [48]. One of the main face recognition challenges for mobile robots surely is that pose and illumination are uncontrolled and might change due to human or robot movements.

A completely different use case is automatic classification of genetic syn-

dromes that have an impact on the face. The topic was introduced by Loos *et al.* [45] in 2003 and extended by Böhringer *et al.* [8]. They used frontal face graphs and the stored texture information to identify genetic syndromes on the basis of a single facial image of a patient. They opined that their classification accuracy was in the order of medical experts, which were supposed to classify the same facial images without further information about the patient. On the basis of these results, they suggested an application that makes a preselection of possible syndromes to allow medical experts to concentrate on tests according to the propounded syndromes. Later, Vollmar *et al.* [91] added face graphs of profile views of the same patients and incorporated texture and geometrical information to enhance classification. Unfortunately, these results are not directly transferable into clinical practice as we [7] showed lately, so there is still a lot of research to be done.

1.2 Methodological Contribution

Most of the current face detection and face recognition algorithms need a lot of hand-labeled or at least hand-classified training data. Additionally, the parameters of these algorithms need to be adjusted to each problem independently. The main contribution of this work is the introduction of a simple parameter-free statistical model that requires only few training data. With slight modifications this model can be deployed to face detection, feature point localization, identity recognition, and image property classification.

Most holistic face detection algorithms detect faces only upright, and in-plane rotation of the face is usually not compensated for. In contrast, eye detectors can be used to normalize the image according to the detected eyes, but these feature detectors might be perturbed by glasses, closed eyes or facial hair overlapping the eyes. I propose a holistic algorithm that allows for (in-plane) multi-angle and multi-scale face detection and normalization and that does not require training data specialized for these cases.

Current face recognition systems either perform training stages on the single feature level, or compare faces by averaging local texture similarities without a training stage. In this work, a mixture of both approaches is presented by learning, which local texture similarities are reliable and which ones differ much. Several texture similarity functions are tested, including those that are based on the parts of the local texture descriptor, namely the Gabor phases, which are ignored by most current face recognition algorithms. Local texture similarities and geometrical relations are integrated to form a more robust face recognition and classification system. Furthermore, it is pointed out that geometrical normalization according to scale and in-plane

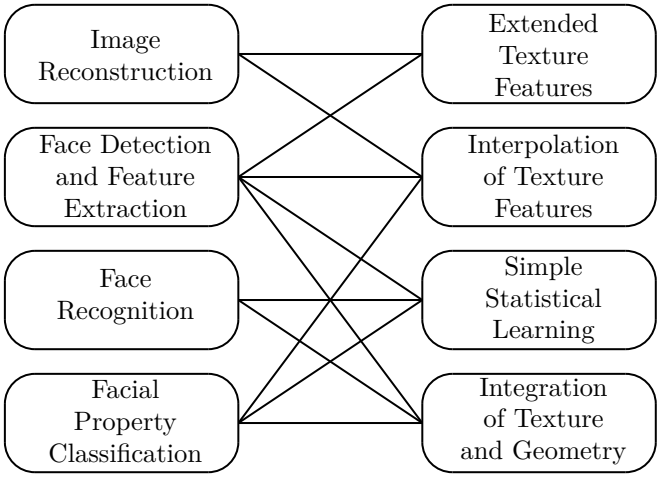


Figure 1.1: Basic Ideas: *This figure displays the four basic problems tackled by this work and their connections to the four basic ideas that are proposed in this thesis.*

rotation angle of the face improves identification.

After several approaches have been made, I soundly solve the reconstruction of images from face graphs by incorporating theoretical investigation results into common engineering solutions for the approximation of positions not covered by the face graph. Additionally, extended texture features are used to reconstruct gray value or even colored images. A procedure for interpolating and averaging these texture features is presented. By reconstructing modified texture features it is pointed out that these features still include useful information.

Figure 1.1 shows a connection graph from the four topics of this thesis to the four basic ideas that are approached in the following chapters. For example, the image reconstruction partly relies on extended texture features and the interpolation of these, while facial property classification uses averaging of texture features, too, and furthermore the simple learning strategy, and the integration of texture and geometry information.

1.3 Demarcation

There are also some issues that are not covered by this work, but which are important for the topics of this thesis. One detail for detection and recognition is the parametrization of the texture feature extraction, which I use[1] according to the setup reported in literature [10, 20, 96].

A related question is which image size to use? Contemporaneously, Jiménez *et al.* [36] and Haufe [30] conducted experiments testing which image size performs best for face recognition. I use a size that performs comparably well according to those results, but maybe different image sizes could further improve landmark localization, recognition, or classification accuracies. The latest trend in face recognition is to use high resolution colored 3D facial images. In the present work, neither color, nor 3D information are utilized for detection or recognition. All algorithms proposed in this work rely on low resolution 2D gray level images.

The face recognition algorithm that I propose works on still images only. How well this algorithm can be applied to video data and if it might be stabilized by estimating the identity over several successive frames needs to be investigated. Furthermore, the system works on frontal images only. Comparing the texture or the geometry of facial parts under out-of-plane image rotations is impossible, e. g., since some parts might not be visible. A striking model-based approach to the comparison of incomparable representations of faces, e. g., frontal and profile views was introduced by Müller [56], and a mapping from non-frontal texture to frontal view texture was proposed by Tewes [86], further research into this direction will hopefully follow.

It is still an open question whether face graphs or grid graphs are better suited for face detection or recognition. Furthermore, it is yet unknown which facial parts are important, and whether homogeneous face parts like the cheek are useful for detection or recognition. In this work, these questions are disregarded and face graphs with a default setup are used throughout. Nonetheless, some experiments comparing the utility of rigid face graphs and elastic face graphs are executed.

1.4 Organization of the Present Work

The outline of this work is as follows: In Chapter 2, the introduction to Gabor wavelets and the Gabor wavelet transform is given. In addition, the theoretical background that is needed for the reconstruction of images, which

[1] Extensions of texture features are used, but texture comparison is always calculated on default features.

is detailed in Chapter 6, is settled. Afterwards, the Gabor jet texture features and the face graphs are presented. These face graphs and are used as face representations by the subsequent chapters.

In Chapter 3, different face detection algorithms are treated in more detail. Afterwards, two major extensions of the elastic bunch graph matching algorithm – a simple statistical add-on and the multi-scale and multi-angle face detection – are introduced.

After presenting the required face recognition quality measures, Chapter 4 gives a small overview of some standard face recognition algorithms and the feature types that are used by these methods. Two algorithms are integrated into a simple statistical technique that learns how to compare the texture and the geometry stored in face graphs. In Section 4.4, this recognition procedure is extended to a classification algorithm and the Facial Image Diagnostic Aid application is presented.

In Chapter 5, experiments for face detection, feature point localization, face recognition, and facial property classification are executed on artificial and on natural facial image databases and the results are compared to state-of-the-art algorithms.

1.5 Acknowledgments

Parts of this thesis rely on the FRGC database that was gratefully provided by the FRGC organizers [65]. Portions of the research in this work use the CAS-PEAL face database [23] controlled under the sponsorship of the Chinese National Hi-Tech Program and ISVISION Tech. Co. Ltd.

Chapter 2

Face Graph

One important step of image processing is to understand what is shown in the image. Since there already exists a nearly perfect system for image understanding – *the brain* – it seems to be a good idea to investigate how it processes (retinal) images. In 1962, Hubel and Wiesel [34] conducted experiments in cat brains. In the primary visual cortex they found *simple cells* that are activated when moving horizontal square waves or gratings with a specific spatial frequency are presented at a specific location in the receptive field, i. e., a certain spatial region in the retinal image of the cat.

Pollen and Ronner [68] found out that pairs of simple cells exist that share the same preferred spatial frequency. Usually, one of these cells has even and the other one has odd symmetry, shifted by a quarter cycle (90°) in the direction of the grating. Pollen and Ronner assumed that there are two more simple cells in 180° and 270° that complete the 360° cycle. Later, Pollen and Ronner [69] proposed a *complex cell* that takes the input of these four simple cells to generate a response to the same spatial frequency, independent of its current position in the receptive field, i. e., its phase shift. Hence, the complex cell is activated whenever at least one of the four connected simple cells activates, see [12] for a depiction of that idea.

Daugman [14] showed that Gabor wavelets model the responses of simple cells in a good approximation. He also proved [14] that two-dimensional Gabor wavelets are optimal in the sense of the general uncertainty relation between spatial and frequency information resolution.

Finally, Jones and Palmer [38] found simple and complex cells that are specialized for bars or gratings that are not horizontally arranged, but rotated in-plane by certain degrees. They [37] also translated these complex cells into Gabor wavelets with different scales, rotation angles, and aspect ratios. The parameters of the experimentally found Gabor wavelets sample quite densely around the center of the frequency domain, i. e., most of the Gabor wavelets prefer low frequencies, while just a few cells respond to high frequency gratings.

Cosine- and sine-based Gabor wavelets can be seen as representing simple cells with even and odd symmetry, respectively, with the odd Gabor wavelet being shifted 90° relative to the even one. The 180° and 270° simple

cell responses are simply modeled by negative responses of the two Gabor wavelets. Hence, one even/odd pair of Gabor wavelets, which can be united to a single complex-valued Gabor wavelet, is sufficient to model two pairs of simple cells. According to Wundrich *et al.* [101], there is "evidence that the magnitudes of the Gabor filter responses [...] are calculated by [...] complex cells", i. e., the complex-valued response of the Gabor wavelet can be seen as simple cell responses, while the absolute value of it stands for the complex cell response.

2.1 Wavelets

Before Gabor wavelets and the Gabor wavelet family are introduced, in this section a short theoretical introduction into wavelet theory is provided. Previous to the use of wavelets, *Fourier analysis* was employed to decompose signals $f(x)$ into waves of different frequencies ω with weights $\check{f}(\omega)$ by using the *Fourier transform* $\mathcal{F}(f)$:

$$[\mathcal{F}(f)](\omega) = \check{f}(\omega) = \int_{-\infty}^{\infty} f(x)\ \mathrm{e}^{-i\omega x}\, dx, \tag{2.1-1}$$

where $\check{f}(\omega)$ are the complex-valued coefficients of the waves:

$$\mathrm{e}^{i\omega x} = \cos(\omega x) + i \sin(\omega x)\ . \tag{2.1-2}$$

The aggregation of weights $\check{f}(\omega)$ is often called the *frequency domain* representation of the function $f(x)$.

The original signal $f(x)$ can easily be reconstructed from these coefficients by a weighted superimposition of the complex waves:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \check{f}(\omega)\ \mathrm{e}^{i\omega x}\, d\omega \tag{2.1-3}$$

This procedure is known as the *inverse Fourier transform*, usually denoted as $\mathcal{F}^{-1}\big(\check{f}\big)$. Note that there are different ways to write the Fourier transform, one having prefactor $(2\pi)^{-1/2}$ for both forward and inverse Fourier transform [6, 78], but to keep equations more simple, I use $(2\pi)^{-1}$ only in the inverse transform (cf. [81, 39]).

2.1.1 1D Wavelets

Two of the major issues of Fourier analysis are the unboundedness of the underlying sine and cosine functions and, following from that, the impossibility of exact spacial localization of single small sine oscillations in the signal $f(x)$ when looking at the transformed signal $\check{f}(\omega)$. Wavelets [49, 13] were developed to overcome this issue by incorporating location and frequency information. A *wavelet* χ is a real- or complex-valued function that has to satisfy at least two constraints [6, 78, 81], namely[1]:

$$\int_{-\infty}^{\infty} |\chi(x)|^2 \, dx < \infty \qquad \Leftrightarrow \qquad \chi \in \mathbf{L}^2 \tag{2.1-4}$$

$$C_\chi = \int_{-\infty}^{\infty} \frac{|\check{\chi}(\omega)|^2}{|\omega|} \, d\omega < \infty \qquad \Rightarrow \qquad \check{\chi}(0) = 0\,. \tag{2.1-5}$$

In order to fulfill Equations (2.1 – 4) and (2.1 – 5), the wavelet χ must be finite and have zero mean in spatial domain, respectively.

In opposition to Fourier analysis, wavelet analysis processes pairs (s, t) of parameters. The *wavelet family* $\chi_{s,t}$ is generated from the *mother wavelet* χ by translation with offset $t \in \mathbb{R}$ and dilation with scale factor $s \in \mathbb{R}\backslash\{0\}$ [6, 78, 81]:

$$\chi_{s,t}(x) = \frac{1}{\sqrt{|s|}} \chi\Big(\frac{x-t}{s}\Big)\,. \tag{2.1-6}$$

Negative scale factors $s < 0$ can be interpreted as a reflection of the wavelet at its offset point t. Moreover, the symmetry condition:

$$\chi_{\text{-}s,t}(x) = \chi_{s,\text{-}t}(\text{-}x) \tag{2.1-7}$$

holds for any wavelet χ. In frequency domain, the wavelet family is calculated similarly:

$$\check{\chi}_{s,t}(\omega) = \sqrt{|s|}\, \mathrm{e}^{-i\omega t}\, \check{\chi}(s\omega)\,. \tag{2.1-8}$$

The prefactor $|s|^{-\frac{1}{2}}$ in Equation (2.1 – 6) arises by tightening the postu-

[1] In literature there are two different versions of Equation (2.1 – 5). Blatter [6] and Schepper [78] multiply C_ψ by the 2π constant due to their different definition of the Fourier transform. In Equation (2.1 – 5), the definition of C_ψ from Sheng [81] is used.

lation of Equation (2.1 – 4):

$$\int_{-\infty}^{\infty} |\chi(x)|^2 \, dx = 1 \,, \tag{2.1-9}$$

so that the wavelet has unit second moment [6, 81]. It is easy to see that this factor is equaled out [6]:

$$\begin{aligned}
\int_{-\infty}^{\infty} |\chi_s(x)|^2 dx &= \int_{-\infty}^{\infty} \left| \frac{1}{\sqrt{|s|}} \chi\Big(\frac{x}{s}\Big) \right|^2 dx \\
&= \frac{1}{|s|} \int_{-\infty}^{\infty} \left| \chi\Big(\frac{x}{s}\Big) \right|^2 dx \\
&= \frac{1}{|s|} \int_{-\infty}^{\infty} \left| \chi\big(x'\big) \right|^2 dx' \, \frac{dx}{dx'} \,, \text{ where } x' = \frac{x}{|s|} \wedge \frac{dx}{dx'} = |s| \\
&= \frac{|s|}{|s|} \int_{-\infty}^{\infty} \left| \chi\big(x'\big) \right|^2 dx' = 1 \,.
\end{aligned}$$

The family of wavelets builds a set of functions that is used in a wavelet transform, similar to the sine/cosine waves with different frequencies ω being basis functions of the Fourier transform. But in opposition to sine functions, the one-dimensional signal $f(x)$ is split up into coefficients $f(s,t)$, which are called the *responses* of the function f to wavelet $\chi_{s,t}$, in the two dimensions s and t:

$$\begin{aligned}
f(s,t) &= \langle f, \chi_{s,t} \rangle \\
&= \int_{-\infty}^{\infty} f(x) \, \overline{\chi_{s,t}(x)} \, dx \\
&= \frac{1}{\sqrt{|s|}} \int_{-\infty}^{\infty} f(x) \, \overline{\chi\Big(\frac{x-t}{s}\Big)} \, dx \,,
\end{aligned} \tag{2.1-10}$$

the complex conjugation $\overline{\chi}$ of the wavelet is inserted following the definition of the complex valued scalar product $< f, \chi >$.

The wavelet transform yields coefficients $f(s,t)$ that are highly redundant and, thus, the reconstruction [6, 78] of the signal from the wavelet coefficients:

$$f(x) = \frac{1}{C_\chi} \int_{\mathbb{R}} \int_{\mathbb{R}\setminus\{0\}} f(s,t)\ \chi_{s,t}(x)\ \frac{dsdt}{|s|^2} + \text{const}\,, \qquad (2.1-11)$$

using C_χ from Equation (2.1 – 5) is over-determined to a very high degree. Therefore, it is sufficient to use only a subset of the wavelet family in the wavelet transform to process the signal, keeping the possibility of a (lossy) signal reconstruction. However, it is not possible to completely remove one of the two dimensions s or t. When the scale s is kept fixed, only signals of this scale can be reconstructed. When using a fixed translation t, the reconstruction of the signal far away from this offset position t becomes unreliable. Thus, wavelet subsets have to include different scales s as well as different offset positions t. Note that the additive constant in Equation (2.1 – 11) is included since the wavelets are zero-mean (cf. Equation (2.1 – 5)) and, hence, the mean value can not be reconstructed.

2.1.2 2D Wavelets

Since image processing handles two-dimensional signals, i. e., images $\mathcal{I}(\vec{x})$, there is also the need for two-dimensional mother wavelets $\chi(\vec{x})$. When the wavelet is not rotation-invariant, additional to scale s the rotation parameter ϑ is included. The resulting two-dimensional wavelet family [103, 78]:

$$\chi_{s,\vartheta,\vec{t}}(\vec{x}) = \frac{1}{s}\,\chi\left(\frac{Q(\vartheta)^{\mathrm{T}}\left(\vec{x}-\vec{t}\right)}{s}\right) \qquad (2.1-12)$$

is generated by scaling and rotating the mother wavelet, with the two-dimensional rotation matrix being defined as:

$$Q(\vartheta) = \begin{pmatrix} \cos(\vartheta) & -\sin(\vartheta) \\ \sin(\vartheta) & \cos(\vartheta) \end{pmatrix}. \qquad (2.1-13)$$

When the family is generated in frequency domain:

$$\check{\chi}_{s,\vartheta,\vec{t}}(\vec{\omega}) = s\,\check{\chi}\left(sQ(\vartheta)^{\mathrm{T}}\vec{\omega}\right)\ \mathrm{e}^{-i\,\vec{\omega}^{\mathrm{T}}\vec{t}}\,, \qquad (2.1-14)$$

the rotation affects only the frequency vector $\vec{\omega}$, which is accordingly rotated in the two-dimensional frequency space.

The two-dimensional wavelet transform now results in intrinsically four-dimensional coefficients:

$$\mathcal{T}\left(s,\vartheta,\vec{t}\right) = \int\limits_{\mathbb{R}^2} \mathcal{I}(\vec{x})\ \overline{\chi_{s,\vartheta,\vec{t}}(\vec{x})}\, d^2x\,. \qquad (2.1-15)$$

Also the reconstruction procedure has to be changed [103, 78] to:

$$\mathcal{I}(\vec{x}) = \frac{1}{C_\chi} \int\limits_{\mathbb{R}^+} \int\limits_{0}^{2\pi} \int\limits_{\mathbb{R}^2} \mathcal{T}\left(s,\vartheta,\vec{t}\right)\ \chi_{s,\vartheta,\vec{t}}(\vec{x})\ \frac{d^2t\, d\vartheta\, ds}{s^3} + \text{const}\,, \qquad (2.1-16)$$

with C_χ calculates as [78]:

$$C_\chi = \int\limits_{\mathbb{R}^2\setminus\{\vec{0}\}} \frac{|\check{\chi}(\vec{\omega})\,|^2}{|\vec{\omega}|^2}\, d^2\omega\,. \qquad (2.1-17)$$

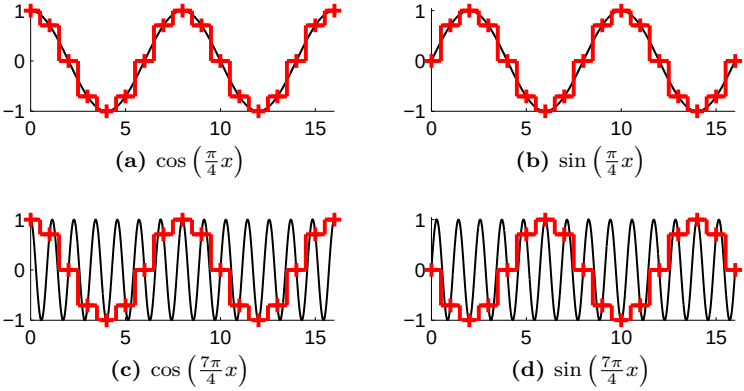
This reconstruction formula is picked up in Section 6.1.1, but beforehand the foundations of the Gabor wavelets and face graphs are set.

2.2 Image Processing with Gabor Wavelets

In the previous section, all parts of the wavelet transform are based on continuous signals and continuous wavelets. Since in image processing images $\mathcal{I}(\vec{x})$ are used that have discrete positions $\vec{x}$ in a finite support, i. e., $\vec{x}$ is limited to discrete positions: $\vec{x} \in \{0,\ldots,R_h\text{-}1\} \times \{0,\ldots,R_v\text{-}1\}$ with $\vec{R} = (R_h, R_v)^{\mathrm{T}}$ being the horizontal and vertical *resolution* of the image, the wavelets have to be discrete and finite, too. The discretization of the Fourier transform [39] and the wavelet transform [103] have been studied extensively and shall not be repeated in detail, but the results that are needed in this thesis are recapitulated.

2.2.1 Discrete Signals

One of the most important facts of the Fourier transform of discrete signals in a finite support is the circumstance that the frequency ω is limited in range: $\omega \in [\text{-}\pi, \pi[$. This issue can be deduced from the Nyquist-Shannon sampling theorem [61, 80], the impact of it is shown in Figure 2.1. Although



(a) $\cos\left(\frac{\pi}{4}x\right)$ **(b)** $\sin\left(\frac{\pi}{4}x\right)$

(c) $\cos\left(\frac{7\pi}{4}x\right)$ **(d)** $\sin\left(\frac{7\pi}{4}x\right)$

Figure 2.1: Under Sampling of Waves: *This figure shows sine and cosine waves in different wavelength, continuous and sampled at integral positions.*

the frequencies $\frac{\pi}{4}$ and $\frac{7\pi}{4}$ of the cosine waves in Figures 2.1(a) and 2.1(c), respectively, are very different in the continuous form (black line), they become identical when they are sampled at integral positions (red marks and steps). The sine waves given in Figures 2.1(b) and 2.1(d) show similar behavior, besides the sign being inverted. In general, the Fourier transform of a sampled signal shows a cyclic and half-symmetric structure, i. e., the frequency can be reduced modulo 2π, and the Fourier transform of negative frequencies can be calculated from the positive ones [39]:

$$\check{f}(\text{-}\omega) = \overline{\check{f}(\omega)}\,, \tag{2.2-1}$$

as long as the input signal f is real-valued. From this it also follows that the Fourier transform at $\omega = \pi$ is real valued, i. e., the imaginary part vanishes since $\check{f}(\pi) = \overline{\check{f}(\text{-}\pi)} = \overline{\check{f}(2\pi - \pi)} = \overline{\check{f}(\pi)}$.

Another point is that the number of linearly independent functions is identical to the number of discrete sampling points, the resolution R. Their discrete frequencies can easily be computed as:

$$\omega_n = \begin{cases} \frac{n\,2\pi}{R} & \text{for } n \in \{0, \ldots, \frac{R}{2}\text{-}1\} \\ \frac{(n-R)\,2\pi}{R} & \text{for } n \in \{\frac{R}{2}, \ldots, R\text{-}1\} \end{cases}\,, \tag{2.2-2}$$

keeping the central frequency $\omega_0 = 0$ at first position.

Finally, all properties of the one-dimensional discrete Fourier transform are also valid when dealing with two-dimensional images $\mathcal{I}(\vec{x})$. Hence, the frequencies $\vec{\omega} \in [-\pi, \pi[^2$ are limited in size and number, again ensuring $\vec{\omega}_0 = \vec{0}$. Furthermore, the symmetry condition $\check{\mathcal{I}}(-\vec{\omega}) = \overline{\check{\mathcal{I}}(\vec{\omega})}$ holds because real-valued images are processed.

2.2.2 Continuous Gabor Wavelet Family

Gabor wavelets are instantiable in continuous and in discrete environments, the main focus of this thesis is on discrete Gabor wavelets. Please note that there is a difference between *continuous wavelets* and the *continuous wavelet family*: On the one hand, continuous wavelets are wavelets in a continuous environment, i. e., $\vec{x}$ has real-valued components. On the other hand, for the continuous wavelet family, the wavelet parameters (k, ϑ) are continuous, while the positions $\vec{x}$ are integral.

The two-dimensional *mother Gabor wavelet*:

$$\psi(\vec{x}) = \frac{1}{\sigma^2} \, \mathrm{e}^{-\frac{\vec{x}^2}{2\sigma^2}} \, \mathrm{e}^{i \, \vec{e}_h^{\mathrm{T}} \vec{x}} \tag{2.2-3}$$

is a complex-valued filter that is divided into two overlaying parts. The first part is a Gaussian with standard deviation[2] σ, which makes the Gabor wavelet localized both in spatial and frequency domain. The second part is the complex-valued even wave $\mathrm{e}^{i \, \vec{e}_h^{\mathrm{T}} \vec{x}}$ with spatial frequency $\vec{e}_h = (1, 0)^{\mathrm{T}}$ pointing along the horizontal axis. The mother Gabor wavelet in frequency domain:

$$\check{\psi}(\vec{\omega}) = \mathrm{e}^{-\frac{\sigma^2 (\vec{\omega} - \vec{e}_h)^2}{2}} \, , \tag{2.2-4}$$

is just a Gaussian with its center moved to frequency coordinates $\vec{e}_h$.

Out of this mother Gabor wavelet, the *continuous Gabor wavelet family* in spatial domain [104][3]:

$$\psi_{k,\vartheta,\vec{t}}(\vec{x}) = k^2 \, \psi\big(k \, Q(\vartheta)^{\mathrm{T}} \, (\vec{x} - \vec{t})\big) \tag{2.2-5}$$

$$= \frac{k^2}{\sigma^2} \, \mathrm{e}^{-\frac{k^2 (\vec{x} - \vec{t})^2}{2\sigma^2}} \, \mathrm{e}^{i \, \vec{e}_h^{\mathrm{T}} \left(k \, Q(\vartheta)^{\mathrm{T}} (\vec{x} - \vec{t})\right)} \tag{2.2-6}$$

is generated. Comparing Equations (2.1 – 12) and (2.2 – 5), two major differences can be found: On the one hand, the scale factor s has changed to the

[2] There are also approaches like [9] that use non-uniform standard deviation matrix Σ in the two dimensions, but I use $\Sigma = \begin{pmatrix} \sigma & 0 \\ 0 & \sigma \end{pmatrix}$ throughout this thesis.

[3] The rotation matrix $Q(\vartheta)$ in [104] rotates into the wrong direction. Therefore, the transposition of it is not included in the equations of Wundrich [104].

center frequency k, which can be regarded as $k = 1/s$. On the other hand, the prefactor of the daughter wavelet changed from $1/s$ to k^2 and, thus, the postulation of the unitary second moment stated in Equation (2.1 – 9) is not fulfilled for Gabor wavelets. Nonetheless, prefactor k^2 has its right to exist. It is used since the image Fourier amplitudes $\check{\mathcal{I}}(\vec{\omega})$ of natural images decay like $1/|\vec{\omega}|$ and, hence, the Gabor wavelets in frequency domain should have norms proportional to k [105]. Although latest tests show that this is not necessarily true for facial images (cf. Appendix C), this prefactor is used throughout this thesis. In frequency domain, the prefactor of the *daughter Gabor wavelet*:

$$\check{\psi}_{k,\vartheta,\vec{t}}(\vec{\omega}) = \check{\psi}\Big(\frac{1}{k}\,Q(\vartheta)^{\mathrm{T}}\,\vec{\omega}\Big)\;\mathrm{e}^{\,i\,\vec{\omega}^{\mathrm{T}}\vec{t}} \tag{2.2-7}$$

is fixed at 1, unlike the value $\frac{1}{k}$ that would be required according to Equation (2.1 – 8).

To come to the Gabor wavelet formula most often used in literature [105, 10, 97, 20, 71, 103], some conversions have to be made. The first step is to remove the offset point $\vec{t}$. This can be achieved by not using the two-dimensional scalar product:

$$\langle \mathcal{I}, g_{\vec{t}} \rangle = \int \mathcal{I}(\vec{x})\;\overline{g_{\vec{t}}(\vec{x})}\,d^2x \tag{2.2-8}$$

to calculate wavelet responses, but the convolution:

$$(\mathcal{I} * g)\big(\vec{t}\big) = \int \mathcal{I}(\vec{x})\;g\big(\vec{t}-\vec{x}\big)\;d^2x\,, \tag{2.2-9}$$

which intrinsically includes the offset point. Since:

$$\psi_{k,\vartheta,\vec{t}}(\vec{x}) = \psi_{k,\vartheta,\vec{0}}\big(\vec{x}-\vec{t}\big) = \psi_{k,\vartheta}\big(\vec{x}-\vec{t}\big) \tag{2.2-10}$$

holds, $\psi_{k,\vartheta}\big(\vec{t}-\vec{x}\big)$ can be used in the convolution.

The second step changes the polar representation of the wave parameters, i. e., k and ϑ to Cartesian coordinates, forming the parameter vector:

$$\vec{k} = \begin{pmatrix} k_h \\ k_v \end{pmatrix} = k\,Q(\vartheta)\;\vec{e}_h = \begin{pmatrix} k\,\cos(\vartheta) \\ k\,\sin(\vartheta) \end{pmatrix}\,. \tag{2.2-11}$$

Please note that the rotation direction of the rotation matrix has changed, see Appendix A.3 for an explanation.

The last change accounts for the fact that the Gabor wavelet as given in Equation (2.2 – 13) is not mean free because the mean value, which is stored

in $\check{\psi}(\vec{\omega}_0)$, is non-zero. Thus, the Gabor wavelet can not be used in a wavelet transform because the admissibility condition from Equation (2.1 – 5) is not fulfilled. Würtz [105] introduced a correction term that subtracts a Gaussian with the maximum value similar to $\check{\psi}(\vec{\omega}_0) = \mathrm{e}^{-\frac{\sigma^2}{2}}$ at the frequency domain center $\vec{\omega}_0 = \vec{0}$.

Including all these changes, the Gabor wavelet in spatial domain can be written as [105]:

$$\psi_{\vec{k}}(\vec{x}) = \frac{\vec{k}^2}{\sigma^2} \mathrm{e}^{-\frac{\vec{k}^2\,\vec{x}^2}{2\sigma^2}} \left[\mathrm{e}^{i\,\vec{k}^{\mathrm{T}}\vec{x}} - \mathrm{e}^{-\frac{\sigma^2}{2}} \right] , \tag{2.2 – 12}$$

and the frequency domain formula now reads as:

$$\begin{aligned} \check{\psi}_{\vec{k}}(\vec{\omega}) &= \mathrm{e}^{-\frac{\sigma^2\left(\vec{\omega}-\vec{k}\right)^2}{2\vec{k}^2}} - \mathrm{e}^{-\frac{\sigma^2}{2}}\,\mathrm{e}^{-\frac{\sigma^2\,\vec{\omega}^2}{2\vec{k}^2}} \\ &= \mathrm{e}^{-\frac{\sigma^2\left(\vec{\omega}-\vec{k}\right)^2}{2\vec{k}^2}} - \mathrm{e}^{-\frac{\sigma^2\left(\vec{\omega}^2+\vec{k}^2\right)}{2\vec{k}^2}} . \end{aligned} \tag{2.2 – 13}$$

One important attribute of the Gabor wavelet is its strongly symmetric nature, both in spatial domain:

$$\psi_{\vec{k}}(-\vec{x}) = \psi_{-\vec{k}}(\vec{x}) = \overline{\psi_{\vec{k}}(\vec{x})} , \tag{2.2 – 14}$$

as well as in frequency domain:

$$\check{\psi}_{\vec{k}}(-\vec{\omega}) = \check{\psi}_{-\vec{k}}(\vec{\omega}) . \tag{2.2 – 15}$$

The symmetry condition from Equation (2.2 – 14) is obvious since in the spatial Gabor wavelet given in Equation (2.2 – 12), only the phase shift $\mathrm{e}^{i\,\vec{k}^{\mathrm{T}}\vec{x}}$ is affected by the sign change, i. e., only the sign of the imaginary part, which corresponds to the axially symmetric sine part of the complex wave, is negated. The symmetry condition stated in Equation (2.2 – 15) is even more obvious when having a look at the roles of $\vec{\omega}$ and $\vec{k}$ in Equation (2.2 – 13).

2.2.3 Discrete Gabor Wavelet Family

The number of Gabor wavelets in the continuous family is infinite and covering the whole frequency domain, besides the frequency domain center $\vec{\omega}_0$. Of course, it is impossible to use all emerging Gabor wavelets and, hence, a selection of Gabor wavelets has to be chosen. The *discrete Gabor wavelet*

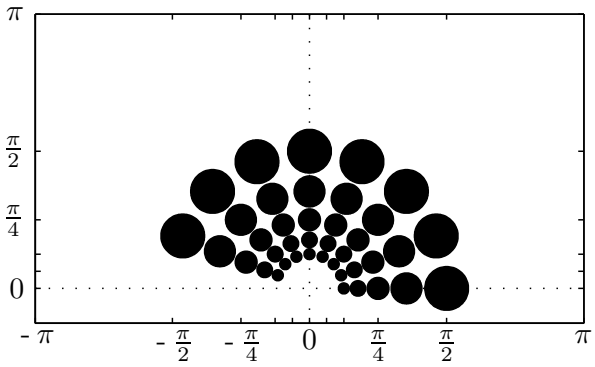


Figure 2.2: Discrete Gabor Wavelet Family: *This figure displays the common discrete Gabor wavelet family* Γ *in frequency domain, divided into* $\zeta_{max} = 5$ *scale levels and* $\nu_{max} = 8$ *directions. Each Gabor wavelet* $\check{\psi}_{\vec{k}_j}$ *is indicated by a circle with radius* $\frac{1}{\sigma_{eff}}$ *centered at* $\vec{k}_j$.

family that is used in common face [20, 97] and object [94] recognition systems is a discretized regular subset of all possible frequencies k and rotation angles φ. It is defined by the parameter set:

$$\Gamma = (\nu_{\max}, \zeta_{\max}, k_{\max}, k_{\text{fac}}, \sigma) \tag{2.2–16}$$

that generates Gabor wavelets in $\nu_{\max} = 8$ directions and $\zeta_{\max} = 5$ scale levels. The discretization of angles is linear as given in Equation (2.2–17), while the center frequencies are scaled logarithmically (Equation (2.2–18)). The highest frequency $k_{\max} = \frac{\pi}{2}$ and the logarithmic distance between two scale levels $k_{\text{fac}} = 2^{\frac{1}{2}}$ are set according to Lades *et al.* [43]. To increase legibility, I measure rotation angles in degrees, while frequencies and phases are reported in radians.

$$\vartheta_\nu = \frac{\nu\, 180^\circ}{\nu_{\max}} \qquad \nu = \{0, \ldots, \nu_{\max} - 1\} \tag{2.2–17}$$

$$k_\zeta = k_{\max}\, k_{\text{fac}}^{\zeta} \qquad \zeta = \{0, \ldots, \zeta_{\max} - 1\} \tag{2.2–18}$$

The value $\sigma = 2\pi$, which was first used by Buhmann *et al.* [10], fixes the number of oscillations of the Gabor wavelet such that the wavelength of the

wave is identical to the effective standard deviation $\sigma_{\text{eff}} = \frac{\sigma}{k}$ of the enveloping Gaussian. The wavelengths of the Gabor wavelets can easily be computed using the center frequencies k_ζ. For the highest frequency $k_0 = k_{\max}$, the wavelength is $\frac{2\pi}{k_0} = 4$ pixel, while the largest wavelength, which belongs to the lowest frequency Gabor wavelets, is $\frac{2\pi}{k_4} = 16$ pixel. Furthermore, for $\sigma = 2\pi$ the mean-free correction term (cf. Equation (2.2 – 12)) of the Gabor wavelets is negligible since $\mathrm{e}^{-\frac{\sigma^2}{2}} \approx 10^{-9}$ vanishes.

The parametrization engendered by the common parameter set Γ is designed in a way that a sub-band in frequency domain is evenly filled, Figure 2.2 displays the centers and the effective standard deviations of the Gabor wavelets of this family. It is sufficient to cover only one half of the frequency domain with Gabor wavelets because images are real-valued and, thus, the second half of the frequency domain does not include new information (cf. Sections 2.2.1 and 2.2.4).

For the following sections, it is superfluous to carry two indexes ζ and ν to enumerate the center frequencies. Thus, they are converted into the one-dimensional index:

$$j = \zeta\, \nu_{\max} + \nu\,, \qquad j \in \{0, \ldots, J\text{-}1\}\,, \tag{2.2 – 19}$$

where $J = \zeta_{\max}\, \nu_{\max}$ is the number of Gabor wavelets generated with parameter set Γ. The resulting Gabor wavelet is tagged with $\psi_{\vec{k}_j}$.

2.2.4 Gabor Wavelet Transform

The discrete family of Gabor wavelets is used to process image $\mathcal{I}$, which for the objective of face detection and recognition shows a face and a more or less cluttered background. From this image, the so-called *Gabor transformed image* $\mathcal{T}$ is generated that consists of J layers $\mathcal{T}_{\vec{k}_j}$. Each of these layers is the result of the convolution of input image $\mathcal{I}$ with the corresponding Gabor wavelet $\psi_{\vec{k}_j}$:

$$\begin{aligned} \mathcal{T}_{\vec{k}_j}\left(\vec{t}\right) &= \left(\psi_{\vec{k}_j} * \mathcal{I}\right)\left(\vec{t}\right) \\ &= \sum_{\vec{x}} \psi_{\vec{k}_j}\left(\vec{t} - \vec{x}\right)\, \mathcal{I}(\vec{x}) \\ &= \sum_{\vec{x}} \overline{\psi_{\vec{k}_j}\left(\vec{x} - \vec{t}\right)}\, \mathcal{I}(\vec{x})\,, \end{aligned} \tag{2.2 – 20}$$

or, alternatively, the Fourier transform of $\mathcal{I}$ to frequency domain, the successive multiplication of $\check{\mathcal{I}}$ with $\check{\psi}_{\vec{k}_j}$ pixel by pixel:

$$\check{\mathcal{T}}_{\vec{k}_j}(\vec{\omega}) = \check{\psi}_{\vec{k}_j}(\vec{\omega})\ \check{\mathcal{I}}(\vec{\omega})\ , \qquad (2.2-21)$$

and the subsequent inverse Fourier transform of $\check{\mathcal{T}}_{\vec{k}_j}$ to spatial domain. Hence, the Gabor transformed image layer $\mathcal{T}_{\vec{k}_j}$ has the same resolution as the input image $\mathcal{I}$, but the pixels $\mathcal{T}_{\vec{k}_j}\left(\vec{t}\right)$, i. e., the responses of Gabor wavelet $\psi_{\vec{k}_j}$ to image $\mathcal{I}$ at offset point $\vec{t}$ are complex-valued. Since most parts of $\check{\psi}(\vec{\omega})$ are negligible, the multiplication can be sped up by filling $\check{\mathcal{T}}(\vec{\omega})$ with 0 at those positions.

When the *Gabor wavelet transform* (GWT) is executed in spatial domain, as given in Equation (2.2 – 20), the resolutions of the image and the Gabor wavelet may differ, and the sum is executed over the smaller resolution of both elements, which is usually the one from the Gabor wavelet. In practical applications, where the GWT is actually calculated in spatial domain, the resolutions of the Gabor wavelets are most often limited to cut off pixels that are away from the center more than $3-5\ \sigma_{\text{eff}}$, i. e., where the enveloping Gaussian is negligible. If the overlay of Gabor wavelet and image extends over the border of the image, commonly convolution with cyclic boundary conditions is applied.

The essential motivation for using frequency domain convolution is that responses for all positions $\vec{t}$ of input image $\mathcal{I}$ are available at once, whereas convolution in spatial domain has to be applied for each offset point separately. When the GWT is executed in frequency domain, the resolutions of Gabor wavelet and image have to be identical. Since Gabor wavelets have fixed spatial extents (see Section 2.3.2 for the impact of this fact), they are scaled in frequency domain. Recalling the discretization of $\vec{\omega}$ from Section 2.2.1, this implies that the circles shown in Figure 2.2 degenerate to ellipses if the image resolution is not square.

The strongly symmetric nature of the Gabor wavelet also extends to the Gabor transformed image, i. e., it is possible to compute the result $\mathcal{T}_{-\vec{k}_j}$ of the convolution of $\mathcal{I}$ with the left-out Gabor wavelets $\psi_{-\vec{k}_j}$ from the corresponding Gabor wavelet response:

$$\mathcal{T}_{-\vec{k}_j}\left(\vec{t}\right) = \overline{\mathcal{T}_{\vec{k}_j}\left(\vec{t}\right)}\,, \qquad \check{\mathcal{T}}_{-\vec{k}_j}(\vec{\omega}) = \overline{\check{\mathcal{T}}_{\vec{k}_j}(-\vec{\omega})}\,, \qquad (2.2-22)$$

both in spatial and in frequency domain.

2.3 Gabor Jet

The information in the Gabor transformed image $\mathcal{T}$ is redundant to a very high degree. For each real-valued pixel of the image $\mathcal{I}$, the corresponding Gabor transformed image $\mathcal{T}$ is composed of $J = 40$ complex values. This means that neighboring pixels share about the same information and, hence, can be approximated (for the approximation of Gabor wavelet responses from surrounding positions, see Section 6.3). Therefore, as already addressed in Section 2.1, only responses of some Gabor wavelets $\psi_{\vec{k}_j}$ at a couple of positions $\vec{t}$ of the Gabor transformed image need to be preserved. To unify the selection process of positions $\vec{t}$ and responses of Gabor wavelets $\psi_{\vec{k}_j}$, Buhmann *et al.* [10] introduced the concept of a *Gabor jet* $\mathcal{J}$. Although they used a different Gabor wavelet family (a different Γ), the idea of the Gabor jet, a simplified depiction of which is displayed in Figure 2.7(a) on page 32, has not changed.

A Gabor jet $\mathcal{J}^{\mathcal{I}}$ of image $\mathcal{I}$ is generated from the Gabor transformed image $\mathcal{T}$ by stringing together the responses of all Gabor wavelets at a specific position $\vec{t}$ into one vector. The elements of $\mathcal{J}^{\mathcal{I}}$ are addressed by the index j of the corresponding Gabor wavelet $\psi_{\vec{k}_j}$:

$$\Big(\mathcal{J}^{\mathcal{I}}\big(\vec{t}\big)\Big)_j = \mathcal{T}_{\vec{k}_j}\big(\vec{t}\big)\ . \tag{2.3-1}$$

Thus, the Gabor jet is an aggregation of all Gabor wavelet responses at the position $\vec{t}$, incorporating one single complex value $(\mathcal{J})_j$ for each Gabor wavelet shown in Figure 2.2.

The representations of the complex-valued Gabor wavelet responses are changed from algebraic to polar description:

$$\big(\mathcal{J}\big)_j = a_j\, \mathrm{e}^{\,i\,\phi_j}\ , \tag{2.3-2}$$

with:

$$a_j = \Big|\big(\mathcal{J}\big)_j\Big|, \qquad\qquad \phi_j = \arg\Big[\big(\mathcal{J}\big)_j\Big]\,, \tag{2.3-3}$$

where each a_j can be seen as the response of a different complex cell [101]. This step has some major advantages:

1. The absolute values are relatively stable around the offset point. For small displacements $\vec{d}$, the absolute values are similar:

$$\left|\Big(\mathcal{J}^{\mathcal{I}}\big(\vec{t}\big)\Big)_j\right| \approx \left|\Big(\mathcal{J}^{\mathcal{I}}\big(\vec{t}+\vec{d}\big)\Big)_j\right|. \tag{2.3-4}$$

2. Local contrast normalization of the texture information can be applied by dividing the Gabor wavelet responses by the norm of the Gabor jet:

$$a_j := \frac{a_j}{\|\mathcal{J}\|} = \frac{a_j}{\sqrt{\sum_{j'=0}^{J-1} a_{j'}^2}} \,. \tag{2.3-5}$$

Obviously, only the absolute values are affected by the normalization. The normalized Gabor jet is, to some extent, independent of lighting conditions and camera parameters, but normalizing Gabor jets extracted at homogeneous, e. g., background locations, where the responses of all Gabor wavelets are low, may lead to unintended behavior.

3. Phase differences can be estimated from displacements [88]:

$$\Delta\phi_j = \arg\left[\left(\mathcal{J}^{\mathcal{I}}\left(\vec{t}\right)\right)_j\right] - \arg\left[\left(\mathcal{J}^{\mathcal{I}}\left(\vec{t}+\vec{d}\right)\right)_j\right] \approx \vec{k}_j^{\mathrm{T}}\vec{d} \bmod 2\pi \,, \tag{2.3-6}$$

with the real-valued modulo function:

$$a \bmod b = \frac{a}{b} - \left\lfloor \frac{a}{b} \right\rfloor \tag{2.3-7}$$

reducing $\vec{k}_j^{\mathrm{T}}\vec{d}$ into $[0, 2\pi[$. This fact is used in the reconstruction of images (see Section 6.3) to approximate the phases of Gabor wavelet responses at missing positions. Note that this estimation only works when the absolute values are high and, thus, no phase jumps occur.

4. The displacement can be estimated using the phase differences of all wavelet responses by calculating the projections $\vec{p}_j$ in the direction of the wave vectors [88]:

$$\vec{p}_j = \frac{\vec{k}_j^{\mathrm{T}}\vec{d}}{\vec{k}_j^2}\,\vec{k}_j \approx \frac{\Delta\phi_j}{\vec{k}_j^2}\,\vec{k}_j \,. \tag{2.3-8}$$

Each of these projections $\vec{p}_j$ suggests a straight line $\beta_j\vec{p}_j^{\perp}$ perpendicular to the wave direction of $\vec{k}_j$. Ideally, all these lines meet at the same vertex $\vec{d}$. Theimer and Mallot [88] show a comprehensive depiction of this idea. Hence, $\vec{d}$ can be calculated by solving the over-determined system of linear equations:

$$\vec{d} = \sum_j \left(\vec{p}_j + \beta_j\,\vec{p}_j^{\perp}\right) \,. \tag{2.3-9}$$

To overcome the problem of noisy and imperfect data, e. g., when the absolute values a_j are low, the displacement can be approximated by minimizing an error function [88] or the Taylor expansion of a similarity measure [98]. The latter one is illustrated in more detail in Appendix B.

A Gabor jet codes the local texture of the image around the offset point $\vec{t}$, where it is extracted. While the responses of the high frequency Gabor wavelets are only including the information from some pixels around $\vec{t}$, the low frequency Gabor wavelets store more general information about the broader surrounding area. In conclusion, different information is embraced by the Gabor jet. Usually, this fact is neglected and the Gabor jet is taken as a vector of real or complex values that stores local texture, treating each component of the Gabor jet identically. This approach is appropriated by most parts of this work, but in some others (cf. Chapter 6 and Appendix B), the creation of the Gabor jet plays a role.

2.3.1 Gabor Jet Similarities

The texture stored in the Gabor jet is used for various applications like face detection and recognition, which are addressed in Chapters 3 and 4, respectively. Most of these applications need to compare Gabor jets from different images, i. e., compute a similarity measure between two Gabor jets. The similarity function $S_{[A]}$ that was introduced[4] by Buhmann *et al.* [10] calculates the similarity between two Gabor jets $\mathcal{J}$ and $\mathcal{J}'$ as the normalized scalar product, i. e., the cosine of the angle between the vectors of absolute values of them:

$$\begin{aligned} S_{[A]}\big(\mathcal{J},\mathcal{J}'\big) &= \cos\angle\big(\mathcal{J},\mathcal{J}'\big) \\ &= \frac{\langle\mathcal{J},\mathcal{J}'\rangle}{\|\mathcal{J}\|\,\|\mathcal{J}'\|} \\ &= \frac{\sum_{j=0}^{J-1} a_j\, a'_j}{\sqrt{\left(\sum_{j=0}^{J-1} a_j^2\right)\left(\sum_{j=0}^{J-1} a_j'^2\right)}}\,. \end{aligned} \tag{2.3-10}$$

This similarity measure implicitly implements the Gabor jet normalization. When normalized Gabor jets are used in the calculation of $S_{[A]}$ and, hence,

[4] Buhmann *et al.* [10] and others [97, 20] called this function S_a, but to be consistent, in this work capital characters are used for Gabor jet similarities, instead. Furthermore, Buhmann *et al.* [10] added some more terms that deal with the length of the Gabor jets, which are usually left out [97, 20].

the norms of the $\mathcal{J}$ and $\mathcal{J}'$ are unity, the denominator of Equation (2.3 – 10) vanishes. This fact can be used to compute similarities between Gabor jets more efficiently, especially when Gabor jet similarities are needed repeatedly.

$S_{[A]}$ only addresses the absolute values a_j, which are relatively stable some pixels around the offset point. Hence, $S_{[A]}$ similarities are usually smooth in terms of displacements $\vec{d}$ (cf. Equation (2.3 – 4)). In opposition, the phases ϕ_j are oscillating very fast when displacements occur. To use this fact, the $S_{[P]}$ similarity measure:

$$S_{[P]}\left(\mathcal{J},\mathcal{J}'\right) = \frac{\sum_{j=0}^{J-1} a_j\, a'_j\, \cos\left(\phi_j - \phi'_j\right)}{\sqrt{\left(\sum_{j=0}^{J-1} a_j^2\right)\left(\sum_{j=0}^{J-1} a_j'^2\right)}} \tag{2.3 – 11}$$

was added by Wiskott *et al.* [98], who used the modified variant:

$$S_{[D]}\left(\mathcal{J},\mathcal{J}'\right) = \frac{\sum_{j=0}^{J-1} a_j\, a'_j\, \cos\left(\phi_j - \phi'_j - \vec{k}_j^{\mathrm{T}}\,\vec{d}\right)}{\sqrt{\left(\sum_{j=0}^{J-1} a_j^2\right)\left(\sum_{j=0}^{J-1} a_j'^2\right)}}\,. \tag{2.3 – 12}$$

for the estimation of $\vec{d}$, see Appendix B for details. Phase differences play an important role in Equations (2.3 – 11) and (2.3 – 12). Therefore, both can be used to solve the correspondence problem (cf. Section 3.2.2). On the one hand, even slight displacements vary the $S_{[P]}$ similarity strongly and, thus, only correct positions have high $S_{[P]}$ similarity values. On the other hand, the displacement $\vec{d}$ estimated by $S_{[D]}$ directly points to the correct position if $\vec{d}$ is sufficiently short. Usually, these two similarity functions are not used for face recognition, but experiments (cf. Section 5.4) show that they are well suited for classification.

A further comparison function between two Gabor jets is the Canberra distance:

$$D_{[C]}\left(\mathcal{J},\mathcal{J}'\right) = \frac{1}{J}\sum_{j=0}^{J-1}\begin{cases} \frac{|a_j - a'_j|}{a_j + a'_j} & \text{, if } \left(a_j + a'_j\right) > \epsilon \\ 0 & \text{, else} \end{cases}\,. \tag{2.3 – 13}$$

This function again uses only the absolute values of the normalized Gabor jets, emphasizing the low-valued elements of them. The results of $D_{[C]}$ are

bound between 0 and 1 since all terms $\frac{|a_j - a'_j|}{a_j + a'_j}$ lie in that interval. Specifically, the term evaluates to 0 when both elements a_j and a'_j are equal, and 1 if one (and only one) of the two elements is zero. The conversion of the Canberra distance metric $D_{[C]}$ to the Canberra similarity: $S_{[C]}(\mathcal{J}, \mathcal{J}') = 1 - D_{[C]}(\mathcal{J}, \mathcal{J}')$ is straightforward.

Additional similarity and distance functions are present in literature. Jiménez *et al.* [36] tested a couple of distance functions including different normalizations and some image resolutions. The function that yielded the best results in their experiments is the modified Manhattan distance:

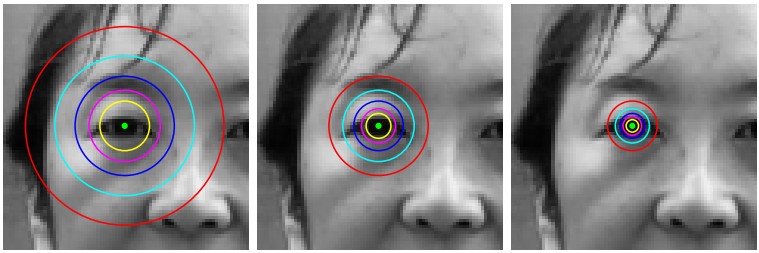
$$D_{[M]}\big(\mathcal{J}, \mathcal{J}'\big) = \frac{\sum_{j=0}^{J-1} |a_j - a'_j|}{\left(\sum_{j=0}^{J-1} |a_j|\right)\left(\sum_{j=0}^{J-1} |a'_j|\right)}, \tag{2.3-14}$$

applied to normalized Gabor jets (cf. Equation (2.3–5)). Hence, $D_{[M]}$ uses two different kinds of normalization, but leaving out one of them decreases recognition accuracy [36]. In turn, the modified Manhattan distance can be transformed to the modified Manhattan similarity function: $S_{[M]}(\mathcal{J}, \mathcal{J}') = 1 - D_{[M]}(\mathcal{J}, \mathcal{J}')$.

2.3.2 Image Resolution

Gabor wavelets have a fixed region in spatial domain, i. e., a certain range with pixel as unit of measurement. This range is independent of the actual image resolution and defined by the Gaussian envelope of the Gabor wavelet $\psi_{\vec{k}_j}$. The Gabor jet $\mathcal{J}^{\mathcal{I}}\big(\vec{t}\big)$ describes the texture around offset point $\vec{t}$. More specifically, the responses of the high frequency Gabor wavelets code the information that is directly next to the offset point, whereas the responses of the low frequency Gabor wavelets reach roughly 32 pixel from $\vec{t}$, i. e., twice the wavelength of the lowest frequency Gabor wavelet (cf. Section 2.2.3). Therefore, the responses of the Gabor wavelets and, consequently, the texture that is coded in the Gabor jet highly depend on the resolution of the image if the faces in the images have the same relative sizes.

How resolution impacts the information stored in the right eye Gabor jet is visualized in Figure 2.3, where the size of the Gabor jet in relation to the image resolution is sketched. The Gabor jet with its center at the green offset point is depicted as five circles that describe the scale levels of the Gabor wavelets. Each circle depicts the envelope of the Gabor wavelet with one standard deviation σ_{eff}. For example, at scale level $\zeta = 0$, the yellow circles

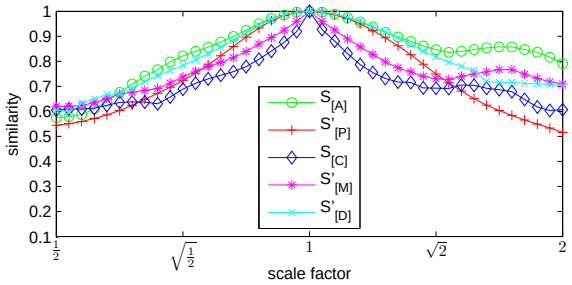


(a) 39 × 39 pixel or $k_{\max} = \frac{\pi}{4}$ **(b)** 79 × 79 pixel or $k_{\max} = \frac{\pi}{2}$ **(c)** 159 × 159 pixel or $k_{\max} = \pi$

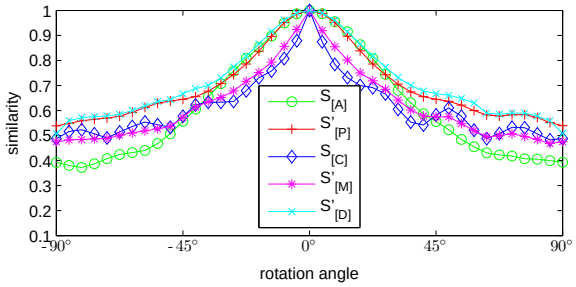
Figure 2.3: Impact of Image Resolution on Gabor Jet: *This figure gives a feeling for the information stored in a Gabor jet depending on the image resolution. The colored circles depict the standard deviations σ_{eff} of Gabor wavelets for scale levels $\zeta = 0$ (yellow) to $\zeta = 4$ (red). The sub-captions include two different ways of reaching this effect: modifying image resolutions or changing k_{max}.*

adumbrate $\sigma_{\text{eff}} = 4$ pixel, whereas the red circles show the largest Gabor wavelets, i. e., with $\zeta = 4$ and the according radius $\sigma_{\text{eff}} = 16$ pixel. When the resolution is low, e. g., such that the part of the face shown in Figure 2.3(a) fits into a 39 × 39 pixel area, the Gabor jet taken at the eye center encloses not only the eye, but also includes the eyebrow and at least parts of the nose. The more the resolution increases (cf. Figures 2.3(b) and 2.3(c)), the smaller the facial area captured by the Gabor jet becomes. The Gabor jet taken at the same landmark position only includes the texture of the iris and parts of the eyelid (cf Figure 2.3(c)), when the image resolution increases by a factor of four compared to Figure 2.3(a). In his BSc thesis [30], Haufe showed that the same effect can be generated by scaling the $k_{\max}$ value of the Gabor wavelet family. Hence, the sub-captions in Figure 2.3 show both the resolution of the image patch and the $k_{\max}$ value, taking Figure 2.3(b) as reference.

Since the encoded texture is different, Gabor jet similarities are very sensitive to scale and rotation variations in the underlying texture patch. To clarify this dependency, an experiment is conducted. The Gabor jet extracted at the eye center shown in Figure 2.3(b) is compared to the Gabor jet at the same location in scaled or rotated versions of the image patch. To avoid rounding issues in the calculation of the transformed offset point, the affine image transformation was applied using the offset point as transformation



(a) Scale Impact



(b) Angle Impact

Figure 2.4: Similarities of Scaled or Rotated Gabor Jets: *This figure shows the similarities of Gabor jets taken at the same position (a) in different image resolutions and (b) with different rotation angles.*

center. Therefore, the position is as exact as possible, but bi-linearly scaling the image might have introduced some glaze.

Figure 2.4 shows the results of that experiment. Clearly, all similarities drop when the scale or the rotation angle of the texture change. In particular, $S_{[C]}$ and $S_{[M]}$ react critically even to small changes. Noting that the expected $S_{[A]}$ similarity of two Gabor jets with completely random elements is 0.75, the similarity values shown in Figure 2.4 are astonishing. To overcome the problem of comparing Gabor jets in different scales, an easy and fast way of scaling and rotating Gabor jets is introduced in Section 3.4. In general, similarity values of different similarity functions are not comparable, i. e., higher similarity values do not imply that this function is better suited for face detection or recognition. For the sake of visualization, the $S_{[P]}$, $S_{[D]}$, and $S_{[M]}$

(a) Label **(b)** Probe Image **(c)** Similarity Map

Figure 2.5: Gabor Jet Similarity Map: *This figure displays (c) a map of $S_{[A]}$-similarities when comparing two Gabor jets, similarity values increase from 0.19 (black) to 0.92 (white). The similarities are calculated between the Gabor jet that is taken at the marked position from (a) and the Gabor jets extracted at each position of the probe image that is shown in (b).*

similarity values shown in Figure 2.4 are rescaled to: $S'_{[P]} = \frac{1}{2}\left(S_{[P]} + 1\right)$, $S'_{[D]} = \frac{1}{2}\left(S_{[D]} + 1\right)$, and $S'_{[M]} = 1 - 2D_{[M]}$, respectively.

2.4 Graphs Labeled with Gabor Jets

A single Gabor jet codes only the texture of a small image patch and cannot reliably be located in a novel *probe image* $\mathcal{I}$. An exemplary similarity map of a single Gabor jet $\mathcal{J}$ is shown in Figure 2.5(c). The similarities $S(\mathcal{J}, \mathcal{I})$ were calculated between the Gabor jet that was extracted at the marked position in Figure 2.5(a) and the Gabor jets of every position $\vec{t}$ of the probe image shown in Figure 2.5(b):

$$S(\mathcal{J}, \mathcal{I})\left(\vec{t}\right) = S_{[A]}\left(\mathcal{J}, \mathcal{J}^{\mathcal{I}}\left(\vec{t}\right)\right) . \qquad (2.4\text{–}1)$$

Obviously, the "eye" Gabor jet is not only found at the proper location, but, e. g., also at the mouth corner of the probe image. Furthermore, the similarity between the eye and the background is high at some positions. This accounts for the fact that the Gabor jet normalization boosts the noise stored in the Gabor jets taken from those regions.

2.4.1 Face Graph

By contrast, a combination of many Gabor jets at different positions is more likely to be found if the relative distances are kept fixed. These positions and their corresponding Gabor jets are assembled to a *face graph* [42] or *model graph* $\mathcal{G}^{\mathcal{M}} = \left(\vec{\mathcal{J}}^{\mathcal{M}}, \vec{\mathcal{L}}^{\mathcal{M}}, \vec{\mathcal{E}}\right)$, where $\vec{\mathcal{J}}^{\mathcal{M}}$ are the Gabor jets that are extracted at the *landmark positions* $\vec{\mathcal{L}}^{\mathcal{M}}$:

$$\forall l \in \{0, \ldots, L\text{-}1\} \quad : \quad \mathcal{J}_l^{\mathcal{M}} = \mathcal{J}^{\mathcal{I}}\left(\mathcal{L}_l^{\mathcal{M}}\right) . \tag{2.4–2}$$

The *edges* $\vec{\mathcal{E}}$ connect neighboring landmarks. The number of Gabor jets, which is equal to the number of landmarks, is denoted by L and the number of edges is marked with E. A reduced depiction of a model graph is displayed in Figure 2.7(b), whereas an example of a hand-labeled face graph can be found in Figure 2.6(a).

The model graph embeds two different types of information. The first type is the texture, which is coded in the Gabor jets $\vec{\mathcal{J}}^{\mathcal{M}}$. This texture can be used for face detection, but also codes the identity of the person shown in the image. The second type, which is disregarded by most recognition algorithms, is the geometry, i. e., the landmark positions $\vec{\mathcal{L}}^{\mathcal{M}}$ and the edges $\vec{\mathcal{E}} \subset \{0, \ldots, L\text{-}1\}^2$ defined by their position vectors:

$$\vec{\Delta}_e = \mathcal{L}_{e_2} - \mathcal{L}_{e_1}, \tag{2.4–3}$$

with e_1 and e_2 being the indexes of the two nodes linked by the edge $\mathcal{E}_e$. These relative positions are very important for face detection, e. g., in an upright facial image it is impossible that the eye landmarks can be found beneath the mouth.

During the *graph matching* procedure, the *graph similarity*:

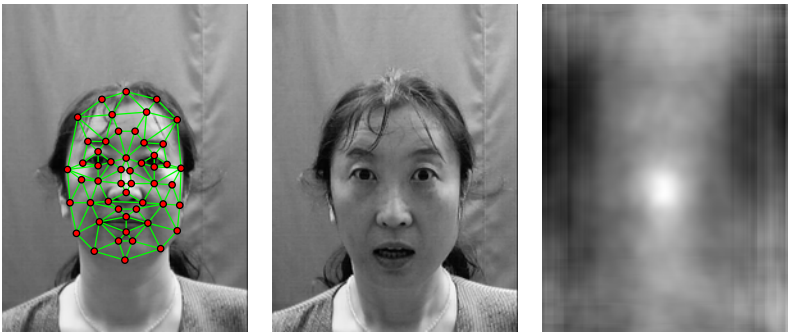
$$S_{[\cdot]}^{\mathcal{G}}\left(\mathcal{G}^{\mathcal{M}}, \mathcal{G}^{\mathcal{I}}\right) = \frac{1}{L} \sum_{l=0}^{L\text{-}1} S_{[\cdot]}\left(\mathcal{J}_l^{\mathcal{M}}, \mathcal{J}_l^{\mathcal{I}}\right) , \tag{2.4–4}$$

with:

$$S_{[\cdot]} \in \{S_{[A]}, S_{[P]}, S_{[D]}, S_{[C]}, S_{[M]}, \ldots\} . \tag{2.4–5}$$

is calculated between the model graph $\mathcal{G}^{\mathcal{M}}$ and *image graph* $\mathcal{G}^{\mathcal{I}}$. The Gabor jet $\mathcal{J}_l^{\mathcal{I}}\left(\vec{t}\right)$ of image graph $\mathcal{G}^{\mathcal{I}}\left(\vec{t}\right)$ is extracted at the corresponding landmark position $\mathcal{L}_l^{\mathcal{M}}$, moved according to the offset point $\vec{t}$:

$$\mathcal{L}_l^{\mathcal{I}}\left(\vec{t}\right) = \mathcal{L}_l^{\mathcal{M}} - \mathcal{C}^{\mathcal{M}} + \vec{t}, \tag{2.4–6}$$



(a) Model Graph **(b)** Probe Image **(c)** Similarity Map

Figure 2.6: Model Graph Similarity Map: *This figure displays (c) a map of $S^{\mathcal{G}}_{[A]}$-similarities comparing the model graph $\mathcal{G}^{\mathcal{M}}$ shown in (a) and image graphs $\mathcal{G}^{\mathcal{I}}\left(\vec{t}\right)$ extracted from the probe image $\mathcal{I}$ shown in (b) at each offset point $\vec{t}$. Similarity values in (c) increase from 0.45 (black) to 0.84 (white).*

with $\mathcal{C}^{\mathcal{M}}$ being the *center of gravity* of the landmark positions:

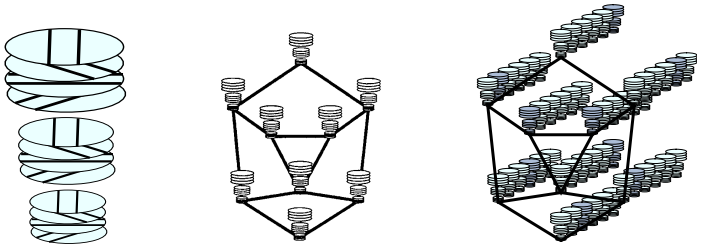
$$\mathcal{C}^{\mathcal{M}} = \frac{1}{L}\sum_{l=0}^{L-1}\mathcal{L}_l^{\mathcal{M}}\,. \tag{2.4-7}$$

The similarity map given in Figure 2.6 was generated computing the similarities:

$$S\left(\mathcal{G}^{\mathcal{M}},\mathcal{I}\right)\left(\vec{t}\right) = S^{\mathcal{G}}_{[A]}\left(\mathcal{G}^{\mathcal{M}},\mathcal{G}^{\mathcal{I}}\left(\vec{t}\right)\right) \tag{2.4-8}$$

at every offset position $\vec{t}$. It shows the advantage of taking a model graph for face detection instead of using a single Gabor jet. Although neither expression, nor size, or illumination of the model graph from Figure 2.6(a) and the probe image shown in Figure 2.6(b) is alike, the $S^{\mathcal{G}}_{[A]}$ similarity map in Figure 2.6(c) shows a clear maximum at the position of the face.

How to deal with landmark positions of the image graph that fall outside of the image boundaries, depends on the quality of the probe image. When the face is inside of the image boundaries on all accounts, these offset points can simply be forbidden and, thus, the resolution of the similarity map is less than the resolution of the image. When parts of the face may also be



(a) Gabor Jet **(b)** Model Graph **(c)** Bunch Graph

Figure 2.7: From Gabor Jet to Bunch Graph: *This figure, which was already shown in [97, 98], displays the creation of a bunch graph: (a) Simplified depiction of a Gabor jet. (b) Gabor jets at different landmark positions are arranged to form a model graph. (c) Several model graphs are combined into a bunch graph.*

outside of the image boundaries, e. g., as in images returned by a surveillance camera, missing nodes can either be set to the nearest position on the image boundaries, wrapped around, or ignored. In the last case, which was employed to generate Figure 2.6, $S^{\mathcal{G}}$ iterates only over the valid nodes, and the sum in Equation (2.4 – 4) is divided by the number of valid nodes.

The graph similarity measure $S^{\mathcal{G}}$ in Equation (2.4 – 4), which was introduced by Krüger *et al.* [42], is very powerful. It is used for face detection as well as for face recognition. Hence, it incorporates the ability to distinguish between faces and non-faces, e. g., background on the one hand, and has the potential to decode the identity that is stored in the model graph on the other hand. Furthermore, when $S_{[A]}$, $S_{[C]}$, or $S_{[M]}$ is used, it is able to deal with incorrectly placed landmark positions. Thus, node positions need not be located perfectly, which is nearly never the case during automatic face detection.

2.4.2 Bunch Graph

In the previous section, model graph $\mathcal{G}^{\mathcal{M}}$ and probe image $\mathcal{I}$ contained the same identity, which made face detection easy. In a common face recognition task, the identity shown in the image should be recognized and is not known in advance. Furthermore, face detection is usually applied to images that contain faces of identities that are novel to the system.

Since there is a high variety of frontal facial images with variations in identity, facial expression, lighting conditions, etc., it is not reasonable to use only a single model graph to detect the face in the probe image. On the other hand, it is not possible to have a training example for each combination. To deal with both issues, the concept of a *general face knowledge* was introduced in the PhD thesis [99] of Wiskott and renamed to *bunch graph* $\mathcal{G}^{\mathcal{B}} = \left(\vec{\mathcal{J}}^{\mathcal{B}}, \vec{\mathcal{L}}^{\mathcal{B}}, \vec{\mathcal{E}}\right)$ by Krüger *et al.* [42] and Wiskott *et al.* [98]. A bunch graph incorporates the information of a set T of usually hand-labeled training face graphs: $T = \{\mathcal{G}^{(b)} \mid b = 0, \ldots, B\text{-}1\}$, all sharing the same graph topology, i. e., having the same landmarks and edges in common. The node positions $\vec{\mathcal{L}}^{\mathcal{B}}$ of the bunch graph are averaged landmark positions of the training graphs rounded to integral pixel positions:

$$\mathcal{L}_l^{\mathcal{B}} = \left\lfloor \frac{1}{B} \sum_{b=0}^{B\text{-}1} \left(\mathcal{L}_l^{(b)} - \mathcal{C}^{(b)} \right) \right\rceil . \tag{2.4–9}$$

The transformation center of graphs is usually[5] in the center of gravity of the graphs, so it can simply be subtracted from the node positions. To keep the equations in Chapter 3 that modify the bunch graph simple, bunch graphs are used with a vanishing center of gravity: $\mathcal{C}^{\mathcal{B}} = (0, 0)^{\mathrm{T}}$.

The *bunch*:

$$\mathcal{J}_l^{\mathcal{B}} = \left\{ \mathcal{J}_l^{(b)} \mid b = 0, \ldots, B\text{-}1 \right\} \tag{2.4–10}$$

embraces the Gabor jets from all training graphs at landmark l. Hence, the Gabor jets in each bunch $\mathcal{J}_l^{\mathcal{B}}$ encode different occurrences of the same landmark l, e. g., the eye bunches incorporate eyes from different persons and in different states. The creation of a bunch graph is clarified in Figure 2.7, which was first shown by Krüger *et al.* [42].

The detection procedure is mostly the same as for a single model graph $\mathcal{G}^{\mathcal{M}}$. The only difference is the similarity calculation between image graph $\mathcal{G}^{\mathcal{I}}$ and bunch graph $\mathcal{G}^{\mathcal{B}}$ [97, 20]:

$$S_{[\cdot]}^{\mathcal{B}}\left(\mathcal{G}^{\mathcal{B}}, \mathcal{G}^{\mathcal{I}}\right) = \frac{1}{L} \sum_{l=0}^{L\text{-}1} \max_b S_{[\cdot]}\left(\mathcal{J}_l^{(b)}, \mathcal{J}_l^{\mathcal{I}}\right) . \tag{2.4–11}$$

Hence, for each Gabor jet $\mathcal{J}_l^{\mathcal{I}}$ the most similar Gabor jet of the bunch at this landmark is chosen. The selection procedure is illustrated in Figure 2.7(c). The darker Gabor jets in the bunches mark the best matching training Gabor

[5] There is only one violation of this rule, the image standardization uses the point between the eyes as transformation center (see Section 2.4.3).

jets, which for each landmark might stem from a different training graph and, thus, from a different face. This makes the algorithm more robust to variations in the facial image, but the required calculation time and memory load scale linearly with the size of training set T, i. e., the size B of the bunches.

2.4.3 Graph and Image Standardization

In order to collect several face graphs into one bunch graph, it is helpful that they share approximately the same size. This is assured by standardizing the graphs and the underlying images using the hand-labeled right and left eye landmarks $\mathcal{L}_0$ and $\mathcal{L}_1$, respectively, as fixed points and taking the center of the eye landmark as transformation center. My choice of the normalization is such that the image resolution $\vec{R}$ is set to 168×224 pixel, the distance between the eyes is $\frac{1}{3.5}$ times the horizontal resolution R_h, i. e., 48 pixel, and the vertical eye position is set to pixel coordinate $\lfloor 0.45\, R_v \rceil = 101$. Hence, the transformation:

$$\vec{x}_{\mathrm{std}} = \left\lfloor s\, Q(\alpha) \left(\vec{x} - \frac{\mathcal{L}_0 + \mathcal{L}_1}{2} \right) + \left(\frac{R_h}{2}, 0.45\, R_v \right)^{\mathrm{T}} \right\rceil , \qquad (2.4-12)$$

with:

$$s = \frac{\frac{1}{3.5}\, R_h}{\|\mathcal{L}_0 - \mathcal{L}_1\|} \qquad \text{and} \qquad \alpha = \arctan\left(\frac{(\mathcal{L}_0 - \mathcal{L}_1)_v}{(\mathcal{L}_0 - \mathcal{L}_1)_h} \right) \qquad (2.4-13)$$

is applied to each landmark position of the graph.

The same transformation is used for the pixels of the image:

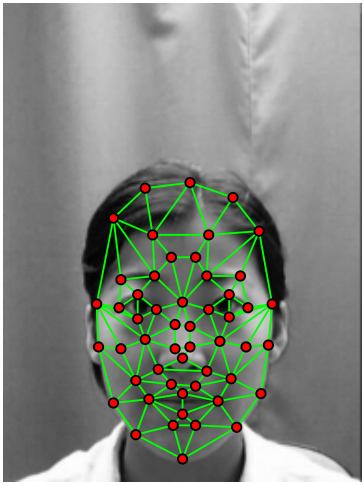
$$\forall\, \vec{x}_{\mathrm{std}} \in \{0, \ldots, R_h\text{-}1\} \times \{0, \ldots, R_v\text{-}1\} \quad : \quad \mathcal{I}_{\mathrm{std}}(\vec{x}_{\mathrm{std}}) = \mathcal{I}(\vec{x}) \ . \qquad (2.4-14)$$

When $\vec{x}$ is not integral in the source image, bi-linear interpolation is employed. In case the face is too near to the image boundary, a special boundary treatment is implemented. When the source pixel $\vec{x}$ is outside the image boundaries, a near pixel $\vec{x}'$ from the rim of the image is copied:

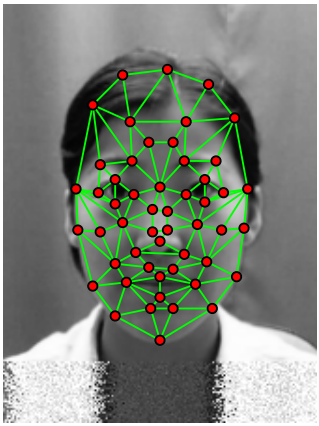
$$\begin{aligned} x'_h &= \mathrm{clip}(x_h + \mathrm{rand}(\text{-}3, 3)\,, 0, R_h) \ , \\ x'_v &= \mathrm{clip}(x_v + \mathrm{rand}(\text{-}3, 3)\,, 0, R_v) \ , \end{aligned} \qquad (2.4-15)$$

and some random noise is added:

$$\mathcal{I}_{\mathrm{std}}(\vec{x}_{\mathrm{std}}) = \mathrm{clip}\big(\mathrm{rand}(0.9, 1.1)\, \mathcal{I}\big(\vec{x}'\big)\,, 0, 255\big) \ . \qquad (2.4-16)$$



(a) Original Image with Hand-Labeled Face Graph



(b) Standardized Image with Standardized Face Graph

Figure 2.8: Graph and Image Standardization: *This figure shows the automatic preprocessing of the original hand-labeled graph shown in (a) to the standardized graph shown in (b). In the lower part of the preprocessed image, noise-padding is performed.*

In Equations (2.4 – 15) and (2.4 – 16), the rand(min, max) function generates a uniformly distributed pseudo-random value between min and max, and function clip(v, min, max) restricts value v to interval [min, max].

A pair of original and preprocessed hand-labeled training graphs is shown in Figure 2.8. Since the face graph does not include nodes at the eye centers, for each eye the average of the four eye nodes is taken instead. How the noise padding impacts the image can be obtained from the lower part of Figure 2.8(b).

Chapter 3

Face Detection and Feature Extraction

Before face recognition or classification of probe image $\mathcal{I}$ can be executed, the facial features of $\mathcal{I}$ must be extracted. In this chapter, state-of-the-art face detection and feature extraction algorithms are described. Afterwards, the *elastic bunch graph matching* (EBGM) algorithm from Wiskott *et al.* [96] is presented and two extensions of that model are introduced: Section 3.3 constitutes the *maximum likelihood* (ML) face detection and landmark localization, and Section 3.4 shows, how a fast and reliable multi-scale and multi-angle face detection can be integrated into the EBGM algorithm.

3.1 Face Detection Algorithms

In the last 20 years, many different approaches to face detection were introduced. Most algorithms use the pixel-based eigenface approach, some use Gabor features and the latest big progression employed Haar-like features. In this section, a small overview of some prominent face detection algorithms is given, a wider collection of methods can be obtained from Yang *et al.* [108] and Zhang *et al.* [109].

All presented face detection algorithms try to find a face in the probe image by employing the sliding window technology, which extracts image patches from several offset positions and at several scales and classifies, whether they contain a face. Dependent on the robustness of the detection algorithm to positioning and scale errors, the number of offset positions and the number of scales vary. Most detection algorithms can only classify patches of a certain size. This can be fulfilled easily by calculating image pyramids, scaling the image with the desired scale factors. Each of the extracted image patches is inspected and a probability for the patch containing a face is estimated. Depending on the task, either every patch with a probability above a certain threshold is accepted or, for the assumption of exactly one face in the image, the patch with the highest probability is chosen.

3.1.1 Face Detection with Eigenfaces

The basic idea of using eigenfaces for face detection goes back to Turk and Pentland [89]. They used an unsupervised learning technique to compute a *face space* by calculating statistics from a set of facial training images $\{\mathcal{I}^{(b)} \mid b = 0, \ldots, B\text{-}1\}$ that are standardized according to size and location of the face. These images are transformed into a vector $\vec{v}^{\,(b)}$ by stringing together all pixels of $\mathcal{I}^{(b)}$. Afterwards, the average face $\vec{\mu}$ and the covariance matrix Σ are calculated from these image vectors:

$$\vec{\mu} = \frac{1}{B}\sum_{b=0}^{B\text{-}1} \vec{v}^{\,(b)}\,, \qquad \Sigma = \frac{1}{B\text{-}1}\sum_{b=0}^{B\text{-}1} \left(\vec{v}^{\,(b)} - \vec{\mu}\right)\left(\vec{v}^{\,(b)} - \vec{\mu}\right)^{\mathrm{T}}\,. \qquad (3.1\text{–}1)$$

The covariance matrix is factorized into: $\Sigma^{-1} = \Phi\Lambda^{-1}\Phi^{\mathrm{T}}$ using the *Karhunen-Loève-transform* (KLT), resulting in the orthonormal transformation matrix Φ that contains the eigenvectors of Σ. Since these eigenvectors, when reinterpreted as images, look like faces [89, 52, 112], they are called *eigenfaces*. Each training image can be interpreted as a linear combination of eigenfaces:

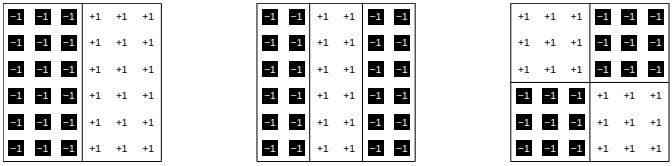
$$\vec{v}^{\,(b)} = \vec{\mu} + \Phi\,\vec{y}^{\,(b)} \qquad \Leftrightarrow \qquad \vec{y}^{\,(b)} = \Phi^{\mathrm{T}}\left(\vec{v}^{\,(b)} - \vec{\mu}\right)\,, \qquad (3.1\text{–}2)$$

where $\vec{y}^{\,(b)}$ is the training vector $\vec{v}^{\,(b)}$ transformed into *eigenspace* or face space. Usually, the eigenvector matrix Φ is shortened by keeping only the eigenvectors with the highest respective eigenvalues. Hence, the reconstruction of an image $\vec{v}$ is imperfect and there is some error:

$$\epsilon^2(\vec{v}) = \|\vec{v} - \vec{\mu} - \Phi\,\vec{y}\|^2 = \|\Phi^{\mathrm{T}}\,(\vec{v} - \vec{\mu}) - \vec{y}\|^2\,. \qquad (3.1\text{–}3)$$

This error measure can be used for face detection [89]. For a given image patch $\vec{v}$, the reconstruction error $\epsilon^2(\vec{v})$ is computed. For *non-face* image patches that do not contain a face, the reconstruction error is expected to be higher than the reconstruction error of face patches.

Unfortunately, also non-face patches might have a low $\epsilon^2(\vec{v})$ error value and be misinterpreted as a face patch. Sung and Poggio [84] tackled this issue by introducing a non-face class. They first calculated six centroids of the face class using an iterative elliptical k-means clustering algorithm, which resulted in six different mean vectors and transformation matrices. Using these centroids, they selected a set of non-face image patches that had small $\epsilon^2(\vec{v})$ errors and constructed another six non-face centroids with six non-face mean vectors and transformation matrices. The number of six face classes and six non-face classes are chosen arbitrarily [84]. For a novel image patch, the *distance in face space* (DIFS), a modified Mahalanobis distance



(a) Two Rectangles **(b)** Three Rectangles **(c)** Four Rectangles

Figure 3.1: Rectangular Filters used by Viola and Jones: *This figure displays the basic types of filters used by Viola and Jones [90].*

measure, and the *distance from face space* (DFFS), which is equal to the $\epsilon^2(\vec{v})$ error, to each of the 12 class means are computed. The resulting 24 distance values are classified to contain a face using an artificial neural network. Sung and Poggio [84] claim that the system works best when the non-face class is present and when both the DIFS and the DFFS are used for each class. They also state that it is very important to have appropriate non-face image patches.

Moghaddam and Pentland [52] caught up with the idea to build the probabilistic visual learning theory that consolidated the DIFS and DFFS idea, but still keeping only the face class, i. e., without having the need of non-faces for training. Since my face detection is built on their work, it is explained in more detail in Section 3.3.

3.1.2 Viola-Jones Face Detector

Another very famous face detection algorithm, which was introduced by Viola and Jones [90], uses Haar-like filters for feature extraction. These filters are built up from two, three, or four conjoined rectangular blocks, Figure 3.1 shows a complete set of filter types. The rectangular filters are scaled independently horizontally and vertically. Additionally, the filters built from two and three rectangles are rotated by 90°. Image features are created by convolving the image with these filters. Instead of calculating the convolutions directly, Viola and Jones used a precalculated *integral image* that permits the convolution with one block being calculated as the sum of four values. From that, each image feature can be computed with a maximum of 16 real-value operations, independent of the scale of the current filter.

The number of features extracted from an image patch is huge, Viola and Jones [90] report a number of 160,000 features per patch, approximately 400 times the number of its pixels. Of course, a single feature is not sufficient to define whether a face is present in the image or not. Hence, a classifier using exactly one of these features is *weak*. Nonetheless, a weighted majority vote of several weak classifiers can build a strong classifier that is highly reliable. Viola and Jones used AdaBoost [77] to learn weights for each weak classifier. Their training set contained 5000 facial images that were randomly downloaded from the Internet, cut out, and scaled to fit into 24×24 pixel images. Additionally, 350 million non-face image patches of the same size were created. The reported duration of the AdaBoost classifier training was in the order of weeks [90].

Even though each single feature can be extracted and classified extremely fast, the application of the complete set of classifiers to each single image patch is infeasible. Fortunately, most of the non-face image patches can be rejected using only a few weak classifiers. Therefore, Viola and Jones used a cascade of classifiers, each one of them being a little more complex. The very first classifier is built upon only two image features, both regarding the eye region, while later classifiers include more parts of the face. The cascade is trained as to permit all face patches, while blocking a good portion of non-face patches. The final classifier cascade consists of 38 stages, but only very few non-face patches pass through the penultimate layer.

3.1.3 Hierarchical Slow Feature Analysis

Although the Viola-Jones face detector is fast and fairly reliable, there is still further progress in face detection algorithms. Recently, the novel *hierarchical slow feature analysis* (HSFA) from Wiskott *et al.* [100] showed detection performance that is even above the Viola-Jones detector [54], but the number of false positives is still very high.

Slow feature analysis (SFA) is an unsupervised learning technique that analyzes the temporal variation in a given signal and extracts the features that change most slowly. Since temporal signal variations are investigated, the order of the training signals is crucial. To apply this technique to face detection, an ordered list of training images is generated such that the identity shown in the image varies fast, while the position of the face is changed slowly. Therefore, the slowest signal to be extracted from the training sequence is the position of the face. The application to probe images is instantaneous, i. e., the existence and the position of a face can be estimated for each single image patch. Hence, image sequences are needed for training, but not for testing.

Since the unsupervised HSFA approach is in principle only dependent on the order of the training images, any facial property might be classified. Escalante *et al.* [18] showed that the very same approach can be easily used to age and gender estimation by simply exchanging the order of the training data such that the slowest varying feature is age or gender[1], respectively.

3.1.4 Scale Invariant Feature Transform

The approach of the *scale invariant feature transform* (SIFT) [46] is different from the other algorithms since it integrates key point detection, feature extraction, scale and angle estimation, and object categorization into one algorithm. The SIFT algorithm is composed of four stages [47]. In the first stage, locations are estimated that do not change much during image scaling. For all candidate positions, the second stage performs key point localization and throws out key points that are not reliable. In the next stage, scale and angle are assigned to each key point based on local image gradients.

The final stage extracts feature vectors, so-called SIFT features, at the remaining key points. The elements of these vectors are composed of local orientation histograms around the key point, i. e., each element codes for amount and strength of the image gradient that points into one of eight uniformly distributed directions. One important characteristic of these orientation histograms is that they are computed relative to the scale and angle assigned for the current key point. This makes the resulting SIFT feature vector independent of the actual scale and angle of the feature. Hence, the same feature can be detected in the probe image even if the object is scaled or rotated, Lowe [47] also reported invariance against minor out-of-plane rotations.

For object detection, SIFT features are extracted from the probe image using the SIFT algorithm. Each of these features is matched into a gallery of stored SIFT features from known objects by computing the nearest neighbor. Since the key point locations of gallery and probe images do not always correspond, many misclassifications occur. If clusters of at least three different SIFT features are found that match in object and assigned scale and angle, the object is said to be detected in the according position, scale, and angle. Finally, the affine transformation, i. e., the 3D rotation, scale, and stretch of the object is estimated from the detected SIFT features [47].

[1] Escalante *et al.* [18] applied HSFA to artificially generated data, where they could continuously vary gender from male to female.

3.2 Elastic Bunch Graph Matching

Since face recognition and facial property classification as described in Chapter 4 rely on face graphs labeled with Gabor jets as representations of the faces, these graphs need to be located in the image. The most accurate way to locate the landmark positions is to hand-label all facial images with a face graph. Certainly, hand-labeling all images of a database by positioning all nodes of the face graph is unreasonable (although it seems that some people [93] actually did this work). Hence, another strategy for face detection and landmark localization needs to be found.

The *elastic graph matching* (EGM) algorithm and its extension *elastic bunch graph matching* (EBGM) [96, 98, 97, 20] are integrated face detection, landmark localization, and feature extraction procedures. The goal of EBGM is to find a face in probe image $\mathcal{I}$ and extract the image graph $\mathcal{G}^{\mathcal{I}}$ that best describes that face. The EBGM algorithm as stated in [96, 98, 97, 20] is split up in two distinctive steps. In the *face detection* step, EBGM tries to locate the position and the size of the face in the image. Afterwards, the *landmark localization* step fine-tunes the node position for each landmark separately. In the original version of the EBGM algorithm [98], two different bunch graphs were used for either of the two steps. While the face detection step employed a graph with a small amount of nodes, the landmark localization step, of course, needed a node for each landmark. In opposition, in this thesis the same graph topology is employed in both steps.

3.2.1 Face Detection

Global Move The face detection is composed of several *moves*. The first move that is applied to probe image $\mathcal{I}$ is the *global move*, which scans the whole image for the best location of the image graph, i. e., the offset point $\vec{t}^{\,*}$ with the highest bunch graph similarity (cf. Equation (2.4 – 11)):

$$\vec{t}^{\,*} = \arg\max_{\vec{t}} S^{\mathcal{B}}_{[\cdot]}\left(\mathcal{G}^{\mathcal{B}}, \mathcal{G}^{\mathcal{I}}\left(\vec{t}\right)\right) \qquad (3.2-1)$$

Similarly to the model graph from section Section 2.4.1, the bunch graph with its fixed node positions is placed at the image at offset point $\vec{t}$. The Gabor jets of the image graph $\mathcal{G}^{\mathcal{I}}$ are extracted at the landmark positions $\mathcal{L}^{\mathcal{I}}_{l}\left(\vec{t}\right)$ (cf. Equation (2.4 – 6)) relative to the particular offset point. Hence, even at the best matching location $\vec{t}^{\,*}$, the Gabor jets of the image graph $\mathcal{G}^{\mathcal{I}}$ will usually not fit perfectly to the Gabor jets stored in the bunches of the bunch graph. Hence, the similarity function between two Gabor jets

must account for displaced Gabor jets, which is done best[2] by the $S_{[\cdot]} = S_{[A]}$ function from Equation (2.3 – 10).

The most accurate, but also most time consuming, approach of detecting $\vec{t}^{\,*}$ would be to calculate the whole similarity map. Expecting this map to show a broad maximum like it does in Figure 2.6(c), the global move can be sped up enormously by using only some sparse grid positions to find an approximate $\vec{t}^{\,+}$ of the correct position $\vec{t}^{\,*}$ and search more densely for $\vec{t}^{\,*}$ only around $\vec{t}^{\,+}$. In most cases, it is sufficient to use interspaces of six or eight pixel in both horizontal and vertical direction to find that $\vec{t}^{\,+}$ that is next to the true maximum $\vec{t}^{\,*}$, and hence the correct $\vec{t}^{\,*}$ is located in the fine tune search. An exemplary result of the global move can be found in Figure 3.2(b), executed in one global search and two successive refinement steps. Although most of the node positions do not correspond to the landmarks very precisely, the face itself is found quite well.

Scale Move The second move is called the *scale move*. As the name suggests, the move tries to detect the best matching scale $\vec{s}^{\,*}$ of the image graph in horizontal and vertical direction. Scaling the image graph with scale $\vec{s} = (s_h, s_v)^{\mathrm{T}}$ is done by simply scaling the node positions, expecting a vanishing center of gravity $\mathcal{C}^{\mathcal{B}}$:

$$\mathcal{L}_l^{\mathcal{I}}\left(\vec{t}^{\,*}, \vec{s}\right) = \mathcal{L}_l^{\mathcal{B}} \bullet \vec{s} + \vec{t}^{\,*}\,, \tag{3.2-2}$$

where $\bullet$ here denotes the element-wise product of two vectors (also known as Hadamard or Schur product).

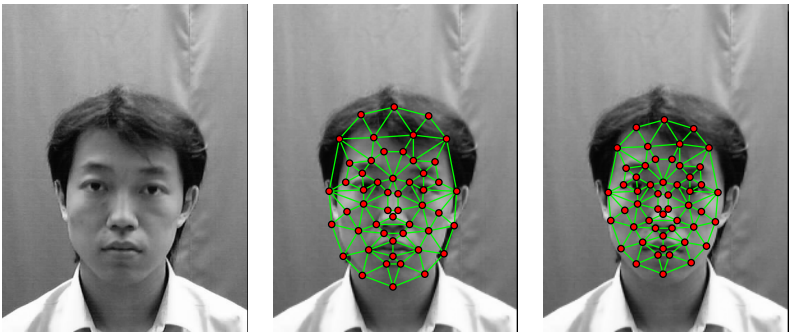
Also the scale move can be done in different granularities. Usually, the first scale move scales the image graph equally in horizontal and vertical direction, i. e., with $\vec{s} = (s, s)^{\mathrm{T}}$, while further scale moves try to improve the maximum similarity:

$$\vec{s}^{\,*} = \arg\max_{\vec{s}} S_{[\cdot]}^{\mathcal{B}}\left(\mathcal{G}^{\mathcal{B}}, \mathcal{G}^{\mathcal{I}}\left(\vec{t}^{\,*}, \vec{s}\right)\right) \tag{3.2-3}$$

by scaling the image graph independently horizontally and vertically.

The graph shown in Figure 3.2(c) was generated with a three round scale estimation. The first round scaled the graph with a fixed aspect ratio, while the second and third move scaled horizontally or vertically, respectively. In this specific case, the best mtching graph was scaled both in horizontal and vertical direction. All scale moves used the $S_{[A]}$ similarity function for

[2] Employing the novel phase correction method (cf. Appendix B), also the $S_{[D]}$ similarity function could be used for face detection. Still, tests if $S_{[D]}$ performs better than $S_{[A]}$ are pending.



(a) Probe Image **(b)** Global Move **(c)** Scale Move

Figure 3.2: EBGM Face Detection Example: *This figure shows exemplary results of the EBGM face detection steps: (b) the global move and (c) the scale move. For comparison, the probe image without any graph is given in (a).*

estimating the similarity between two Gabor jets. In general, also the $S_{[P]}$ or $S_{[D]}$ functions might be used, but there is no evidence, which function is better.

Often, the image graph has to be scaled because the size of the face in the image does not correspond to the size of the bunch graph. This includes that the texture that is extracted for the scaled image graph is also not of the same size as the texture stored in the bunch graph. As shown in Section 2.3.2, comparing Gabor jets of different scales is possible only in a very limited range of scales. Usually, when the absolute size of the face in the probe image differs from the faces stored in the bunch graph for scale factors $s > 1.2$ or $s < 0.8$, the face is unlocatable with the original EBGM approach. In Section 3.4, a novel procedure is introduced that also scales Gabor jets and, thus, enables a much larger scale range to be spanned.

Since scaling is done with respect to the center of gravity of the graph, but the most accurately locatable facial features, i. e., the eyes including the eyebrows are usually not in the center of the graph, scaling might result in an image graph slightly shifted against its optimal position $\vec{t}^{\,*}$. Hence, the search for $\vec{t}^{\,*}$ and $\vec{s}^{\,*}$ can be done by iteratively combining global and scale moves, e. g., by first scanning the whole image with a global move, doing the first scale move, performing further global and scale moves refining $(\vec{t}^{\,+}, \vec{s}^{\,+})$ found in the previous round, and so on.

3.2.2 Landmark Localization

After face detection, the image graph that is scaled to the best matching size is positioned over the face. Nonetheless, the nodes of the graph do not necessarily correspond to the correct landmarks yet, but might be displaced. This is due to the fact that on the one hand, faces of different persons are likely to have their facial features arranged slightly different, and on the other hand, facial expressions and head rotations in any direction influence the relative positions of the landmarks. Furthermore, the position and scale pair $(\vec{t}^{\,*}, \vec{s}^{\,*})$ that gives the best bunch graph similarity score does not always correspond to the most accurate position of the face in the image, e. g., when the setup of the face detection schedule is imperfect for the current image.

Hence, the landmark positions have to be fine-tuned, and now the graph geometry, i. e., the edges $\vec{\mathcal{E}}$ of the bunch graph come into play. To prevent the graph structure from being deformed to much, a graph geometry factor, which punishes the movement of nodes, is included into the bunch graph similarity from Equation (3.2 – 4) [96, 98]. To increase legibility, the $(\vec{t}^{\,*}, \vec{s}^{\,*})$-notation is omitted, but $\mathcal{L}_l^{\mathcal{I}} = \mathcal{L}_l^{\mathcal{I}}\left(\vec{t}^{\,*}, \vec{s}^{\,*}\right)$ and accordingly $\mathcal{J}_l^{\mathcal{I}} = \mathcal{J}^{\mathcal{I}}\left(\mathcal{L}_l^{\mathcal{I}}\left(\vec{t}^{\,*}, \vec{s}^{\,*}\right)\right)$ are used instead:

$$S^{\mathcal{B}}_{[\cdot,\mathcal{E}]}\left(\mathcal{G}^{\mathcal{B}}, \mathcal{G}^{\mathcal{I}}\right) = \frac{1}{L}\sum_{l=0}^{L-1} \max_{b} S_{[\cdot]}\left(\mathcal{J}_l^{\,(b)}, \mathcal{J}_l^{\mathcal{I}}\right) - \frac{w_{\mathcal{E}}}{E}\sum_{e=0}^{E-1} \frac{(\vec{\Delta}_e^{\mathcal{I}} - \vec{\Delta}_e^{\mathcal{B}})^2}{(\vec{\Delta}_e^{\mathcal{B}})^2}\,. \tag{3.2-4}$$

The new term of Equation (3.2 – 4) needs some explanation. Firstly, the penalty term includes the edge vector difference $\vec{\Delta}_e^{\mathcal{I}} - \vec{\Delta}_e^{\mathcal{B}}$ between image graph and bunch graph. Since the edge vectors are identical after the global move (but not after the scale moves), originally the difference is $\vec{0}$. The more one of the two linked nodes in the image graph moves, the higher the distance and, thus, the higher the punishment becomes. Secondly, the length of the edge vector difference is divided by the length of the original edge[3]. Hence, the longer the original edge is, the more the connected nodes may move in the image graph. Furthermore, this division makes the punishment term dimensionless, i. e., independent of the absolute distance of the landmarks and, therefore, independent of the size of the bunch graph. Finally, the geometry weight factor $w_{\mathcal{E}}$ defines, how important the geometry costs

[3]This extension was made in [98], the original version from Wiskott [99] did not use this normalization.

are. The extremes, i. e., $w_{\mathcal{E}} = 0$ and $w_{\mathcal{E}} \to \infty$ have the impacts that the nodes can move freely without punishment, or can not move at all, respectively. To set up this weight is difficult since the dimensionless punishment term and the similarity measure between two jets are hardly comparable. Wiskott *et al.* [98, 97] introduced weight factor $w_{\mathcal{E}} = 2$, which was also used to generate the graphs shown in Figure 3.3.

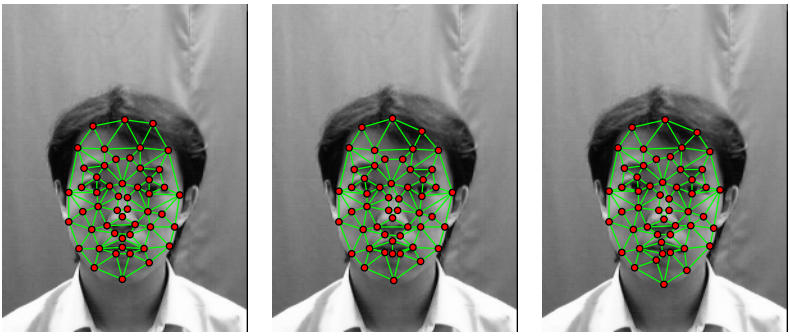
For each landmark $\mathcal{L}_l^{\mathcal{I}}$, the landmark localization procedure tries to find the displacement vector $\vec{d}_l$ that results in the highest bunch graph similarity. Afterwards, the distance vector $\vec{d}_l$ is added to the landmark position $\mathcal{L}_l^{\mathcal{I}}$ and the geometry costs are calculated using the moved landmark. This is done for each landmark independently. Of course, moving one landmark influences the geometry costs of all linked nodes. Wiskott *et al.* [98] addressed this issue by defining a pseudo-random order in which the nodes are searched, in Section 3.2.3 a more sophisticated approach to that problem is brought up.

Scan Local Move The search for the best fitting landmark position can be done in several ways. The first and most simple approach is the so-called *scan local move*, an exhaustive local search for the best displacement vector $\vec{d}_l$ in a fixed local region $\mathcal{R}$ around the previously estimated landmark position. Thus, for all $\vec{d}_l \in \mathcal{R}$, the Gabor jet $\mathcal{J}^{\mathcal{I}}\left(\mathcal{L}_l + \vec{d}_l\right)$ is extracted and compared to the Gabor jets of the bunch of this landmark. This time, the $S_{[P]}$ similarity function (cf. Equation (2.3 – 11)) is employed in the bunch graph similarity function. The result of this simple approach is given in Figure 3.3(a). Most of the nodes are located well, but the mouth and the left eye nodes miss their landmarks slightly. Since most parts of the similarity function are not changed by the local move of a single landmark, it is also possible to calculate only those values that are influenced by the node movement. Hence, it is sufficient to compute:

$$S^{\mathcal{B}_l}_{[P,\mathcal{E}]}\left(\mathcal{G}^{\mathcal{B}}, \mathcal{G}^{\mathcal{I}}, \vec{d}_l\right) = \max_b \left\{ S_{[P]}\left(\mathcal{J}_l{}^{(b)}, \mathcal{J}^{\mathcal{I}}\left(\mathcal{L}_l^{\mathcal{I}} + \vec{d}_l\right)\right)\right\} - \frac{w_{\mathcal{E}}}{E_l} \sum_{\vec{\Delta}_e \in \vec{\mathcal{E}}_l} \frac{(\vec{\Delta}_e^{\mathcal{I}} + \vec{d}_l - \vec{\Delta}_e^{\mathcal{B}})^2}{(\vec{\Delta}_e^{\mathcal{B}})^2} \tag{3.2–5}$$

with $\vec{\mathcal{E}}_l$ being the set of edges vectors from landmark $\mathcal{L}_l$ to linked landmarks, and E_l being the number of links of $\mathcal{L}_l$.

Disparity Local Move The *disparity local move*, the second approach, which is presented in the original EBGM paper [98], tries to estimate the disparity vector $\vec{d}$ directly by exploiting the phases stored in the Gabor jets.



(a) Scan Local Move **(b)** Disparity Local Move **(c)** Disparity Scan Local Move

Figure 3.3: EBGM Landmark Localization Example: *This figure shows a comparison between different versions of the EBGM landmark localization: (a) was generated by the scan local move, while (b) used the disparity local move, whereas (c) employed a combination of both.*

The Gabor jet ${\mathcal{J}_l}^{(b)}$ of the bunch is compared to the image Gabor jet $\mathcal{J}_l^{\mathcal{I}}$ using the $S_{[D]}$ similarity function (cf. Equation (2.3 – 12)). The disparities $\vec{d}_l^{(b)}$ are computed for all landmarks $\mathcal{L}_l$ and for all Gabor jets ${\mathcal{J}_l}^{(b)}$ of the bunch of this landmark. Details of the disparity vector estimation are given in Appendix B. Afterwards, the similarity for the landmark is determined by maximizing the $S_{[D]}$ similarity and the geometry costs of the estimated disparity over all Gabor jets in the bunch:

$$S^{\mathcal{B}_l}_{[D,\mathcal{E}]}\left(\mathcal{G}^{\mathcal{B}},\mathcal{G}^{\mathcal{I}}\right) = \max_b \left\{ S_{[D]}\left({\mathcal{J}_l}^{(b)}, \mathcal{J}^{\mathcal{I}}\left(\mathcal{L}_l^{\mathcal{I}}\right)\right) - \frac{w_{\mathcal{E}}}{E_l} \sum_{\vec{\Delta}_e \in \vec{\mathcal{E}}_l} \left(\frac{(\vec{\Delta}_e^{\mathcal{I}} + \vec{d}_l^{(b)} - \vec{\Delta}_e^{\mathcal{B}})^2}{(\vec{\Delta}_e^{\mathcal{B}})^2} \right) \right\}. \tag{3.2-6}$$

The estimated displacement is usually not integral, but the Gabor jets can be taken only from integral pixel positions. Hence, the disparity vector has to be rounded in order to move the node. This implies that the result of the $S_{[D]}$ function is not the same as the result of the $S_{[P]}$ function.

Disparity Scan Local Move The result of the disparity scan local move for each single node is shown in Figure 3.3(b), but the landmark localization is in general worse than the results of the scan local move. The *disparity scan local move*, a combination of both approaches I invented, seems to generate better results. The landmark positions of Figure 3.3(c) were located using the same setup as for the scan local move, but instead of using the $S_{[P]}$ function for comparing Gabor jets, the $S_{[D]}$ function was employed. The geometry costs then include both the estimated disparity $\vec{d}_l^{\,(b)}$ for the current Gabor jet of the bunch and the currently scanned displacement $\vec{d}_l$. One advantage of the disparity scan local move is the fact that the density of the search grid might be reduced, e. g., to interspaces of 4×4 pixel since neighboring image Gabor jets usually create disparity vectors targeting the same location (cf. Appendix B.5), without loosing the benefits of the scan local move.

3.2.3 Iterative Local Moves

Wiskott *et al.* [98] proposed a pseudo-random order in which the nodes of the graphs are fine-tuned. As an alternative, the node position correction $\vec{d}_l^{\,*}$ can be computed for each landmark before applying all of them at the same time, instead of adding each vector directly after calculation. Hence, the actual order of iterating through the nodes of the graph does no longer influence the result and, thus, one node being moved towards the wrong direction will not influence the remaining nodes.

But, of course, the influence of a node pulled correctly also no longer has an effect on the geometry costs of the remaining nodes. Therefore, it is more likely for the nodes to stay at their previous positions since the geometry costs are vanishing there. To avoid this, the schedule is extended to include several rounds of scan local moves, or disparity scan local moves, with increasing geometry costs $w_{\mathcal{E}}$ in each round. For the first move, a geometry weight of, say, $w_{\mathcal{E}} = 0.01$ allows all nodes to move nearly without geometry influence, permitting single nodes to be misplaced. In the further rounds with higher weights, these solitary outliers are recaptured by geometry restrictions.

3.2.4 Issues of the EBGM Algorithm

There are some issues about the EBGM algorithm that need to be discussed. The first point is the detection accuracy. Sometimes, the EBGM algorithm detects faces in the background, even if the background is just plain with a little noise. In Section 2.4, it is shown that single Gabor jets are well comparable to the background, but it is stated that Gabor jets in a fixed spatial

arrangement can be located more robustly. In opposition to the graph similarity map shown in Figure 2.6, now there is bunch of jets at each landmark and – and that is the most important point – the jet from the bunch is taken that generates highest similarity value. Hence, in the background the Gabor jet of the bunch is taken that best matches the background texture. This implies that the similarity of bunches to background noise increase if more Gabor jets are in the bunch. Using training Gabor jets that are taken from, e. g., an occluded part of the face amplifies that effect, even if there is only a single corrupted Gabor jet in the bunch.

The second argument against a high number of Gabor jets in the bunch is that the detection time is linear in that number. The number of graphs in the bunch graph that Wiskott [99] assumes to be sufficient for placing graphs automatically is 70. Hence, the detection procedure would need approximately 70 times the time of matching a single graph. In real-time applications this is not sufficient. On the other hand, the detection accuracy is also bad if the number of Gabor jets in the bunch is to low. Following the argument of having different occurrences of each landmark in the bunch, occurrences that are not in the bunch are harder to detect. And if one hand-labeled node position in the training set T is only slightly misplaced with respect to the correct landmark position, this landmark has a higher probability to be misplaced in the landmark localization step. In summary, both the number and the nature of the Gabor jets in the bunches are very critical.

3.2.5 Former Extensions of EBGM

Another problem of the EBGM algorithm is that, although the model itself is elastic, it cannot deal with pose variations and extreme facial expressions. These variations introduce node positions that are not captured by scale and local moves and also the texture of the landmarks changes. Tewes [87] extended the face graph to build a *flexible object model* (FOM), which learns the node position and texture variations created by facial expressions or poses for a single person. For each node, he computed a *shape to texture map* (STTM) that maps the texture stored in the Gabor jet according to global graph deformations. Integrating information from different identities, he further invented the *extended flexible object model* (EFOM), in which each node holds the STTM's from several persons.

Finally, he added a *learned move* to the EBGM schedule. Assuming the location of the face already to be found by previous global moves, the learned move generates expression or pose variations of the model and tests, which of them fits best to the current probe image location. Since the texture modification that occurs together with the pose or expression variation is

learned by the model, the texture comparison is always done according to the currently tested pose or expression.

Additionally, Tewes [87] opined that these texture transformations could be used for face recognition. As soon as the correct pose or expression is detected, the texture stored in the Gabor jets can be transformed into the texture of a frontal face with neutral expression. Tewes *et al.* [86] showed some images reconstructed from texture transformed in this manner, which indicate that this might successfully improve identification of the face. Unfortunately, they learned the transformations identity-dependent, the demonstration of the method working for unknown faces is still pending. Especially, creating identity-independent STTM's that would allow unknown face textures to be normalized was not yet achieved.

A further previous approach to learning, how to improve face detection in the elastic bunch graph architecture was done by Heinrichs *et al.* [32]. Instead of holding a bunch of jets for each node, they performed a PCA on the Gabor jets of the bunch and used only the first 80 eigenvectors. Afterwards, they applied a similarity measure quite similar to one that is described in Section 3.3.1, including all issues of that function. They showed that the node positioning error was approximately halved on artificial data.

Heinrichs *et al.* [32] also stated that more precisely detected nodes would improve face recognition accuracy, but their plots depict the opposite. When they disregarded the eigenvectors with the highest eigenvalues, their positioning error increased, while the recognition error dropped. In some cases, the detected nodes performed even better than the ground truth node positions. To show that their assumption does not hold, Neuser [60] conducted experiments intentionally misplacing the nodes of the graphs. He showed that identity-dependently misplaced nodes are able to improve recognition accuracy, while randomly misplaced nodes usually result in the opposite. Hence, having a face detection and landmark localization algorithm that is able to introduce identity-dependent node positioning errors boosts identification rates, they are higher than having an algorithm that perfectly hits all landmarks [60].

3.3 Maximum Likelihood Face Detection

To overcome some of the issues of the EBGM algorithm described in the last section, it should be considered to use a simple learning algorithm that exploits the statistics of the Gabor jets stored in the graphs. Since hand-labeled face graphs are used for training, this learning algorithm needs to deal with few training examples and, hence, should not have too many free variables.

Furthermore, to avoid the need of setting up task-dependent parameters, the proposed model should have as few parameters as possible.

3.3.1 Preliminary Work

The basic idea of the proposed learning algorithm goes back to the work of Moghaddam and Pentland [52, 53]. In their approach, image patches of different size are extracted from the source image, normalized to a fixed number of pixels and linearized into input patterns $\vec{v}$ (cf. Section 3.1.1) of length $N = R_h\, R_v$. The *likelihood* $\mathcal{P}(\vec{v} \mid \Omega)$ of input pattern $\vec{v}$ given the class Ω of faces is estimated:

$$\mathcal{P}(\vec{v} \mid \Omega) = \frac{\mathrm{e}^{-\frac{1}{2}(\vec{v}-\vec{\mu})^{\mathrm{T}}\Sigma^{-1}(\vec{v}-\vec{\mu})}}{(2\pi)^{\frac{N}{2}}\,|\Sigma|^{\frac{1}{2}}}\,, \tag{3.3-1}$$

assuming a Gaussian distribution of patterns $\vec{v}$ with mean $\vec{\mu}$ and covariance matrix Σ. The likelihood $\mathcal{P}(\vec{v} \mid \Omega)$ is calculated for each image patch, and the best matching patch is taken as the one containing the face. Mean vector $\vec{\mu}$ and covariance matrix Σ are calculated:

$$\vec{\mu} = \frac{1}{B}\sum_{b=0}^{B-1}\vec{v}^{\,(b)} \qquad \Sigma = \frac{1}{B-1}\sum_{b=0}^{B-1}\left(\vec{v}^{\,(b)}-\vec{\mu}\right)\left(\vec{v}^{\,(b)}-\vec{\mu}\right)^{\mathrm{T}} \tag{3.3-2}$$

from a number of training image patterns $\vec{v}^{\,(b)}$ $(b = 0, \ldots, B\text{-}1)$ that show standardized faces, i. e., all images have the same resolution and the faces are aligned to the hand-labeled eye positions.

Instead of calculating the likelihood $\mathcal{P}(\vec{v} \mid \Omega)$ from Equation (3.3–1) directly, the inverted covariance matrix is decomposed into $\Sigma^{-1} = \Phi\Lambda^{-1}\Phi^{\mathrm{T}}$ employing Karhunen-Loève-transform (KLT), with Λ being the diagonal matrix of eigenvalues λ_n $(n = 0, \ldots, N\text{-}1)$ and Φ the matrix of eigenvectors $\vec{\Phi}_n$. This matrix Φ is ordered such that the eigenvectors are arranged according to decreasing eigenvalues: $\lambda_0 \geq \lambda_1 \geq \ldots \geq \lambda_{N-1}$. With this matrix of eigenvectors, the input pattern $\vec{v}$ is transformed into the eigenspace:

$$\vec{y} = \Phi^{\mathrm{T}}\,(\vec{v}-\vec{\mu})\,, \tag{3.3-3}$$

and the exponent of the exponential function from Equation (3.3–1), which is also known as the dimensionless *Mahalanobis distance* from $\vec{v}$ to $\vec{\mu}$, can be

rewritten as:

$$\begin{aligned} D(\vec{v},\vec{\mu}) &= \|\vec{v}-\vec{\mu}\|_{\Lambda} \\ &= (\vec{v}-\vec{\mu})^{\mathrm{T}}\,\Sigma^{-1}\,(\vec{v}-\vec{\mu}) \\ &= \vec{y}^{\mathrm{T}}\Lambda^{-1}\vec{y} \\ &= \sum_{n=0}^{N-1}\frac{y_n^2}{\lambda_n}\,. \end{aligned} \tag{3.3-4}$$

Still, the transformed vector $\vec{y}$ has the same dimension N as the original input pattern $\vec{v}$, but most of the information in the vector, i. e., the values that correspond to eigenvectors with low eigenvalues can be regarded as noise and, thus, left out. This is usually done by reducing the number of columns of the eigenvector matrix Φ to $M < B \ll N$ and removing the latest $N - M$ eigenvectors. Hence, the dimensionality of Φ is decreased to $N \times M$, and the resulting transformed vector $\vec{y}$ is of dimension M.

Moghaddam and Pentland [53] split up the Mahalanobis distance into two parts, the *distance in feature space* (DIFS) and the *distance from feature space* (DFFS):

$$\begin{aligned} \hat{D}(\vec{v},\vec{\mu}) &= \sum_{n=0}^{M-1}\frac{y_n^2}{\lambda_n} + \sum_{n=M}^{N-1}\frac{y_n^2}{\rho} \\ &= \underbrace{\sum_{n=0}^{M-1}\frac{y_n^2}{\lambda_n}}_{} + \underbrace{\frac{\epsilon^2(\vec{v})}{\rho}}_{} \\ &\qquad \mathrm{DIFS} \;+ \mathrm{DFFS}\,, \end{aligned} \tag{3.3-5}$$

with:

$$\rho = \frac{1}{B-M}\sum_{n=M}^{B-1}\lambda_n \tag{3.3-6}$$

being the average of the non-zero eigenvalues from the left-out eigenvectors. The residual error $\epsilon^2(\vec{v})$ can easily be computed using the fact that the KLT does not change the length of the vectors, and, thus:

$$\epsilon^2(\vec{v}) = \sum_{n=M}^{N-1} y_n^2 = \|\vec{v}-\vec{\mu}\|^2 - \sum_{n=0}^{M-1} y_n^2 \tag{3.3-7}$$

holds. Therefore, the values $y_M, \ldots, y_{N-1}$ do not need to be computed to estimate the residual error. Integrating Equation (3.3 – 5) into the probability

estimation from Equation (3.3 – 1) yields:

$$\begin{aligned}\hat{\mathcal{P}}(\vec{v} \mid \Omega) &= \left[\frac{\mathrm{e}^{-\frac{1}{2}\sum_{n=0}^{M-1}\frac{y_n^2}{\lambda_n}}}{(2\pi)^{\frac{M}{2}}\prod_{n=0}^{M-1}\lambda_n^{\frac{1}{2}}}\right]\left[\frac{\mathrm{e}^{-\frac{\epsilon^2(\vec{v})}{2\rho}}}{(2\pi\rho)^{\frac{N-M}{2}}}\right] \\ &= \mathcal{P}_{\mathrm{DIFS}}(\vec{v} \mid \Omega) \quad \hat{\mathcal{P}}_{\mathrm{DFFS}}(\vec{v} \mid \Omega)\ .\end{aligned} \tag{3.3 – 8}$$

Moghaddam and Pentland [53] claimed that it is also possible to use the DIFS measure disregarding DFFS, but they also showed that the detection accuracy is worse than taking both measures into account.

How to get to a valuable number M of kept eigenvectors is by far neither easy nor perspicuous. Of course, M must be less than the number of training images B since only B-1 uncorrelated eigenvectors can be computed from B training vectors. Moghaddam and Pentland [53] use $M = 5$, $M = 10$, $M = 20$ or $M = 100$ on different databases, without giving explanations to these numbers.

3.3.2 Simplifications

In his MSc thesis [85], Teixeira pointed out that due to the products in the denominators the likelihood in Equation (3.3 – 8) is numerically incomputable. He showed a simple example, where the value of the multiplied denominators was in the order of $10^{30,000}$, which is far far beyond any number representable in the double precision IEEE floating point format.

Teixeira's solution to this problem was the factorization $\hat{\mathcal{P}}(\vec{v} \mid \Omega) = f(\Omega) \cdot g(\vec{v}, \Omega)$ of the likelihood Equation (3.3 – 8) into:

$$\begin{aligned}f(\Omega) &= \frac{1}{(2\pi)^{\frac{M}{2}}\prod_{n=0}^{M-1}\lambda_n^{\frac{1}{2}}} \cdot \frac{1}{(2\pi\rho)^{\frac{N-M}{2}}} \\ g(\vec{v}, \Omega) &= \mathrm{e}^{-\frac{1}{2}\left[\sum_{n=0}^{M-1}\frac{y_n^2}{\lambda_n} + \frac{\epsilon^2(\vec{v})}{\rho}\right]}\ .\end{aligned} \tag{3.3 – 9}$$

He showed that the relation of two probabilities $\mathcal{P}(\vec{v}_1 \mid \Omega)$ and $\mathcal{P}(\vec{v}_2 \mid \Omega)$ is preserved, when only the g-function is used:

$$\mathcal{P}(\vec{v}_1 \mid \Omega) < \mathcal{P}(\vec{v}_2 \mid \Omega) \Longleftrightarrow g(\vec{v}_1, \Omega) < g(\vec{v}_2, \Omega)\ . \tag{3.3 – 10}$$

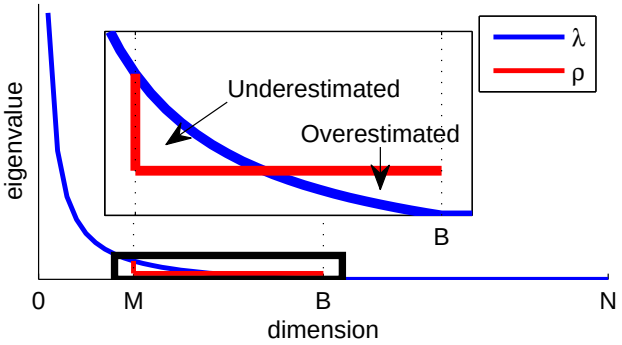


Figure 3.4: Averaging Eigenvalues: *This figure, which was replicated from a graphic shown by Teixeira [85], illustrates the effect of averaging eigenvalues in Equation (3.3 – 11).*

Taking the logarithm of g and removing the $\frac{1}{2}$ factor, Teixeira [85] defined the maximum likelihood score:

$$S^{\mathrm{ML}}(\vec{\mu}, \Sigma, \vec{v}) = -\sum_{n=0}^{M-1} \frac{y_n^2}{\lambda_n} - \frac{\epsilon(\vec{v})}{\rho}\,, \qquad (3.3-11)$$

which is much easier than Equation (3.3 – 8) from [53], but holds basically the same information. Another nice point is that this score is still dimensionless. Therefore, different scores that, e. g., belong to different types of features can easily be added up.

Unfortunately, Equation (3.3 – 11) yet includes two parts that are not very intuitive. A first issue is that the ρ value used in the DFFS calculation from Equation (3.3 – 5) looks somewhat odd, Teixeira[85] showed a nice depiction of the effect of averaging the eigenvalues that is replicated in Figure 3.4. Taking a closer look at it, obviously the value of y_M is divided by ρ, which is much smaller than λ_M and, thus, the importance of this value is higher than the importance of the last dimension y_{M-1} of the DIFS. Therefore, the choice of the correct M, which is the second issue in Equation (3.3 – 11), is very important.

Another point is the need for many training data, which comes from the fact that PCA computes a full covariance matrix Σ, or at least covariance matrix of size $B \times B$ when using the snapshot method as described in the

MSc thesis [106] of Yambor[4]. With just a few training inputs in the high dimensional input space, the complete set of variables in the covariance matrix is in no way reliably predictable. And since hand-labeled Gabor graphs are used for maximum likelihood face detection in this chapter, the number of training inputs is very limited.

In our publication [29], we introduced a way to deal with all those issues at once: the PCA is eliminated from the calculation, and only the mean $\vec{\mu}$ and variance $\vec{\kappa}$ from the training data are used, completely disregarding covariances. The probability score from Equation (3.3 – 11) can further be simplified, yielding:

$$S^{\mathrm{ML}}(\vec{\mu}, \vec{\kappa}, \vec{v}) = -\sum_{n=0}^{N\text{-}1} \frac{(v_n - \mu_n)^2}{\kappa_n} , \tag{3.3 – 12}$$

with mean $\vec{\mu}$ and variance $\vec{\kappa}$ being estimated by:

$$\begin{aligned} \vec{\mu} &= \frac{1}{B} \sum_{b=0}^{B\text{-}1} \vec{v}^{\,(b)} , \\ \vec{\kappa} &= \frac{1}{B\text{-}1} \sum_{b=0}^{B\text{-}1} \left(\vec{v}^{\,(b)} - \vec{\mu}\right) \bullet \left(\vec{v}^{\,(b)} - \vec{\mu}\right) . \end{aligned} \tag{3.3 – 13}$$

Equation (3.3 – 12) has some major advantages. One of the most important facts is the time complexity. Since probability scores are estimated very often, they need to be computed fast. When employing Equation (3.3 – 12) for probability estimation, time complexity $O(N)$ is achieved, in comparison to $O(N\,M)$ for PCA from Equation (3.3 – 11). But also the training stage, i. e., the estimation of $\vec{\mu}$ and $\vec{\kappa}$ is linear in N and the number of training graphs B, whereas the PCA needs $O\left(N^3\right)$ for the inversion of the covariance matrix, or at least $O\left(B^3\,N\right)$ when using the snapshot method [106]. Another point is that no residual error ϵ^2 needs to be computed and no M has to be chosen. Finally, the benefit of the Mahalanobis-like distance measure is not lost, the resulting S^{ML} probability score is still dimensionless.

Of course, for the original eigenface approach of Moghaddam and Pentland [53], the algorithm would break down when PCA is removed. Neighboring pixels in the source images are highly correlated, but these correlations are not included into the pixel vectors $\vec{v}$. When applied to Gabor wavelet responses, the correlations between neighboring pixels are already included in the Gabor jets and, hence, do not need to be impressed artificially.

[4] Yambor [106] gives nice introductions into PCA and LDA algorithms for image classification. A must read!

3.3.3 Maximum Likelihood Estimators

The texture statistics of landmark $\mathcal{L}_l$ can be estimated by the statistics of the training Gabor jets taken at this landmark. In the EBGM approach, these Gabor jets are stored in the bunch $\mathcal{J}_l^{\mathcal{B}} = \{\mathcal{J}_l{}^{(b)} \mid b = 1, \ldots, B\}$ (cf. Equation (2.4 – 10)). This bunch is now replaced by the *maximum likelihood Gabor jet* $\mathcal{J}_l^{\mathrm{ML}} = (\vec{\mu}_l^{[a]}, \vec{\kappa}_l^{[a]}, \vec{\mu}_l^{[\phi]}, \vec{\kappa}_l^{[\phi]})$ that includes the mean and variance information of the absolute and phase values of the training Gabor jets. Mean and variance of the absolute values are computed:

$$\begin{aligned} \mu_{l;j}^{[a]} &= \frac{1}{B} \sum_{b=0}^{B\text{-}1} a_{l;j}^{(b)} \,, \\ \kappa_{l;j}^{[a]} &= \frac{1}{B\text{-}1} \sum_{b=0}^{B\text{-}1} \left(a_{l;j}^{(b)} - \mu_{l;j}^{[a]} \right)^2 \end{aligned} \tag{3.3 – 14}$$

identically to Equation (3.3 – 13). Please note that all Gabor jets are normalized to unit norm as given in Equation (2.3 – 5).

Due to the circular structure of the phase values, the calculation of mean and variance of the phases is more complicated:

$$\begin{aligned} \mu_{l;j}^{[\phi]} &= \arg \left[\frac{1}{B} \sum_{b=0}^{B\text{-}1} \left(\mathcal{J}_l{}^{(b)} \right)_j \right] \,, \\ \kappa_{l;j}^{[\phi]} &= \frac{1}{B\text{-}1} \sum_{b=0}^{B\text{-}1} \frac{d_\phi \left(\phi_{l;j}^{(b)} - \mu_{l;j}^{[\phi]} \right)^2 a_{l;j}^{(b)}}{\mu_{l;j}^{[a]}} \,. \end{aligned} \tag{3.3 – 15}$$

The mean phase is computed as the phase of the mean complex value. The phase variance is calculated using the phase difference normalization function d_ϕ that reduces $\phi - \mu^{[\phi]}$ modulo 2π into $[\text{-}\pi, \pi[$. To minimize the impact of phase jumps due to low absolute values, the factor $\frac{a}{\mu^{[a]}}$ was added empirically.

For the landmark localization step, a further estimator for the edges is needed. This estimator should later estimate the correctness of a node position in correlation to its linked nodes, similar to the geometry factor of the bunch graph similarity function from Equation (3.2 – 4). For that purpose, the distribution of the edge vectors $\vec{\Delta}_e$ between linked landmarks is learned. Since landmarks have very different tendencies for horizontal and vertical movement, both directions are learned independently. Hence, for

either direction, a separate estimator has to be trained:

$$\begin{aligned}
\mu_e^{[\mathcal{E}_h]} &= \frac{1}{B}\sum_{b=0}^{B\text{-}1}\Delta_{e;h}^{(b)}\,,\\
\kappa_e^{[\mathcal{E}_h]} &= \frac{1}{B\text{-}1}\sum_{b=0}^{B\text{-}1}\left(\Delta_{e;h}^{(b)}-\mu_e^{[\mathcal{E}_h]}\right)^2,\\
\mu_e^{[\mathcal{E}_v]} &= \frac{1}{B}\sum_{b=0}^{B\text{-}1}\Delta_{e;v}^{(b)}\,,\\
\kappa_e^{[\mathcal{E}_v]} &= \frac{1}{B\text{-}1}\sum_{b=0}^{B\text{-}1}\left(\Delta_{e;v}^{(b)}-\mu_e^{[\mathcal{E}_v]}\right)^2,
\end{aligned} \tag{3.3–16}$$

where $\Delta_{e;h}$ and $\Delta_{e;v}$ name the horizontal or vertical component of the edge vector $\vec{\Delta}_e$, respectively. These four estimators are combined to edge label $\mathcal{E}_e^{\mathrm{ML}} = (\mu^{[\mathcal{E}_h]}, \kappa^{[\mathcal{E}_h]}, \mu^{[\mathcal{E}_v]}, \kappa^{[\mathcal{E}_v]})$.

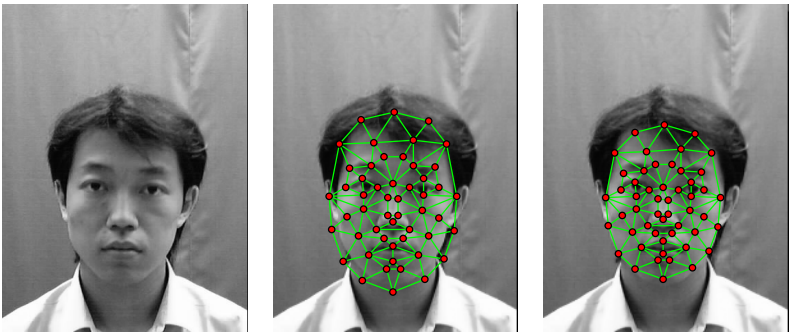
Finally, all maximum likelihood estimators are collected in the *maximum likelihood graph* $\mathcal{G}^{\mathrm{ML}} = (\vec{\mathcal{J}}^{\mathrm{ML}}, \vec{\mathcal{L}}^{\mathrm{ML}}, \vec{\mathcal{E}}^{\mathrm{ML}})$ similar to the bunch graph. The landmarks $\vec{\mathcal{L}}^{\mathrm{ML}}$ are computed as the average of the landmark positions of the training graphs, they are identical to $\vec{\mathcal{L}}^{\mathcal{B}}$ in Equation (2.4–9).

3.3.4 Face Detection

In the face detection stage, the estimators are applied to probe image $\mathcal{I}$ in order to detect the face in it. In principle, the same moves as in the EBGM face detection step (see Section 3.2.1) are applied, and only the estimation of the similarities changes.

Global Move In the global move, again the best matching offset point $\vec{t}^{\,*}$ is obtained:

$$\vec{t}^{\,*} = \arg\max_{\vec{t}}\; S_{[A]}^{\mathrm{ML}}\left(\mathcal{G}^{\mathrm{ML}}, \mathcal{G}^{\mathcal{I}}\left(\vec{t}\right)\right)\,. \tag{3.3–17}$$



(a) Probe Image **(b)** Global Move **(c)** Scale Move

Figure 3.5: ML Face Detection Example: *This figure shows an exemplary result of the face detection steps employing maximum likelihood estimators: (b) the global move and (c) the scale move. For comparison, the probe image without any graph is given in (a).*

Similarly to the EBGM global move, only the absolute values of the Gabor jets are taken into account. Hence, the similarity score is calculated as:

$$S_{[A]}^{\mathrm{ML}}\left(\mathcal{G}^{\mathrm{ML}}, \mathcal{G}^{\mathcal{I}}\right) = \frac{1}{L}\sum_{l=0}^{L-1} S_{[A]}^{\mathrm{ML}}\left(\mathcal{J}_l^{\mathrm{ML}}, \mathcal{J}_l^{\mathcal{I}}\right) , \tag{3.3-18}$$

$$S_{[A]}^{\mathrm{ML}}\left(\mathcal{J}^{\mathrm{ML}}, \mathcal{J}^{\mathcal{I}}\right) = -\frac{1}{J}\sum_{j=0}^{J-1} \frac{\left(a_j - \mu_j^{[a]}\right)^2}{\kappa_j^{[a]}} , \tag{3.3-19}$$

in this case the normalization factors $\frac{1}{L}$ and $\frac{1}{J}$ could also be left out. Exemplary detection results of the maximum likelihood global move can be observed in Figure 3.5. Compared to the global move result of Figure 3.2(b), the result in Figure 3.5(b) is shifted vertically. In the former, the nodes of the mouth fit their image features, whereas in the latter, the eye landmarks are placed more appropriately.

Scale Move For the scale move, in turn the best matching scale $\vec{s}^{\,*}$ is scanned at the previously found best position $\vec{t}^{\,*}$:

$$\vec{s}^{\,*} = \arg\max_{\vec{s}} \; S_{[\cdot]}^{\mathrm{ML}}\left(\mathcal{G}^{\mathrm{ML}}, \mathcal{G}^{\mathcal{I}}\left(\vec{t}^{\,*}, \vec{s}\right)\right) , \tag{3.3-20}$$

where S^{ML} can use either absolute values only, in this case $S^{\mathrm{ML}}_{[\cdot]}$ is $S^{\mathrm{ML}}_{[A]}$ from Equation (3.3 – 19), or it can include phases, in which case:

$$S^{\mathrm{ML}}_{[P]}\left(\mathcal{J}^{\mathrm{ML}}, \mathcal{J}^{\mathcal{I}}\right) = -\frac{1}{2J}\sum_{j=0}^{J-1}\left[\frac{\left(a_j - \mu_j^{[a]}\right)^2}{\kappa_j^{[a]}} + \frac{d_\phi\left(\phi_j - \mu_j^{[\phi]}\right)^2 a_j}{\kappa_j^{[\phi]}\,\mu_j^{[a]}}\right] \quad (3.3-21)$$

need to be chosen. In turn, the factor $\frac{a}{\mu^{[a]}}$ was added empirically. The result of the scale move, which employed the $S^{\mathrm{ML}}_{[A]}$ estimator in this case, is shown in Figure 3.5(c). In comparison to the EBGM scale move (cf. Figure 3.2(c)) a different vertical scale was found and the scale of the graph fits better to the scale of the face.

3.3.5 Landmark Localization

Scan Local Move In the scan local move, the geometry costs are integrated into the similarity estimation:

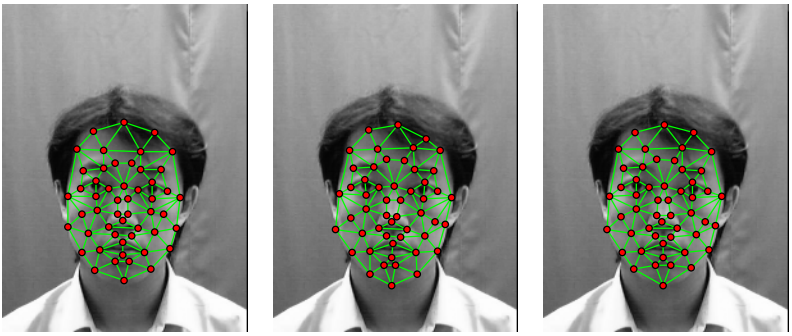
$$\begin{aligned} S^{\mathrm{ML}}_{[P,\mathcal{E}]}\left(\mathcal{G}^{\mathrm{ML}}, \mathcal{G}^{\mathcal{I}}\right) = \frac{1}{L}\sum_{l=0}^{L-1} S^{\mathrm{ML}}_{[P]}\left(\mathcal{J}_l^{\mathrm{ML}}, \mathcal{J}_l^{\mathcal{I}}\right) + \\ \frac{w_{\mathcal{E}}}{E}\sum_{e=0}^{E-1} S^{\mathrm{ML}}_{[\mathcal{E}]}\left(\mathcal{E}_e^{\mathrm{ML}}, \mathcal{E}_e^{\mathcal{I}}\right)\ . \end{aligned} \quad (3.3-22)$$

with the according geometry cost function:

$$S^{\mathrm{ML}}_{[\mathcal{E}]}\left(\mathcal{E}^{\mathrm{ML}}, \mathcal{E}^{\mathcal{I}}\right) = -\frac{1}{2}\left[\frac{\left(\Delta_h - \mu^{[\mathcal{E}_h]}\right)^2}{\kappa^{[\mathcal{E}_h]}} + \frac{\left(\Delta_v - \mu^{[\mathcal{E}_v]}\right)^2}{\kappa^{[\mathcal{E}_v]}}\right] \quad (3.3-23)$$

estimating the quality of the node positions, relative to their linked nodes. The weight of the geometry costs is different from the one of the EBGM algorithm (cf. Equation (3.2 – 4)). Since both parts of Equation (3.3 – 22) are dimensionless, it is more comprehensible (but unfortunately not easier) to set this weight. For the graphs shown in Figure 3.6, $w_{\mathcal{E}} = \frac{1}{2}$ was employed.

Disparity Local Move Equivalently to the disparity estimation with the bunch, the displacement can be estimated using the difference of the



(a) Scan Local Move **(b)** Disparity Local Move **(c)** Disparity Scan Local Move

Figure 3.6: ML Landmark Localization Example: *This figure shows three different versions of the maximum likelihood local move: (a) the scan local move, (b) the disparity local move, and (c) a combination of both.*

phases ϕ_j stored in the Gabor jet and the mean phases $\mu_j^{[\phi]}$ of the according $\mathcal{J}^{\mathrm{ML}}$:

$$S_{[D]}^{\mathrm{ML}}\left(\mathcal{J}^{\mathrm{ML}}, \mathcal{J}^{\mathcal{I}}\right) = -\frac{1}{2J}\sum_{j=0}^{J-1}\left[\frac{\left(a_j - \mu_j^{[a]}\right)^2}{\kappa_j^{[a]}} + \frac{d_\phi\left(\phi_j + \vec{k}_j^{\mathrm{T}}\vec{d} - \mu_j^{[\phi]}\right)^2}{\kappa_j^{[\phi]}}\right]. \tag{3.3-24}$$

Again, the details of the estimation of $\vec{d}$ are given in Appendix B. One problem of combining the disparity local move and the $\mathcal{J}^{\mathrm{ML}}$ jets is that there is only one estimate[5] for each landmark and, thus, there is no way to include the geometrical relations into the resulting disparity estimation. Therefore, the example shown in Figure 3.6(b) was created without accounting for geometry. Nonetheless, most of the landmarks are placed accurately, even those with few texture information, e. g., the cheek landmarks.

[5] When using the bunch graph approach, there were B estimates, one for each Gabor jet $\mathcal{J}^{(b)}$ in the bunch.

Disparity Scan Local Move In turn, the disparity estimation and the scan local move can be combined into the disparity scan local move, an exemplary result is shown in Figure 3.6(c). Including texture and geometry information, the placement of the nodes is very good, but still not as good as hand-labeled node positions would be.

3.4 Multi-Scale Face Detection

Both algorithms described in Sections 3.2 and 3.3 have a certain limitation: The absolute size of the faces in the probe image has to fit to the size of the bunch graph or the accordant maximum likelihood graph, respectively, and also the in-plane rotation angle needs to be appropriate. This limitation originates from the fact that Gabor jets in different scales and angles look very different, an illustration of this issue is given in Figure 2.4.

A simple and very time consuming, but most exact approach to detect a scaled and rotated face would be to rotate and scale the probe image with some scales s and angles α, and to use the pair (s^*, α^*) with the maximum probability score. Moghaddam *et al.* [52] used a similar approach. They extracted parts of the image in different positions and scales (without dealing with in-plane rotation), transformed them into eigenspace, estimated the probability of the input patch to be a face, and computed a multi-scale saliency map.

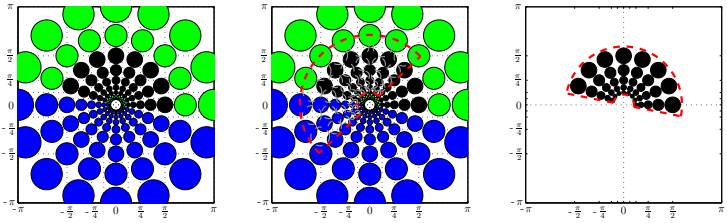
In the EGM schedule, this would require a full Gabor wavelet transform for each rotated and scaled version of the probe image. Since the GWT is one of the most time consuming steps of the schedule, this approach is absolutely inapplicable.

3.4.1 Gabor Jet Interpolation

As an alternative, it is also possible to interpolate the available Gabor wavelet responses stored in the Gabor jets. To do so, there is the need for more than the usual $\zeta_{\max} = 5$ levels of Gabor wavelets, which are, adumbrated with a red border, once again shown in Figure 3.7(c). Hence, the discrete Gabor wavelet family Γ (cf. Section 2.2.3) has to be extended to $\Gamma^{\{e\}} = (\nu_{\max}, \zeta_{\max}^{\{e\}}, k_{\max}^{\{e\}}, k_{\text{fac}}, \sigma)$ to include more Gabor wavelet levels:

$$\zeta_{\max}^{\{e\}} = \zeta_{\max} + 4 = 9\,, \qquad k_{\max}^{\{e\}} = 2k_{\max} = \pi\,, \tag{3.4–1}$$

more precisely each two levels of lower and higher frequency Gabor wavelets. These additional levels are represented by the green circles in Figure 3.7(a). The blue circles of Figure 3.7(a), which depict the Gabor wavelets



(a) Extended Family $\Gamma^{\{e\}}$ of Gabor Wavelets

(b) Indicated Transformation

(c) Common Family Γ of Gabor Wavelets

Figure 3.7: Gabor Jet Transformation: *This figure displays (a) the extended family* $\Gamma^{\{e\}}$ *of Gabor wavelets and (b) an indicated transformation with scale* 1.2 *and angle* -55°*, which is rotated back to (c) the common family* Γ *of Gabor wavelet responses.*

$\check{\psi}_{-\vec{k}_j}$ and cover the second half of the frequency domain, can be estimated from the green or black ones using the symmetry condition given in Equations (2.2 – 15) and (2.2 – 22), i. e., $\check{\psi}_{-\vec{k}_j}(\vec{\omega}) = \check{\psi}_{\vec{k}_j}(\text{-}\vec{\omega})$ and $\check{\mathcal{T}}_{-\vec{k}_j}(\vec{\omega}) = \overline{\check{\mathcal{T}}_{\vec{k}_j}(\text{-}\vec{\omega})}$, respectively.

The goal is to interpolate an *extended Gabor jet* $\mathcal{J}^{\{e\}}$ that includes responses of the extended Gabor wavelet family $\Gamma^{\{e\}}$ into a Gabor jet $\mathcal{J}$ of the common Gabor wavelet family Γ, i. e., to transform the indicated gray circles from Figure 3.7(b) back to the black circles shown in Figure 3.7(c). Recalling that $\mathcal{J}^{\{e\}}$ includes one complex-valued response for each Gabor wavelet, the circles in Figure 3.7(b) can be interpreted as the entries of the Gabor jet $(\mathcal{J}^{\{e\}})_j$. The missing values, i. e., the blue circles, which specify the responses and which are called $\left(\mathcal{J}^{\{e\}}\right)_{-j}$ for analogy, can be calculated as: $\left(\mathcal{J}^{\{e\}}\right)_{-j} = \overline{\left(\mathcal{J}^{\{e\}}\right)_j}$.

Each of the gray circles of Figure 3.7(b) is surrounded by four colored circles. Hence, the value of the Gabor jet $(\mathcal{J})_j$ can be approximated by a linear combination of four responses of the extended Gabor jet:

$$(\mathcal{J})_j = \sum_{i=1}^{4} w_i \left(\mathcal{J}^{\{e\}}\right)_{j_i} , \qquad (3.4-2)$$

totalizing real and imaginary values of the complex-valued Gabor wavelet responses independently. The weights $w_1, \ldots, w_4$ are calculated from the

scale s and the angle α such that the rotation angle is interpolated linearly, while the interpolation of the scale is logarithmically to the basis k_{fac}:

$$\begin{aligned} w_{\text{higher}} &= \log_{k_{\text{fac}}}(s) \mod 1\,, & w_{\text{lower}} &= 1 - w_{\text{higher}}\,, \\ w_{\text{left}} &= \frac{\nu_{\max}}{180^\circ}\,\alpha \mod 1\,, & w_{\text{right}} &= 1 - w_{\text{left}}\,, \end{aligned} \tag{3.4–3}$$

employing the real-valued modulo operator (cf. Equation (2.3–7)). Thus, rmod 1 keeps only the decimal places, the integral digits code for complete level or direction shifts. With the extended Gabor wavelet family $\Gamma^{\{e\}}$, scale factors in logarithmical range $s \in [\frac{1}{2}, 2]$ can be interpolated, while the rotation angle $\alpha \in [0^\circ, 360^\circ]$ can be chosen arbitrarily. Finally, the weights are computed as:

$$\begin{aligned} w_1 &= w_{\text{higher}}\, w_{\text{left}}\,, & w_2 &= w_{\text{higher}}\, w_{\text{right}}\,, \\ w_3 &= w_{\text{lower}}\, w_{\text{left}}\,, & w_4 &= w_{\text{lower}}\, w_{\text{right}}\,. \end{aligned} \tag{3.4–4}$$

Of course, the weights for all four surrounding Gabor wavelet responses sum up to unity.

3.4.2 Face Detection

The Gabor jet transformation can be used in the face detection step to detect faces in different scales and in-plane rotation angles. In this section, the terminology of the maximum likelihood face detection, i. e., the maximum likelihood graph $\mathcal{G}^{\text{ML}}$ is used, but the same procedure can be executed with the bunch graph $\mathcal{G}^{\mathcal{B}}$.

The approach that we proposed in [29] transforms $\mathcal{G}^{\text{ML}}$ according to various scales $\vec{s}$ and angles α to generate a conglomerate of $\mathcal{G}^{\text{ML}}(\vec{s}, \alpha)$. This transformation includes the landmark positions:

$$\mathcal{L}_l^{\text{ML}}(\vec{s}, \alpha) = Q(\alpha)\left(\mathcal{L}_l^{\text{ML}} \bullet \vec{s}\right) \tag{3.4–5}$$

and the Gabor jets of the training graphs, which, of course, need to have extended Gabor jets attached. Gabor jet transformation is applied into the direction α, the scale $s = \sqrt{s_h\, s_v}$ is calculated as the geometric mean of the horizontal scale s_h and vertical scale s_v. The graphs $\mathcal{G}^{\text{ML}}(\vec{s}, \alpha)$ are used to determine the best matching position $\vec{t}^{\,*}$, scale $\vec{s}^{\,*}$ and in-plane rotation angle α^*:

$$(\vec{t}^{\,*}, \vec{s}^{\,*}, \alpha^*) = \arg\max_{\vec{t}, \vec{s}, \alpha} S_{[A]}^{\text{ML}}\left(\mathcal{G}^{\text{ML}}(\vec{s}, \alpha)\,, \mathcal{G}^{\mathcal{I}}\left(\vec{t}, \vec{s}, \alpha\right)\right)\,, \tag{3.4–6}$$

(a) $s = 1, \alpha = 0°$ **(b)** $s = \sqrt{2}, \alpha =$ -45° **(c)** $s = 1.2, \alpha =$ -55°

Figure 3.8: Reconstructed $\mathcal{G}^{\mathrm{ML}}(\vec{s}, \alpha)$ Graphs: *This figure shows reconstructions of the mean absolute $\vec{\mu}^{[a]}$ and phase $\vec{\mu}^{[\phi]}$ values of $\mathcal{G}^{\mathrm{ML}}(\vec{s}, \alpha)$ in different scales $\vec{s} = (s, s)^{\mathrm{T}}$ and orientations α, including transformed Gabor jets. In (a) the upright graph in default scale $s = 1$ is shown, whereas (b) and (c) show rotated and scaled versions of the graph. In (b), Gabor wavelet responses are copied from other scales and angles, while (c) requires interpolation of Gabor wavelet responses.*

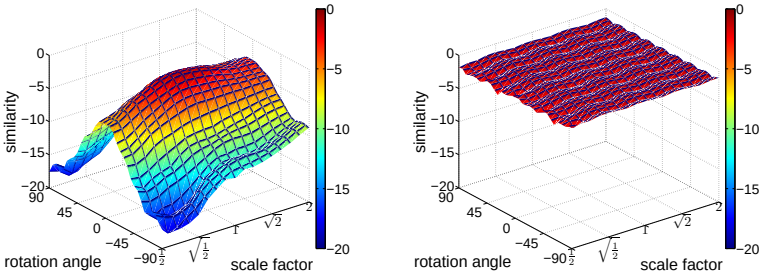
where the landmark positions of $\mathcal{G}^{\mathcal{I}}\left(\vec{t}, \vec{s}, \alpha\right)$ are calculated as:

$$\begin{aligned} \mathcal{L}_l^{\mathcal{I}}\left(\vec{t}, \vec{s}, \alpha\right) &= Q(\alpha)\left(\mathcal{L}_l^{\mathrm{ML}} \bullet \vec{s}\right) + \vec{t} \\ &= \mathcal{L}_l^{\mathrm{ML}}(\vec{s}, \alpha) + \vec{t}. \end{aligned} \tag{3.4-7}$$

Figure 3.8 shows image reconstructions (cf. Chapter 6) of average absolute and phase values for three $\mathcal{G}^{\mathrm{ML}}(\vec{s}, \alpha)$ graphs with different $(\vec{s}, \alpha)$-pairs, taking the same scale $s_h = s_v = s$ for horizontal and vertical graph scaling. These $\mathcal{G}^{\mathrm{ML}}(\vec{s}, \alpha)$ were generated by incorporating the information of 18 hand-labeled and standardized face graphs from nine men and nine women of the CAS-PEAL database [23], all showing a neutral facial expression. For our approach of [29], these graphs are matched directly against the input image to detect the best matching $(\vec{t}^{\,*}, \vec{s}^{\,*}, \alpha^*)$ triplet by employing a coarse-to-fine search.

Another way to deal with scaled and rotated faces in the image is to use a *scan scale and rotate global move*. This move, as the name suggests, scans the whole image for the best matching position $\vec{t}^{\,*}$, the best matching scale $\vec{s}^{\,*}$ and the best matching in-plane rotation angle α^*, but this time using the same $\mathcal{G}^{\mathrm{ML}}$, throughout:

$$(\vec{t}^{\,*}, \vec{s}^{\,*}, \alpha^*) = \underset{\vec{t}, \vec{s}, \alpha}{\arg\max}\ S_{[A]}^{\mathrm{ML}}\left(\mathcal{G}^{\mathrm{ML}}, \mathcal{G}^{\mathcal{I}}\left(\vec{t}, \vec{s}, \alpha\right)\right) . \tag{3.4-8}$$



(a) Untransformed Gabor Jet **(b)** Transformed Gabor Jet

Figure 3.9: Gabor Jet Similarities after Image Transformation: *This figure shows the* $S^{\mathrm{ML}}_{[A]}\left(\mathcal{G}^{\mathrm{ML}}, \mathcal{G}^{\mathcal{I}}(\vec{s}, \alpha)\right)$ *similarities obtained after rotating the input image. For (a) the Gabor jets were not transformed, for (b) the image scale and rotation was canceled out by Gabor jet transformation.*

Instead, the Gabor jets in the image graph $\mathcal{G}^{\mathcal{I}}$ need to be scaled and rotated. Therefore, the probe image needs to be Gabor wavelet transformed with the extended set $\Gamma^{\{e\}}$ since extended Gabor jets $\mathcal{J}^{\{e\}}$ are required for each position in the image. To be comparable to the upright $\vec{\mathcal{J}}^{\mathrm{ML}}$ in $\mathcal{G}^{\mathrm{ML}}$, the image Gabor jets $\mathcal{J}^{\mathcal{I}}$ are transformed into the opposite direction, i. e., employing rotation angle -α and scale $\frac{1}{\sqrt{s_h\, s_v}}$.

Figure 3.9 investigates, how well the scan scale and rotate global move detects scaled and rotated faces in a probe image that contains an identity that is not included in the training set. In a preparatory step, the probe image, which is the leftmost image shown in Figure 3.10(a), was standardized, i. e., fit to scale 1 and angle 0° of $\mathcal{G}^{\mathrm{ML}}$. Simulating a rotated and scaled face in the image, image $\mathcal{I}$ and graph $\mathcal{G}^{\mathcal{I}}$ were scaled and rotated equally with different scales and angles. Gabor jets were extracted at the new positions and the $S^{\mathrm{ML}}_{[A]}\left(\mathcal{G}^{\mathrm{ML}}, \mathcal{G}^{\mathcal{I}}(\vec{s}, \alpha)\right)$ probability scores were computed. For the map shown in Figure 3.9(a), the common Gabor jets in the image graph were taken unmodifiedly. Clearly, the S^{ML} similarity values decrease dramatically when scale and angle of the face in the input image do not fit to the ones of $\mathcal{G}^{\mathrm{ML}}$. Especially for rotation angles over 25°, correct detection is highly improbable. For Figure 3.9(b) the extended Gabor jets in the graph were transformed as described above. Obviously, the Gabor jet transformation canceled out the affine image transformation, the similarity values in Figure 3.9(b) remain high for all investigated scales and angles.

Comparably to the face detection schedule of the EBGM algorithm, this new approach is used in a multi-step coarse-to-fine search for the optimal $(\vec{t}^*, \vec{s}^*, \alpha^*)$ triplet. The first scan scale and rotate global move uses only few scales and angles on sparse grid positions to get an estimate of the best parameters, while the next steps refine the previously found triplets. One advantage of this schedule is that position, scale, and angle are tested at the same time and, thus, graph placement errors caused by scaling or rotation of the graph can be annihilated right away, which is not the case in the original EBGM schedule. One disadvantage of this move is an increased number of S^{ML} estimations.

The two approaches proposed in [29] and in this thesis should work approximately equally well, but they have very different execution times and memory footprints. The former needs either to re-train the mean and variance vectors for each considered (s, α) pair every time it is required, including Gabor jet transformations of the Gabor jets of the training set, or to store these values. In opposition, the latter algorithm needs only a single training stage, which could be done off-line, but the Gabor jet transformation has to be applied to each Gabor jet of the probe image for every (s, α) pair. Nonetheless, empirically the second algorithm works faster, requires less memory, and is easier assimilable into the existing EBGM algorithm. Hence, the second approach is used in the experiments of this thesis, although latest tests show that the former method might be more reliable (cf. Section 5.3.3 and [29]).

In his PhD thesis, Lades [44] proposed a similar approach to Gabor jet transformation, though there are some major differences to my approach. Firstly, Lades used linear weights for the interpolation between different Gabor wavelet scales, while in Equation (3.4 – 3) scales are interpolated logarithmically. Secondly, Lades only interpolated the absolute Gabor wavelet responses, ignoring the phases. While absolute values can be simply copied in the second half of the frequency domain, the complex responses need to be conjugated. Furthermore, one important point is that complex values are interpolated using the algebraic form with real and imaginary values, not the polar form with absolutes and phases. Additionally, Lades did not modify the Gabor wavelet family and, thus, had to fill in some 0 values for missing Gabor wavelet responses in higher or lower scales. Finally, Lades [44] proposed to detect scale, angle, and position independent of each other, i. e., first performing a global move, then a scale move and a rotate move afterwards. Due to the fact that Gabor jets in different scales and angles cannot be compared (cf. Section 2.3.2 and Figure 3.9), already the global move most probably fails for high scale or angle deviations.

Heintz *et al.* [33] scaled and rotated Gabor jet entries for multi-scale and

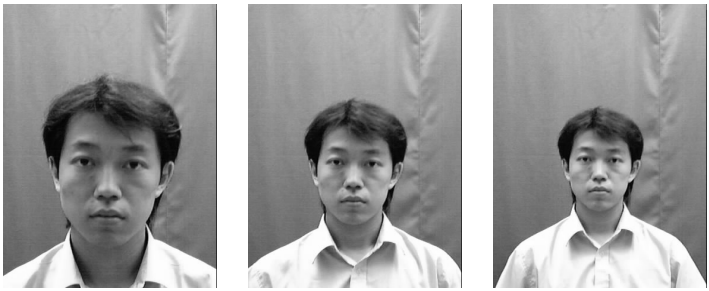
multi-angle object detection. In opposition to Lades and me, they did not interpolate the responses of the Gabor wavelets, but they computed cross-correlations between two Gabor jets to find out, which orientation and scale works best. Using this algorithms, their resulting scales and angles are limited to the discrete scale levels ζ and orientations ν of the Gabor wavelet family.

3.4.3 Image Standardization

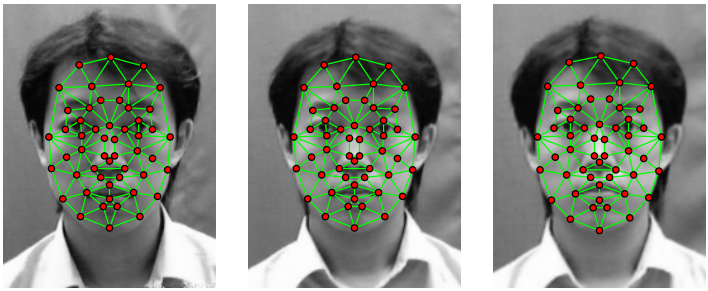
After detecting the face including scales $\vec{s}^*$ and angle α^*, the image is normalized according to these values, i. e., $\mathcal{I}$ is rotated according to -α^* and scaled with scale factor $\frac{1}{\sqrt{s_h^* \, s_v^*}}$ using the detected $\vec{t}^*$ as transformation center. If the face was found correctly, this transformation brings it to the same size as $\mathcal{G}^{\mathrm{ML}}$. Hence, after cropping the image to the training image size 168×224 pixel (cf. Section 2.4.3) and re-applying a Gabor wavelet transform with the common Γ-set, the Gabor jets of the probe image code texture of size comparable to $\vec{\mathcal{J}}^{\mathrm{ML}}$. Thus, the optimal situation is given, and the landmark localization step does not need further Gabor jet transformations.

The image normalization is also needed if the resulting face graphs should be used for face recognition, e. g., as described in Chapter 4. When the gallery graphs and the probe graphs store Gabor jets including texture information of different image sizes, the Gabor jets are hardly comparable and, thus, a Gabor jet of the same image size would most probably be preferred over the Gabor jet of the same identity, as we showed empirically in [29].

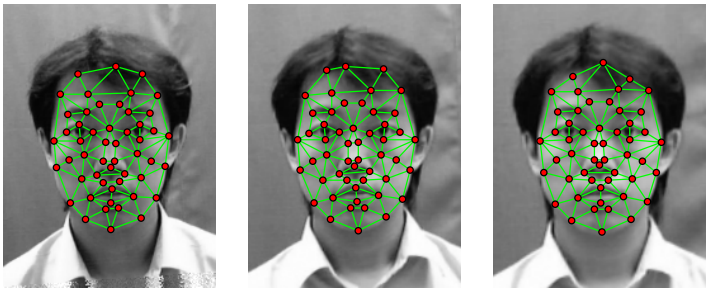
Since scale and angle estimated during face detection might be slightly incorrect, e. g., because the scale was under sampled, another image normalization might be executed after the landmark localization, now standardizing according to the found eye position as described in Section 2.4.3. Figure 3.10 displays exemplary results of the multi-scale face detection and the subsequent image normalization according to the detected scale and angle. The graphs shown in Figures 3.10(b) and 3.10(c) were detected using a $\mathcal{G}^{\mathrm{ML}}$ graph that integrates the information of 18 hand-labeled face graphs of neutral facial images. Since the landmark positioning step is executed on images normalized for scale and rotation, the node placement errors are, apart from possible misdetections and image scaling artifacts, independent of the size of the face in the original image. After landmark localization, the image is normalized again, the result is shown in Figure 3.10(c). Clearly, the scales of the images suit much better than the ones shown in Figure 3.10(b), which are normalized after the face detection step.



(a) Original Images



(b) Detected Faces



(c) Placed Landmarks

Figure 3.10: Detection of Scaled Faces: *This figure shows (a) the original images, (b) graphs and images that were automatically standardized according to the detected scale and angle, and (c) the graphs after landmark localization and the final image standardization according to the detected eye positions.*

Chapter 4

Face Recognition and Facial Property Classification

Automatic face recognition became famous in the last couple of years. It is used in a broad variety of tasks, like *access control*, *database search* or *monitoring*.

In the access control scenario, e. g., in high security areas in hospitals, each person that wants to access the secured area needs to verify his or her identity. Commonly, the *client* has an ID card storing a password or PIN that must be entered into a terminal. Of course, an *impostor* stealing the ID card and spying out the PIN can enter the area, too. Hence, it is more secure to store personalized characteristics, i. e., biometric information like a fingerprint, a voice sample, or a facial image of the client onto the ID card. Instead of entering a PIN, a fingerprint, a voice sample, or a photo is taken and compared with the stored one by calculating the similarity of both items. If the similarity exceeds a certain threshold Θ, the person is accepted as a client. The idea of verifying identity by biometric information comparison can also be extended to similar scenarios, e. g., replacing the PIN on the cash card by a facial image.

A similar approach can be used when the clients do not have an ID card, but a central *gallery* database stores all clients. In this case, the similarity between the person and all clients in the gallery is computed. A fully automatic system tests if the highest similarity value is above threshold Θ and, thus, the person can be identified. Unfortunately, similarity values from different people vary and, thus, some problems arise. Doddington *et al.* [16] give a short overview of those cases, which are subsumed under the term *biometric zoo*. Therefore, usually half-automatic systems are used, where the gallery clients that gain the highest similarity values are returned to security staff that decides manually.

The most challenging task is the monitoring, e. g., of public places, where people of public interest should be detected. This is especially hard since the

people in the observed area do not cooperate. This also makes it impossible to use other biometric features than the face. Of course, surveillance has also a legal issue. During the Super Bowl 2001, the football stadium was scanned for people from a mugshot database [17], the City of Tampa won the 2001 US Big Brother Award for spying on all of the Super Bowl attendees. A more legitimate experiment was made by the German Bundeskriminalamt [11] in 2007, where the faces of a group of participants were collected into a gallery. During a four month period, the participants were asked to walk through the eye of the camera every once in a while. The results show that the recognition accuracy is heavily dependent on the time of day. While at noon-time with natural light most of the participants were detected correctly, the recognition accuracy dropped dramatically in the night under artificial lighting conditions, or during winter when the participants wore scarfs covering parts of their faces.

On an irregular basis, the *National Institute of Standards and Technology* (NIST) performs *face recognition vendor tests* (FRTV) that evaluates current state-of-the-art face recognition algorithms. Since the FRVT tests commercial face recognition systems, unfortunately the employed methodology is not available. In the first test in 1994 [64], the FERET database consisted of 831 gray level gallery images and probe sets in different severities. The resulting recognition rates were around 90% for the easiest tasks, i. e., comparing neutral vs. neutral frontal faces in controlled illumination, and dropped below 60% when gallery and probe images were taken at different days. Comparing frontal images with quarter or half profile images did not work at all, the recognition rates were below 30% or 10%, respectively. In the following FRVT evaluations in 2000 [5], 2002 [63], and 2006 [66], color images in high resolution were provided, the number of images in gallery and probe was increased, the difficulty was raised, e. g., by including out-door images, 3D images were used, and finally the evaluation method was changed from *recognition rates* to *verification rates* (see next section). During that period, commercial face detection and recognition algorithms advanced greatly in terms of recognition accuracy, and the recognition of frontal facial images with neutral expression and controlled lighting conditions is nowadays practically solved. Still, the recognition rates under uncontrolled illumination conditions is quite poor, but O'Toole *et al.* [62] and Phillips *et al.* [66] showed that humans cannot handle these types of images either and that state-of-the-art face recognition systems already surpass human face recognition capabilities on those images.

4.1 Quality Measures

To be able to compare face recognition algorithms off-line, face databases were generated, including at least two images per person and *ground truth* (GT) information about the identities shown in the images. Most often, there is additional information about the person in the image, or about the conditions under which the image was taken. Finally, nearly all databases include hand-labeled eye positions of all faces. The databases that are used in the experiments described in this thesis are presented in Section 5.1.

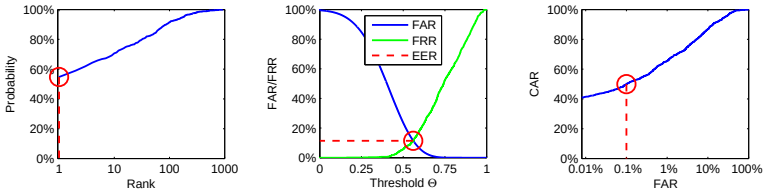
Some of these face databases also include default experimental setups, which usually split up the database into two different groups: *training* images and *test* images. The training images are used to train the recognition system, if it needs training at all. Mostly, the identities of the training images are unique, i. e., they are not part of test subset. The test subset is used to compute a quality measure, two of which will be described in this section, a complete overview of quality measures can be obtained from the National Science and Technology Council [59]. The test set is often once more split up into so-called *gallery* and *probe* subsets, where usually all *neutral* images (cf. Chapter 5) are put into the gallery: $\{\mathcal{I}^{(g)} \mid g = 0, \ldots, G\text{-}1\}$, and all others serve as probe images: $\{\mathcal{I}^{(p)} \mid p = 0, \ldots, P\text{-}1\}$. For each of the probe images, the similarities $S\left(\mathcal{I}^{(g)}, \mathcal{I}^{(p)}\right)$ to all gallery images are computed. These similarities are used to define the recognition accuracy in one of the two following ways:

4.1.1 Cumulative Match Characteristics

The *cumulative match characteristics* (CMC) is a rather old, but easy quality measure. It is especially designed to classify half-automatic database search algorithms. It lists, how many probe images have their corresponding gallery image recognized amongst the r most probable gallery images. For each probe image $\mathcal{I}^{(p)}$, the *rank*:

$$r = \left| \left\{ \mathcal{I}^{(g)} \;\middle|\; S\left(\mathcal{I}^{(g)}, \mathcal{I}^{(p)}\right) \geq S\left(\mathcal{I}^{(g^*)}, \mathcal{I}^{(p)}\right), g \in \{0, \ldots, G\text{-}1\} \right\} \right| \quad (4.1-1)$$

of its corresponding gallery image is calculated by computing the similarity of the probe image to all gallery images and count, how many gallery images of other identities are more similar than the image $\mathcal{I}^{(g^*)}$ that shows the same identity. If the gallery image $\mathcal{I}^{(g^*)}$ is most similar, rank $r = 1$ is assigned, while rank $r = 5$ means that four other gallery images were more similar. Hence, the worst rank a probe image can get is identical to the number of gallery images.



(a) CMC Curve **(b)** FAR/FRR Curve **(c)** ROC Curve

Figure 4.1: Quality Measures: *This figure displays examples of two popular quality measures, namely (a) the CMC and (c) the ROC, both with logarithmic axis of abscissa. The FAR/FRR curve in (b) is used to generate the ROC curve shown in (c).*

The *match characteristic* totalizes the achieved ranks of all probe images divided by the size P of the probe set. Therefore, the rank r match characteristic accounts for the probability that a probe item has rank r. To generate the CMC curve, the match characteristic is cumulated, starting at rank 1. Thus, rank r in the CMC curve specifies the probability for a probe image to be at rank r *or less*. The rank 1 characteristic is also called the *recognition rate* (RR), this value is often used when two face recognition algorithms are compared. Since match characteristics are computed by normalizing out probe size P, but not gallery size G, the recognition rate is dependent on G, but independent of P. Figure 4.1(a) displays an exemplary CMC curve with the recognition rate marked by a red circle. This CMC curve was generated on a database of 1000 identities, each of which had one image in the gallery and one in the probe subset.

4.1.2 Receiver Operating Characteristics

The *receiver operating characteristics* (ROC) presents a more recent quality measure that is designed to classify fully-automatic access control algorithms. During access control, there are four possible cases: A client can be accepted (*correct acceptance*, CA) or rejected (*false rejection*, FR), and an impostor can be rejected (*correct rejection*, CR) or accepted (*false acceptance*, FA). Hence, the threshold Θ should be chosen to grant access to as many as possible clients, i. e., the *correct acceptance rate* (CAR), which is also called the *verification rate* (VR), is maximized without allowing impostor access, corresponding to a vanishing *false acceptance rate* (FAR). The relations between

these four rates are:

$$\mathrm{FAR} = 100\% - \mathrm{CRR}\,, \qquad \mathrm{FRR} = 100\% - \mathrm{CAR}\,. \tag{4.1-2}$$

A typical FAR/FRR curve with both curves plotted against the threshold Θ is shown in Figure 4.1(b). With a low Θ, many impostors are accepted and the FAR is high, while with a higher Θ value, many clients are rejected, i. e., the FRR increases. One criterion for a good FAR/FRR curve is a low *equal error rate* (EER), i. e., the FAR or FRR value at the intersection of both curves.

The ROC curve finally plots the CAR against the FAR, making the measure independent of the value range of Θ and of the sizes of gallery and probe subsets. Figure 4.1(c) shows the ROC curve that was generated from the FAR/FRR curve from Figure 4.1(b), which in turn is based on the same recognition experiment as Figure 4.1(a). When a single value is needed, e. g., for comparing different recognition algorithms, usually the CAR at FAR = 0.1% is chosen. This value corresponds to the statistical probability for granted client access when 999 out of thousand impostors are correctly rejected.

4.2 Popular Face Recognition Algorithms

The presented quality measures need to compare two facial images, computing the similarity of the two faces as a single value. In the last decades, two main streams of face comparison algorithms became popular: Algorithms based on Gabor graphs and eigenface based algorithms. Two completely new types of features, which are briefly described afterwards, came to the center of attention lately.

4.2.1 Elastic Graph Matching

The first type of image comparison algorithms is based on the Gabor graphs presented in section Section 2.4.1. The *elastic graph matching* (EGM) algorithm is an integrated face detection, feature extraction, and face recognition system. The idea of the EGM algorithm is to compute the similarity of gallery and probe image $S\left(\mathcal{I}^{(g)}, \mathcal{I}^{(p)}\right)$ by matching gallery graph $\mathcal{G}^{(g)}$ onto the current probe image $\mathcal{I}^{(p)}$. Hence, for each gallery graph, the best matching position in the probe image is estimated. In this case, the gallery graphs can be grid graphs as used by Buhmann *et al.* [10] or Lades *et al.* [43], hand-labeled face graphs, or face graphs extracted by the elastic bunch graph matching algorithm (cf. Section 3.2).

One disadvantage of this algorithm is that the best position of the graph needs to be estimated for each gallery graph individually, which requires much computational time, and eventually some hand-labeling of gallery graphs. Hence, for big galleries, this algorithm is not feasible.

4.2.2 Gabor Graph Comparison

A faster alternative is to extract face graphs with the EBGM algorithm for both gallery and probe images. How this detection algorithm can be improved to allow more image variations is discussed in Sections 3.3 and 3.4. After extraction, face image comparison is simply achieved by comparing the corresponding face graphs. Basically, there are two different approaches to compare two graphs. The first family of methods compares the graph structure, i. e., the landmark positions $\vec{\mathcal{L}}$ or the edges $\vec{\mathcal{E}}$. A small adaptation of an already known function (cf. Equation (3.2 – 4)) is:

$$D^{\mathcal{G}}_{[\mathcal{E}_{h/v}]}\left(\mathcal{G},\mathcal{G}'\right) = \frac{1}{E}\sum_{e=0}^{E-1}\frac{\left(\vec{\Delta}_e - \vec{\Delta}'_e\right)^2}{\left(\vec{\Delta}_e\right)^2 + \left(\vec{\Delta}'_e\right)^2}\,, \tag{4.2-1}$$

which computes the edge difference, normalized by the length of the edges to punish deviations of short edges more than deviations of large ones. Another edge comparison function calculates the distance:

$$D^{\mathcal{G}}_{[\mathcal{E}]}\left(\mathcal{G},\mathcal{G}'\right) = \frac{1}{E}\sum_{e=0}^{E-1}\frac{\left|\left(\vec{\Delta}_e\right)^2 - \left(\vec{\Delta}'_e\right)^2\right|}{\left(\vec{\Delta}_e\right)^2 + \left(\vec{\Delta}'_e\right)^2} \tag{4.2-2}$$

as the difference of the edge length, again normalized by their sum. One difference between Equations (4.2 – 1) and (4.2 – 2) is that the former punishes graph rotation, whereas the latter does not.

More common Gabor graph comparison functions use the Gabor jets attached to the nodes. They usually compute the similarity of two graphs:

$$S^{\mathcal{G}}_{[\cdot]}\left(\mathcal{G},\mathcal{G}'\right) = \frac{1}{L}\sum_{l=0}^{L-1} S_{[\cdot]}\left(\mathcal{J}_l,\mathcal{J}'_l\right) \tag{4.2-3}$$

as the average of the Gabor jet similarities of the nodes. In Equation (4.2 – 3), $S_{[\cdot]}$ can be any Gabor jet comparison function like the ones introduced in Section 2.3.1. Originally, only the $S_{[A]}$ function was used for recognition, but

experiments (cf. Chapter 5) show that other functions like $S_{[C]}$ on average out-perform $S_{[A]}$, sometimes even functions including phase information like $S_{[P]}$ and $S_{[D]}$ are well suited for recognition.

4.2.3 Eigenfaces

Algorithms of the second type recognize facial images based on comparisons of gray values of corresponding pixels. Of course, this can only work when the images have the same resolution, and it is a good idea to have the faces in the image normalized such that corresponding facial features are located at identical pixel positions. Usually, these algorithms use the hand-labeled eye positions to normalize the images [4, 107, 53, 51] and an elliptical mask to cut off background pixels.

A very simple ansatz of computing the distance $D(\mathcal{I}, \mathcal{I}')$ of two facial images would be to totalize the pixels gray value differences:

$$D\big(\mathcal{I}, \mathcal{I}'\big) = \sum_{\vec{x}} \big(\mathcal{I}(\vec{x}) - \mathcal{I}'(\vec{x})\big)^2 . \tag{4.2–4}$$

Since the distance value is independent of the relation between neighboring pixels, each of the images can be aligned into one long vector $\vec{v}$ of size N, and it holds:

$$D\big(\mathcal{I}, \mathcal{I}'\big) = D\big(\vec{v}, \vec{v}'\big) = \sum_{n=0}^{N-1} \big(v_n - v'_n\big)^2 = \|\vec{v} - \vec{v}'\|^2 . \tag{4.2–5}$$

Instead of computing the distance in the original image space, it is also possible to perform a coordinate system transformation using orthonormal transformation matrix Φ, after subtracting the mean vector $\vec{\mu}$:

$$\vec{y} = \Phi^{\mathrm{T}} \left(\vec{v} - \vec{\mu}\right) . \tag{4.2–6}$$

The transformation matrix and the mean vector result from the *principal component analysis* (PCA), which is detailed in Section 3.1.1, using a lot of training images. The matrix Φ consists of eigenvectors $\vec{\Phi}_n$ that are sorted according to their corresponding eigenvalues λ_n in decreasing order. Still the Euclidean distance keeps the same [106]:

$$\begin{aligned} D\big(\vec{y}, \vec{y}'\big) &= \|\vec{y} - \vec{y}'\|^2 \\ &= \|\Phi^{\mathrm{T}} \left(\vec{v} - \vec{\mu}\right) - \Phi^{\mathrm{T}} \left(\vec{v}' - \vec{\mu}\right) \|^2 \\ &= \|\Phi^{\mathrm{T}} \left(\vec{v} - \vec{v}'\right) \|^2 \\ &= \|\vec{v} - \vec{v}'\|^2 , \end{aligned} \tag{4.2–7}$$

although it is now calculated in eigenspace.

Since the eigenvectors $\vec{\Phi}_n$, when reinterpreted as images, look like faces, they are are often called *eigenfaces* [89]. The first eigenfaces, which correspond to the directions of the maximal variances in the face space, are said to code for identity-unspecific image attributes like illumination, the latest eigenfaces only contain noise. Yambor *et al.* [107] show some eigenfaces that strengthen that belief. Hence, it is possible to use only a subset of the eigenvectors, i. e., by shortening the transformed vector $\vec{y}$ to length $M \ll N$ to compute the distance, hoping that recognition accuracy increases. Unfortunately, it is very complicated to select the best eigenvectors and the choice is different for different databases.

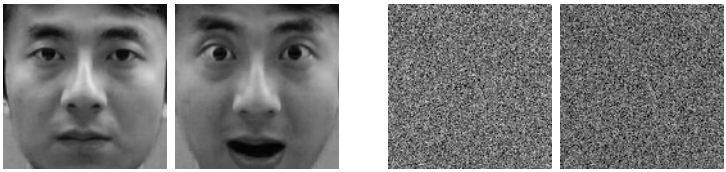
Another weak point of Equation (4.2 – 7) is that the Euclidean distance measure might not be the best choice. Yambor *et al.* [107][1] and many, many others tried different distance measures and reported results on different databases, but still there is no real conclusion, which distance metric is the best [4, 15]. Popular functions are the Euclidean distance, the Manhattan distance, or the Mahalanobis distance:

$$\begin{aligned} \|\vec{y} - \vec{y}'\| &= \sum_{n=0}^{M-1} \left(y_n - y'_n\right)^2 , \\ \|\vec{y} - \vec{y}'\|_1 &= \sum_{n=0}^{M-1} |y_n - y'_n| , \\ \|\vec{y} - \vec{y}'\|_\Lambda &= \sum_{n=0}^{M-1} \frac{\left(y_n - y'_n\right)^2}{\lambda_n} , \end{aligned} \tag{4.2 – 8}$$

respectively. In the Mahalanobis distance, the difference of y-values are divided by the eigenvalues Λ obtained during PCA, see also Equation (3.3 – 4).

Finally, each pixel of $\mathcal{I} \mathrel{\hat{=}} \vec{v}$ spans its own dimension in the input space, but of course it is not independent of the neighboring pixel values. Hence, there is the need to re-introduce proximity relations by computing co-variances. The images shown in Figure 4.2 present two versions of image standardization for PCA. In Figure 4.2(a), the images are aligned at the eye positions and cut to image size 128×128 pixel. The images in Figure 4.2(b) are identical, but the placement of the pixels is changed in a defined order. Although the second images do not seem to contain faces, both image standardizations are identical to eigenface based recognition algorithms, the

[1] The Mahalanobis distance that Yambor *et al.* [107] propose is wrong (see also [4]), but still performs good.



(a) Standardized Images

(b) Controlledly Shuffled Versions

Figure 4.2: Standardized Images for PCA: *This figure displays (a) standardized images as used for eigenface computation, and (b) regularly shuffled version of them, containing the same information.*

distance of the two images in Figure 4.2(a) is identical to the distance of the images in Figure 4.2(b).

4.2.4 Linear Discriminant Analysis

A more sophisticated approach tries to learn not only how faces look and how they are distributed, but also which image variations are *intrapersonal* and which are *extrapersonal*. Intrapersonal image differences are typically illumination changes or facial expressions, while extrapersonal variations code the facial features that enable the discrimination between faces of different identities, although, of course, lighting conditions or facial expressions as well play a role. The *linear discriminant analysis* (LDA) [19] calculates the *within class scatter matrix* $\mathcal{S}_w$ and the *between class scatter matrix* $\mathcal{S}_b$ using the identity information from the face database:

$$\mathcal{S}_w = \sum_{c=0}^{C\text{-}1} \mathcal{P}(c)\ \Sigma_c\,, \tag{4.2-9}$$

$$\mathcal{S}_b = \sum_{c=0}^{C\text{-}1} \mathcal{P}(c)\ (\vec{\mu} - \vec{\mu}_c)\,(\vec{\mu} - \vec{\mu}_c)^{\mathrm{T}}\ . \tag{4.2-10}$$

Here, $\mathcal{P}(c)$ is the prior probability of identity $c \in \{0, \ldots, C\text{-}1\}$, $\vec{\mu}_c$ the averaged face of identity c, and Σ_c the covariance matrix of all images of that identity, while $\vec{\mu}$ as before is the average of all faces. Hence, the $\mathcal{S}_w$ matrix codes the intrapersonal scatter, while $\mathcal{S}_b$ holds the distance of the identity means to the overall mean.

Having these scatter matrices, the goal is to find a linear transformation that minimizes the impact of intrapersonal changes and maximizes that of extrapersonal variations. This goal is reached by minimizing:

$$\frac{\Phi^{\mathrm{T}} \mathcal{S}_b \Phi}{\Phi^{\mathrm{T}} \mathcal{S}_w \Phi} \tag{4.2-11}$$

solving the generalized eigenvector equation [19]:

$$\mathcal{S}_b \, \Phi = \Lambda \, \mathcal{S}_w \, \Phi \,, \tag{4.2-12}$$

i. e., by computing the eigenvector matrix Φ and the corresponding eigenvalue matrix Λ of $\mathcal{S}_b^{-1}\mathcal{S}_w$. The number of non-zero eigenvalues is one less than the number of identities: C-1 [19]. Honoring the inventor, the eigenvectors are called *Fisher faces*.

The recognition task is identical to the eigenface approach. Both images $\vec{v}$ and $\vec{v}'$ are projected into Fisher space: $\vec{y} = \Phi^{\mathrm{T}} \vec{v}$ and the distance is computed between $\vec{y}$ and $\vec{y}'$. Again, a variety of distance measures were tested including the ones from Equation (4.2 – 8), and none of them seem to be superior to each other one [4, 15].

Some face recognition algorithms that are based on Fisher faces use a two-step transformation, employing a mixture of eigenfaces and Fisher faces. First, for dimensionality reduction and noise removal, the input vectors are transformed into eigenspace. On top of these transformed vectors, an LDA is computed, i. e., these vectors are further projected from face space into *Fisher space*. Zhao *et al.* [112] show some pure Fisher faces and some Fisher faces based on PCA. Although this does not necessarily have an impact on recognition accuracy, the latter ones look much better since they are much less noisy.

Employing LDA for face recognition has a major inconsistency in the basic idea. Since LDA is a classification algorithm, it normally needs to know the classes, i. e., the identities of the gallery images during training, contradicting current face recognition test standards that require independent training and test identities. The transformation matrix Φ is trained such that the identities of the training set are well-separated in Fisher space and, hence, hyper-regions in Fisher space are assigned to specific identities. Unfortunately, this is not the case for an image of a probe identity that was not in the training set. The probe image is projected into Fisher space, and its projection is hopefully near the projection of the gallery image of the same identity. It turns out that the recognition accuracy is above the pure PCA-based transformation only when images of all test identities are in the training set [23]. In opposition, when training and test identities are disjoint,

LDA performs even worse than PCA [4, 15]. In any case, the correct choice of training parameters is fundamental for good recognition.

4.2.5 PCA and LDA on Gabor Wavelet Responses

Instead of using the pixel gray values, Gabor wavelet responses, or more precisely the absolute values of these, can be used as a basis for the principal component analysis and an optionally following linear discriminant analysis. Gao *et al.* [23] use the same image normalization as described for the eigenfaces (cf. Section 4.2.3) and extracts Gabor jets at node positions from a 15×15 element grid. The absolute values of the Gabor wavelet responses are aligned into one vector of dimensionality $15 \times 15 \times 40 = 9000$, after they were normalized[2].

These 9000 element vectors are now treated as input vectors $\vec{v}$ for the PCA and the following LDA, the procedure is identical. The results of the combination of Gabor wavelet responses and a statistical learning seem to be superior to LDA applied to eigenfaces [23], but still the number of kept eigenvalues and the which distance function to employ are big issues that are not solved by this approach, either.

4.2.6 Scale Invariant Feature Transformation

SIFT features are usually exploited for rotation and scale invariant object detection and recognition in cluttered scenes [46, 47]. This is done by estimating position, scale, and orientation of the object that most of the SIFT features vote for, see Section 3.1.4 for details. Nonetheless, there are experiments using SIFT features for face recognition [2, 24, 41].

One important step in the SIFT feature extraction is the autonomous key point localization, which is independent of the intended task. Hence, the number and the location of SIFT features might differ between gallery and probe image. Aly [2] tried to use automatically detected SIFT features for face recognition. For each extracted SIFT feature, he classified the person by a nearest neighbor search of previously stored gallery SIFT features employing default distance metrics. Afterwards, he performed a majority vote over the extracted features to define his final recognition decision.

Križaj *et al.* [41] solved the issue by extracting SIFT features at specific locations in the image that are most often selected by the SIFT feature detector in the training set. To achieve SIFT feature correspondences, the

[2]Gao *et al.* [23] normalize each Gabor wavelet response to zero mean and unit variance, the normalization is not comparable to the Gabor jet normalization as given in Section 2.3.

images are aligned according to the eye positions. Instead of performing a majority voting, the average distance between accordant SIFT features is used for recognition. Križaj *et al.* [41] proved that their algorithm is better suited for identification under illumination variation than PCA, LDA, and the original SIFT approach.

4.2.7 Local Binary Pattern Histogram Sequence

A rather new approach [1] uses *local binary pattern histogram sequences* (LBPHS) for face recognition. For a given offset point, the local binary pattern is composed of the relations of the gray value of the pixel to the gray values of eight adjacent pixels. For each neighbor, a 1 is stored if the neighbor's gray value is greater than the central gray value, and 0 otherwise. Hence, an 8-bit vector of relations to the neighbors, the so-called *local binary pattern* (LBP), is given for each pixel in the image. The image is tessellated into several small rectangular regions, in which histograms of these local binary patterns are generated. These histograms are used to recognize the face shown in the image by employing some default histogram intersection measures. The results that Ahonen *et al.* [1] reported on the FERET database seem to out-perform the EBGM algorithm, ignoring the fact that the EBGM algorithm was used fully automatically, i. e., including face detection, while their LBP approach relies on hand-labeled eye positions.

Several extensions to LBP were made, one successful trial [110] extracted local binary patterns from absolute values of Gabor wavelet responses and built the *local Gabor binary pattern* (LGBP). Later, [111] also Gabor phases were included into the *enhanced local Gabor binary pattern*. Instead of a single histogram for each region, *histogram sequences* of 40 or 80 histograms are computed, one or two for each Gabor wavelet response, respectively. In combination with the small region size of 8×8 pixels, a huge number of histograms are generated. For the comparison of these histograms, two different methods are proposed: The first one totalizes histogram similarities of all histograms, while the second one calculates weights for each histogram by exploiting intrapersonal and extrapersonal means and variances of histogram similarities, following the idea of Moghaddam and Pentland [51] that is described in more detail in Section 4.3.

4.2.8 Ranking Lists

Müller [58, 57, 56] introduced a model-based recognition scheme that enables to indirectly compare face images that cannot (reliably) be compared directly, e. g., under pose variation. His assumption is: If two faces are similar in one

pose, they are also similar in another pose. Hence, given a model database, a ranking list of a frontal gallery image can be generated by computing the image similarities to all images of the model set that show frontal poses. Hence, the identity of the gallery image is no longer coded by a texture model, but by a ranking list of similarities to other identities. For a given probe image in profile view, the ranking list is again computed, this time comparing the profile probe image to profile images of the model set. Now, both the gallery as well as the probe image are characterized by ranking lists, which are compared by an appropriate ranking list comparison function.

There are some mayor advantages of Müller's approach. Apparently, the ranking lists can be computed using a variety of image comparison functions. Thus, virtually any function that can be used for face recognition can be further processed by ranking list. Müller [56] used Gabor graphs with different graph comparison functions, as well as *graph based local Gabor binary pattern histogram sequences* (GBLGBPHS). His recognition results comparing faces with different poses or under different illumination conditions are outstanding and outperform other state-of-the-art face recognition algorithms by far [57]. Fortunately, even when the pose or lighting condition of the probe image is not given, estimated poses or illumination conditions decrease the recognition accuracy only slightly.

4.3 The Intrapersonal/Extrapersonal Classifier

All recognition methods described in Section 4.2 have a common basis: each algorithm learns how to transform the probe image into a feature vector, where the transformation stage in most cases is very complex. The recognition of the face is done by comparing this feature vector to a gallery of similarly created feature vectors. For this comparison, default distance or similarity functions are used.

Hence, big efforts are made to generate feature vectors that ideally describe the face, but for the interesting part, i. e., the comparison of two feature vectors, the most simple algorithms are employed. This section describes a statistical algorithm that learns how to compare two feature vectors in order to enhance recognition accuracy. This algorithm basically can be used to compare any type of feature vector and also allows for the integration of different feature types or feature comparison functions.

4.3.1 Preliminary Work

One well known technique for the comparison of two facial images is the *Bayesian intrapersonal/extrapersonal classifier* (BIC) that was introduced by Moghaddam *et al.* [51]. It follows an ansatz that is comparable to PCA+LDA by exploiting the statistics of intrapersonal and extrapersonal image changes, using a non-linear Gaussian model. But instead of performing PCA on images themselves and calculating the difference in eigenspace, BIC investigates the statistics of image differences $\vec{u} = \vec{v} - \vec{v}'$ by performing PCA directly on $\vec{u}$.

Moghaddam *et al.* [51] defined two classes of image differences, the *intrapersonal class* Ω_I containing differences of images showing the same identity, and the *extrapersonal class* Ω_E comprising image differences of different persons. They calculated the *a posteriori* (AP) probability of image difference $\vec{u}$ belonging to the intrapersonal class Ω_I using Bayes rule:

$$\mathcal{P}(\Omega_I \mid \vec{u}) = \frac{\mathcal{P}(\vec{u} \mid \Omega_I)\ \mathcal{P}(\Omega_I)}{\mathcal{P}(\vec{u} \mid \Omega_I)\ \mathcal{P}(\Omega_I) + \mathcal{P}(\vec{u} \mid \Omega_E)\ \mathcal{P}(\Omega_E)}\,. \tag{4.3-1}$$

The likelihood $\mathcal{P}(\vec{u} \mid \Omega)$ of difference $\vec{u}$ given the class $\Omega \in \{\Omega_I, \Omega_E\}$ is estimated by [53, 51]:

$$\mathcal{P}(\vec{u} \mid \Omega) \approx \left[\frac{\mathrm{e}^{-\frac{1}{2}\sum_{n=0}^{M-1}\frac{y_n^2}{\lambda_n}}}{(2\pi)^{\frac{M}{2}}\prod_{n=1}^{M}\lambda_n^{\frac{1}{2}}}\right]\left[\frac{\mathrm{e}^{-\frac{\epsilon^2(\vec{u})}{2\rho}}}{(2\pi\rho)^{\frac{N-M}{2}}}\right], \tag{4.3-2}$$

which is identical to the probability Equation (3.3–8) that they[52] used for the maximum likelihood face detection presented in Section 3.3.1. The main difference to Section 3.3.1 is that the underlying PCA is not performed on images $\vec{v}$, but on image differences $\vec{u}$.

In fact, difference vector $\vec{u}$ can be the result of any kind of image comparison as long as the distribution of these values is approximately Gaussian. Notably, it can handle distance functions $D(\mathcal{I}, \mathcal{I}')$ as well as similarities $S(\mathcal{I}, \mathcal{I}')$. As an example, Moghaddam *et al.* [50] used deformable image models to define the difference vector $\vec{u}$ between two images and claimed that this method is better suited for face recognition than the direct image difference.

For both classes Ω_I and Ω_E, the likelihood from Equation (4.3–2) is computed separately. The likelihoods are estimated by exploiting the statistics of two distinct training sets:

$$\begin{aligned} T_I &= \{\vec{u}_I^{(b)} \mid b = 0, \ldots, B_I\text{-}1\}\,, \\ T_E &= \{\vec{u}_E^{(b)} \mid b = 0, \ldots, B_E\text{-}1\}\,, \end{aligned} \tag{4.3-3}$$

where $\vec{u}_I^{(b)}$ was generated as the difference of two images of the same person, while $\vec{u}_E^{(b)}$ is the result of comparing two images of different identities. B_I and B_E stand for the number of intrapersonal and extrapersonal image differences that are generated from the training set, respectively. For each of the two training sets, a separate PCA is performed. Hence, two transformation matrices Φ_I and Φ_E and their according eigenvalues Λ_I and Λ_E as well as the mean vectors $\vec{\mu}_I$ and $\vec{\mu}_E$ are calculated. Furthermore, Moghaddam *et al.* [51] used a different number of eigenvalues M_I and M_E in their experiments. Interestingly, they set their prior probabilities of the classes to be identical: $\mathcal{P}(\Omega_I) = \mathcal{P}(\Omega_E)$, although the sizes ($B_I = 74, B_E = 296$) of their reported training sets differ [51].

4.3.2 Simplifications

Teixeira [85], who simplified the maximum likelihood similarity scores for face detection (cf. Section 3.3.2), also noticed that the a posteriori probability calculation from Equation (4.3–1) can be rewritten as:

$$\begin{aligned} S^{\mathrm{AP}}(\vec{u}) = & \quad S^{\mathrm{ML}}(\vec{\mu}_I, \Sigma_I, \vec{u}) \quad - \quad S^{\mathrm{ML}}(\vec{\mu}_E, \Sigma_E, \vec{u}) \\ = & - \left[\sum_{n=0}^{M_I-1} \frac{y_{I;n}^2}{\lambda_{I;n}} + \frac{\epsilon_I(\vec{u})}{\rho_I} \right] + \left[\sum_{n=0}^{M_E-1} \frac{y_{E;n}^2}{\lambda_{E;n}} + \frac{\epsilon_E(\vec{u})}{\rho_E} \right]. \end{aligned} \tag{4.3–4}$$

In turn, the relation:

$$P(\Omega_I \mid \vec{u}_1) < P(\Omega_I \mid \vec{u}_2) \Longleftrightarrow S^{\mathrm{AP}}(\vec{u}_1) < S^{\mathrm{AP}}(\vec{u}_2) \tag{4.3–5}$$

holds. One important fact of these conversions is that the dependence on the class priors that Moghaddam *et al.* [51] did not specify satisfactorily is eliminated. In other words, Teixeira herewith removed the "Bayesian" part of the Bayesian intrapersonal/extrapersonal classifier.

Similar to Section 3.3, I replace the preparatory PCA calculations used in Equations (4.3–2) and (4.3–4) by the estimation of mean $\vec{\mu}$ and variance $\vec{\kappa}$ of the two classes, building the *intrapersonal/extrapersonal classifier* (IEC). The S^{IEC} similarity between two images is defined as:

$$S^{\mathrm{IEC}}(\vec{u}) = \sum_{n=0}^{N-1} \left[-\frac{(u_n - \mu_{I;n})^2}{\kappa_{I;n}} + \frac{(u_n - \mu_{E;n})^2}{\kappa_{E;n}} \right], \tag{4.3–6}$$

with $(\vec{\mu}_I, \vec{\kappa}_I)$ and $(\vec{\mu}_E, \vec{\kappa}_E)$ specifying mean and variance vectors of the two classes Ω_I and Ω_E, respectively, the calculation of these vectors is identical

to Equation (3.3 – 13):

$$\vec{\mu}_I = \frac{1}{B_I} \sum_{b=0}^{B_I\text{-}1} \vec{u}_I^{\,(b)}\,, \quad \vec{\kappa}_I = \frac{1}{B_I\text{-}1} \sum_{b=0}^{B_I\text{-}1} \left(\vec{u}_I^{\,(b)} - \vec{\mu}_I\right) \bullet \left(\vec{u}_I^{\,(b)} - \vec{\mu}_I\right)\,, \tag{4.3-7}$$
$$\vec{\mu}_E = \frac{1}{B_E} \sum_{b=0}^{B_E\text{-}1} \vec{u}_E^{\,(b)}\,, \quad \vec{\kappa}_E = \frac{1}{B_E\text{-}1} \sum_{b=0}^{B_E\text{-}1} \left(\vec{u}_E^{\,(b)} - \vec{\mu}_E\right) \bullet \left(\vec{u}_E^{\,(b)} - \vec{\mu}_E\right)\,.$$

In turn, unimodal Gaussian distributions of the difference vectors $\vec{u}_I$ and $\vec{u}_E$ are assumed.

As noted in Section 4.2.3, neighboring pixels tend to be correlated. By removing the PCA in the S^{IEC} similarity estimation, these correlations are disregarded. Therefore, this approach does most probably not work with direct image differences $\vec{u} = \vec{v} - \vec{v}'$. Instead, comparisons of Gabor graphs are used to define $\vec{u}$ vectors, some of them comparing texture by using Gabor jet similarities and some exploiting the graph structure. Again, the resulting S^{IEC} values are dimensionless allowing for easy integration of different data types. Notably, combining graph structure and texture comparison functions should help to increase recognition accuracy.

Furthermore, in the S^{IEC} environment it is possible to find outliers, e. g., graphs that are misplaced during face detection. The difference $\vec{u}$ of these outliers to *any* other graph is most often far from the intrapersonal mean $\vec{\mu}_I$ *and* far from the extrapersonal mean $\vec{\mu}_E$. Hence, if both $S^{\mathrm{ML}}(\vec{\mu}_I, \vec{\kappa}_I, \vec{u})$ and $S^{\mathrm{ML}}(\vec{\mu}_E, \vec{\kappa}_E, \vec{u})$ (cf. Equation (3.3 – 12)) are low (i. e., have highly negative values), $\vec{u}$ belongs to neither class and, thus, the corresponding graph might be refused in an automatic face detection and recognition system.

4.3.3 Graph Comparison Functions

The first group of comparison functions compares the texture of two images, or more precisely the Gabor jets of two face graphs that were detected in those images. The Graph similarity function $S^{\mathcal{G}}_{[\cdot]}$ from Equation (4.2 – 3) calculates the similarity of two graphs as the average of Gabor jet similarity at corresponding landmarks. Although the recognition results are quite good, they should be improvable by learning the distribution of these values, e. g., by studying, which similarities are affected by intrapersonal changes, and which code for extrapersonal variations. Hence, the first graph comparison function is the node-wise comparison of the Gabor jets:

$$u_l^{[\cdot]} = S_{[\cdot]}\left(\mathcal{J}_l, \mathcal{J}_l'\right) \tag{4.3-8}$$

resulting in a similarity vector $\vec{u}^{[\cdot]}$ of dimensionality $N^{[\cdot]} = L$, i. e., the number of landmarks of the graph. Again, $S_{[\cdot]}$ can be any Gabor jet comparison function, e. g., $S_{[A]}$ will result in the similarity vector $\vec{u}^{[A]}$. Please note that the graph similarity function $S^{\mathcal{G}}_{[\cdot]}$ can be rewritten as:

$$S^{\mathcal{G}}_{[\cdot]}\left(\vec{u}^{[\cdot]}\right) = \frac{1}{N^{[\cdot]}} \sum_{n=0}^{N^{[\cdot]}-1} u_n^{[\cdot]}\,, \tag{4.3-9}$$

while the novel S^{IEC} similarity function reads as:

$$S^{\mathrm{IEC}}_{[\cdot]}\left(\vec{u}^{[\cdot]}\right) = \sum_{n=0}^{N^{[\cdot]}-1} \left[-\frac{\left(u_n^{[\cdot]} - \mu_{I;n}^{[\cdot]}\right)^2}{\kappa_{I;n}^{[\cdot]}} + \frac{\left(u_n^{[\cdot]} - \mu_{E;n}^{[\cdot]}\right)^2}{\kappa_{E;n}^{[\cdot]}} \right]. \tag{4.3-10}$$

In Equation (4.3–10), $\vec{\mu}_I^{[\cdot]}, \vec{\mu}_E^{[\cdot]}, \vec{\kappa}_I^{[\cdot]}, \vec{\kappa}_E^{[\cdot]}$ are the results of the appropriate training (cf. Equation (4.3–7)) with function $S_{[\cdot]}$ using the two training sets T_I and T_E.

To see, whether the assumption of the Gaussian distribution of similarity values holds, Figure 4.3 shows exemplary distributions of some results of Gabor jet based similarity functions. The values were generated using the Gabor jets extracted at the nose-tip landmark, employing the complete FaceGen database (cf. Section 5.1.1) of artificial faces including 1000 identities with each two images per identity, resulting in 1000 intrapersonal and 999,000 extrapersonal graph comparisons. Obviously, intrapersonal and extrapersonal similarities have different distributions. Although the two classes overlap, where especially $S_{[P]}$ has many low intrapersonal similarity values, a combination of many of these distributions should boost recognition.

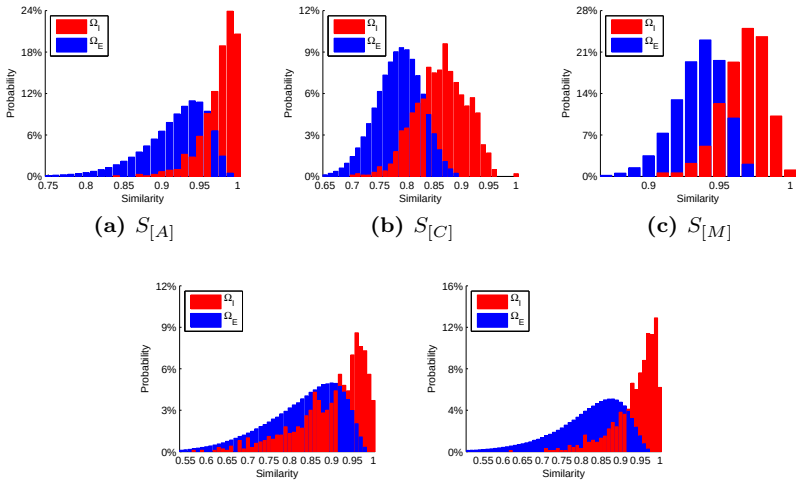
Another set of texture comparison functions using Gabor jets is the direct comparison of the absolute values of the Gabor wavelet responses, e. g.:

$$u_n^{[c]} = D_{[c]}\left(a_{l;j}, a'_{l;j}\right) = \frac{|a_{l;j} - a'_{l;j}|}{a_{l;j} + a'_{l;j}} \tag{4.3-11}$$

or:

$$u_n^{[a]} = D_{[a]}\left(a_{l;j}, a'_{l;j}\right) = |a_{l;j} - a'_{l;j}|\,, \tag{4.3-12}$$

with $n = l\,J + j$ and the corresponding difference vector dimensionality $N^{[c]} = N^{[a]} = L\,J$. For notational convenience, comparison functions that handle Gabor wavelet responses individually are indexed with a lower case letter in the superscript, while Gabor jet comparison functions have capital



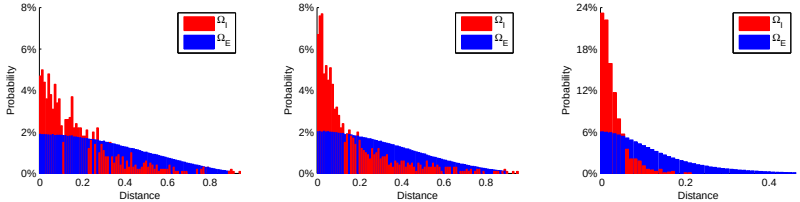
(a) $S_{[A]}$ **(b)** $S_{[C]}$ **(c)** $S_{[M]}$

(d) $S_{[P]}$ **(e)** $S_{[D]}$

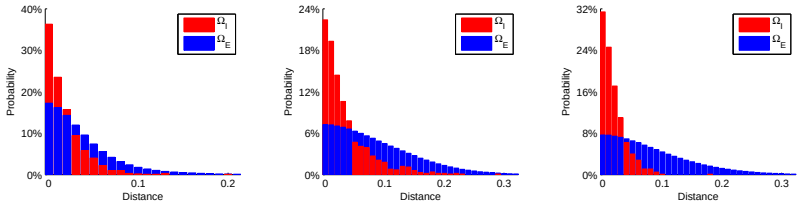
Figure 4.3: Similarity Value Histogram: *This figure displays histogram plots of 1000 intrapersonal and 999,000 extrapersonal* $S_{[A]}$, $S_{[C]}$, $S_{[M]}$, $S_{[P]}$, *or* $S_{[D]}$ *similarity value distributions computed between Gabor jets taken from the nose tip landmark of the FaceGen database (cf. Section 5.1.1). The histogram bin width is* 0.01 *in all cases.*

letters. In opposition to S^{IEC}, the $S^{\mathcal{G}}$ graph comparison function needs to add similarity values $S(\cdot,\cdot)$ instead of distance functions $D(\cdot,\cdot)$. For that purpose, the similarity values are computed as $S(\cdot,\cdot) = 1 - D(\cdot,\cdot)$ (or simply $S(\cdot,\cdot) = -D(\cdot,\cdot)$ in case $D(\cdot,\cdot)$ does not have a maximal value of 1). This implies that the two functions $S^{\mathcal{G}}_{[C]}$ and $S^{\mathcal{G}}_{[c]}$ are identical.

Figure 4.4 displays sample distributions of $D_{[c]}$ and $D_{[a]}$ distance values that were generated employing the same dataset as for Figure 4.3, this time using Gabor jets taken at the nose root landmark. From these Gabor jets, the responses of vertically oriented Gabor wavelets ($\nu = 0$) in three different scale levels ($\zeta \in \{0, 2, 4\}$) are compared. Although the orientation of the Gabor wavelets are identical, the distribution of the distance values of the three levels are quite different. While for the high frequency Gabor wavelets (cf. Figures 4.4(a) and 4.4(d)), there are many high intrapersonal distances, the intrapersonal distances of low frequency Gabor wavelets (see



(a) $D_{[c]}, \vec{k} = \left(\frac{\pi}{2}, 0\right)^{\mathrm{T}}$ **(b)** $D_{[c]}, \vec{k} = \left(\frac{\pi}{4}, 0\right)^{\mathrm{T}}$ **(c)** $D_{[c]}, \vec{k} = \left(\frac{\pi}{8}, 0\right)^{\mathrm{T}}$



(d) $D_{[a]}, \vec{k} = \left(\frac{\pi}{2}, 0\right)^{\mathrm{T}}$ **(e)** $D_{[a]}, \vec{k} = \left(\frac{\pi}{4}, 0\right)^{\mathrm{T}}$ **(f)** $D_{[a]}, \vec{k} = \left(\frac{\pi}{8}, 0\right)^{\mathrm{T}}$

Figure 4.4: Distance Value Histogram: *This figure displays histogram plots of 1000 intrapersonal and 999,000 extrapersonal $D_{[c]}$ and $D_{[a]}$ distance value distributions computed between Gabor jets taken from the nose root landmark of the FaceGen database (cf. Section 5.1.1), using responses of vertically oriented Gabor wavelets of three different scale levels. The histogram bin width is* 0.01 *in all cases.*

Figures 4.4(c) and 4.4(f)) are comparably small. Nevertheless, the overlap of both classes is on average much higher than when taking Gabor jet similarities, but this might be compensated by the higher number of dimensions of the $\vec{u}^{[c]}$ or $\vec{u}^{[a]}$ vectors. Note that these distributions are idealized since the underlying faces are computer-generated.

In his MSc thesis [31], Haufe tested some other functions comparing Gabor jets that exploit absolute values and phases of the Gabor wavelet responses. He showed that Gabor phases are well suited for face recognition when node positioning errors that affects the phase difference are diminished by disparity estimation (cf. Appendix B). Thus:

$$S_{[d]}\left(\phi_j, \phi'_j\right) = \cos\left(\phi_j - \phi'_j - \vec{k}_j^{\mathrm{T}}\, \vec{d}\right) \qquad (4.3-13)$$

works in general better than the simple:

$$S_{[p]}\big(\phi_j, \phi'_j\big) = \cos\big(\phi_j - \phi'_j\big) \ . \tag{4.3–14}$$

Haufe [31] also demonstrated that the combination of absolute and phase values outperforms solo absolutes or phases. This combination can easily be achieved by stringing together $\vec{u}^{[c]}$ and $\vec{u}^{[d]}$:

$$u_{2n}^{[c,d]} = S_{[c]}\big(a_{l;j}, a'_{l;j}\big) \ , \qquad u_{2n+1}^{[c,d]} = S_{[d]}\big(\phi_{l;j}, \phi'_{l;j}\big) \ , \tag{4.3–15}$$

the resulting dimensionality of the comparison vector is: $N^{[c,d]} = 2LJ$.

The second group of functions deals with the geometry of the graphs, i. e., with the nodes or edges. Two functions that compare the geometry of two graphs by averaging the edge differences over all edges were already presented in Section 4.2.2. Again, the sum of Equation (4.2–1) is split up and its horizontal and vertical components are used independently:

$$\begin{aligned} u_{2e}^{[\mathcal{E}_{h/v}]} &= D_{[\mathcal{E}_h]}\Big(\vec{\Delta}_e, \vec{\Delta}'_e\Big) = \frac{\big(\Delta_{e;h} - \Delta'_{e;h}\big)^2}{\Big(\vec{\Delta}_e\Big)^2 + \Big(\vec{\Delta}'_e\Big)^2} \ , \\ u_{2e+1}^{[\mathcal{E}_{h/v}]} &= D_{[\mathcal{E}_v]}\Big(\vec{\Delta}_e, \vec{\Delta}'_e\Big) = \frac{\big(\Delta_{e;v} - \Delta'_{e;v}\big)^2}{\Big(\vec{\Delta}_e\Big)^2 + \Big(\vec{\Delta}'_e\Big)^2} \ , \end{aligned} \tag{4.3–16}$$

with $N^{[\mathcal{E}_{h/v}]} = 2E$ being twice the number of edges of the graphs. Accordingly, Equation (4.2–2) can be used in the IEC environment by computing:

$$u_e^{[\mathcal{E}]} = D_{[\mathcal{E}]}\Big(\vec{\Delta}_e, \vec{\Delta}'_e\Big) = \frac{\left|\Big(\vec{\Delta}_e\Big)^2 - \Big(\vec{\Delta}'_e\Big)^2\right|}{\Big(\vec{\Delta}_e\Big)^2 + \Big(\vec{\Delta}'_e\Big)^2} \ , \tag{4.3–17}$$

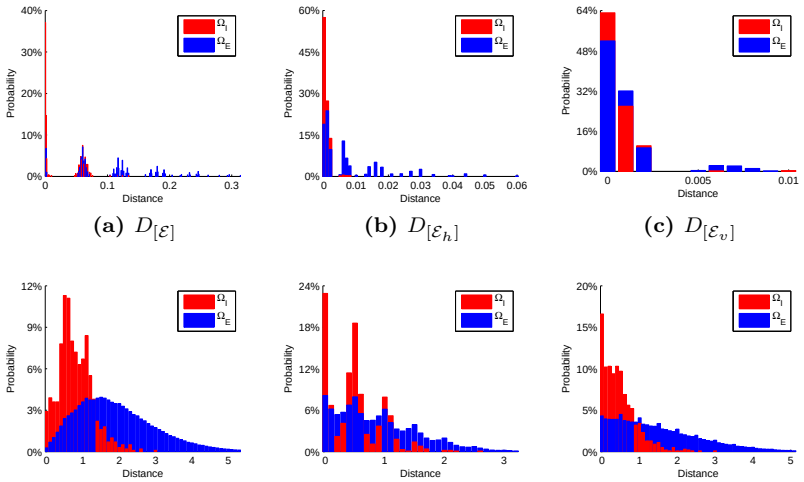
with the difference vector dimensionality $N^{[\mathcal{E}]} = E$.

The landmark positions of the graphs can also be used directly. The functions:

$$\begin{aligned} u_{2l}^{[\mathcal{L}_{h/v}]} &= D_{[\mathcal{L}_h]}\big(\mathcal{L}_l, \mathcal{L}'_l\big) = \big|(\mathcal{L}_{l;h} - \mathcal{C}_h) - \big(\mathcal{L}'_{l;h} - \mathcal{C}'_h\big)\big| \ , \\ u_{2l+1}^{[\mathcal{L}_{h/v}]} &= D_{[\mathcal{L}_v]}\big(\mathcal{L}_l, \mathcal{L}'_l\big) = \big|(\mathcal{L}_{l;v} - \mathcal{C}_v) - \big(\mathcal{L}'_{l;v} - \mathcal{C}'_v\big)\big| \ , \end{aligned} \tag{4.3–18}$$

and:

$$u_l^{[\mathcal{L}]} = D_{[\mathcal{L}]}\big(\mathcal{L}_l, \mathcal{L}'_l\big) = \big\|(\mathcal{L}_l - \mathcal{C}) - \big(\mathcal{L}'_l - \mathcal{C}'\big)\big\| \tag{4.3–19}$$



(a) $D_{[\mathcal{E}]}$ **(b)** $D_{[\mathcal{E}_h]}$ **(c)** $D_{[\mathcal{E}_v]}$

(d) $D_{[\mathcal{L}]}$ **(e)** $D_{[\mathcal{L}_h]}$ **(f)** $D_{[\mathcal{L}_v]}$

Figure 4.5: Geometry Distance Value Histogram: *This figure displays histogram plots of 1000 intrapersonal and 999,000 extrapersonal* $D_{[\mathcal{E}]}$ *and* $D_{[\mathcal{L}]}$ *distance value distributions and their variants that treat horizontal and vertical components individually. The edge comparison functions are applied on the edge between left and right alar wing of the nose, while the landmark comparison functions compared the chin landmarks, both exploiting ground truth node positions of the FaceGen database (cf. Section 5.1.1). The histogram bin width is* 0.001 *for* $D_{[\mathcal{E}]}$ *and* 0.1 *for* $D_{[\mathcal{L}]}$.

use the distance of the landmark positions $\mathcal{L}_l$ relative to the according center of gravity $\mathcal{C}$ of the graph. As before, the two variants of the landmark comparison functions compare the landmark positions using Euclidean distance (see Equation (4.3 – 19)) or horizontal and vertical component independently (cf. Equation (4.3 – 18)) with $N^{[\mathcal{L}]} = L$ and $N^{[\mathcal{L}_{h/v}]} = 2L$, respectively.

Figure 4.5 displays sample distributions of the geometry comparison functions $D_{[\mathcal{E}]}$ and $D_{[\mathcal{L}]}$ and their according $\mathcal{E}_{h/v}$ and $\mathcal{L}_{h/v}$ variants. The edge distances were computed using the edge from left to right alar wing of the nose. This edge seems to be very descriptive for the identity since the intrapersonal distance is usually much smaller than the extrapersonal one (see Figure 4.5(a)), especially when only the horizontal part is regarded, as

shown in Figure 4.5(b). In opposition, the vertical part from Figure 4.5(c) does not show a clear differentiation between intrapersonal and extrapersonal changes. Similarly, the vertical distribution of chin landmark distances (cf. Figure 4.5(f)) is different between intra- and extrapersonal variations, while the horizontal difference shown in Figure 4.5(e) does not show this difference equally palpable.

The combination of comparison functions of different feature types is very easy. Although the ranges of the original similarity and distance values might differ, the S^{IEC} similarity scores are dimensionless and, thus, can easily be added up. To combine, e. g., the Canberra jet similarity $S_{[C]}$ with the landmark comparison $D_{[\mathcal{L}]}$, the corresponding integrated S^{IEC} similarity score is calculated as:

$$S^{\mathrm{IEC}}_{[C,\mathcal{L}]}\left(\mathcal{G},\mathcal{G}'\right) = S^{\mathrm{IEC}}_{[C,\mathcal{L}]}\left(\vec{u}^{\,[C,\mathcal{L}]}\right) = S^{\mathrm{IEC}}_{[C]}\left(\vec{u}^{\,[C]}\right) + S^{\mathrm{IEC}}_{[\mathcal{L}]}\left(\vec{u}^{\,[\mathcal{L}]}\right) . \qquad (4.3-20)$$

4.4 Classification

In the previous section, the IEC approach is employed to classify whether two images show the same person. The same approach can be used to classify other properties of the face by exchanging the training graph pairs in T_I and T_E accordingly. One apparent property of a facial image is the gender of the shown person, but also properties such as the facial expression or the illumination condition, under which the image was taken, can be classified.

4.4.1 Classification with IEC

More abstractly, the probe graph $\mathcal{G}^{(p)}$ is classified into one of the categories $c = 0, \ldots, C\text{-}1$. In this thesis, two possible ways to achieve classification are used. The first way compares probe graph $\mathcal{G}^{(p)}$ to all gallery graphs $\mathcal{G}^{(g)}$ and assign the category of the most similar gallery item:

$$c = \mathrm{Cat}\left(\underset{\mathcal{G}^{(g)}}{\arg\max}\ S\left(\mathcal{G}^{(g)}, \mathcal{G}^{(p)}\right) \right) . \qquad (4.4-1)$$

For computing the similarity S, the $S^{\mathcal{G}}$ and the S^{IEC} similarity measures can be used. In the latter case, the IEC system is not trained to classify intrapersonal and extrapersonal graph differences, but *intra-category* and *extra-category* differences. Hence, the intra-category training vectors $\vec{u}_I$ are computed as the similarity between two graphs of the same category, while $\vec{u}_E$ comprise the similarities of two graphs of different categories. How two graphs are converted into a similarity vector $\vec{u}$ is detailed in Section 4.3.3.

A more direct classification computes similarities to approximated category centers:

$$\mathcal{G}_c = \frac{1}{|T_c|} \sum_{\mathcal{G} \in T_c} \mathcal{G} \qquad (4.4-2)$$

with $T_c = \{\mathcal{G}^{(b)} \mid \mathrm{Cat}\big(\mathcal{G}^{(b)}\big) = c\}$ being the set of training graphs of category c. The mean graphs $\mathcal{G}_c$ are calculated by averaging the node positions and the Gabor jets, where the latter averages the complex Gabor wavelet responses in the algebraic form. Instead of performing the IEC training on similarities of two graphs, the similarities to the category centers are learned. Thus, the intra-category training set T_I contains graph differences $\vec{u}_I$ that are computed as the similarity of graphs $\mathcal{G}^{(b)}$ to centers $\mathcal{G}_c$ of the correct category $c = \mathrm{Cat}\big(\mathcal{G}^{(b)}\big)$, while T_E contains the vectors $\vec{u}_E$ of $\mathcal{G}^{(b)}$ to all other categories $c' \neq c$:

$$\vec{u}_I = S\big(\mathcal{G}_c, \mathcal{G}^{(b)}\big) \ , \qquad \vec{u}_E = S\big(\mathcal{G}_{c'}, \mathcal{G}^{(b)}\big) \ . \qquad (4.4-3)$$

The classification stage assigns the category c for probe graph $\mathcal{G}^{(p)}$ according to the highest category similarity:

$$c = \arg\max_{c} S\big(\mathcal{G}_c, \mathcal{G}^{(p)}\big) \ , \qquad (4.4-4)$$

where, again, S might be either $S^{\mathcal{G}}$ or S^{IEC}. To differentiate between the two classification approaches, the functions comparing graphs to category centers are denoted with $S^{\mathcal{G}+c}$ and $S^{\mathrm{IEC}+c}$, respectively. The time complexity varies between S^{IEC} and $S^{\mathrm{IEC}+c}$ since the number of graph comparisons is different. While $S^{\mathrm{IEC}+c}$ compares the probe graph to the graph centers $\mathcal{G}_c$ of the C categories only and, thus, only C graph comparisons need to be computed, S^{IEC} requires to compare the probe graph with each of the G gallery graphs. Similarly, the IEC training times of the two systems vary.

4.4.2 Leave-One-Out Cross-Validation

Sometimes, classification databases suffer from having only a small number of images. Most often, it is not possible to reasonably split up these databases into training and test subsets since the results of different subset-splits might be incomparable.

When the classifier training is fast, it is also feasible to perform a *leave-one-out cross-validation* (LOOCV). In this procedure, successively each of the examples is tested against the rest of the database. To achieve this, the classifier is trained on the complete database except for the currently

left-out probe element. When the classifier needs a gallery, e. g., in case of Equation (4.4 – 1), the training set also serves as such, and the category centers $\mathcal{G}_c$ (cf. Equation (4.4 – 2)) are computed from the training set as well. The probe item is classified with this trained classifier, and the procedure is repeated for the following database item.

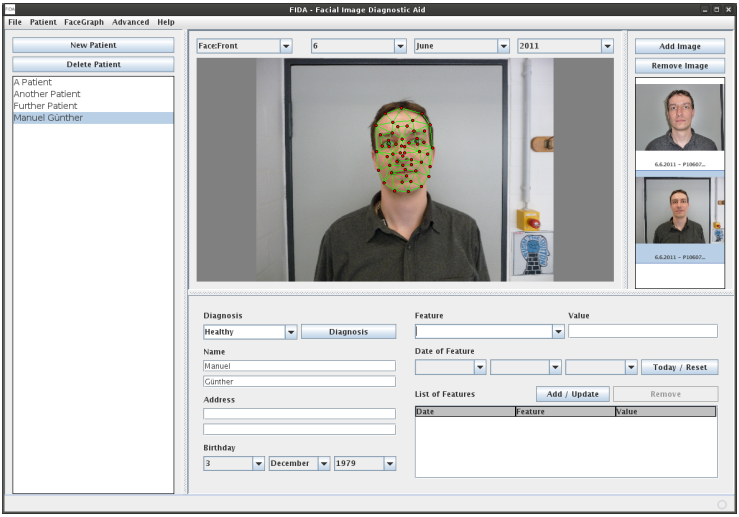
With the classification results of the single elements, a CMC curve as introduced in Section 4.1.1 can be generated. The rank 1 match characteristics is now called the *classification rate* (CR). Note that in general the CMC curves differ between S^{IEC} and $S^{\mathrm{IEC}+c}$ since the gallery size is identical to the training set size for the former and number of categories for the latter. In opposition, classification rates can be compared directly between the two approaches.

4.4.3 Facial Image Diagnostic Aid

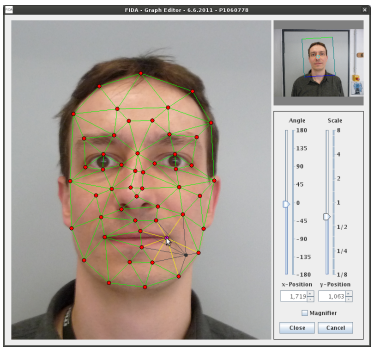
The IEC classification is not limited to direct properties of the face like gender or facial expression of the person shown in the image, but any classifiable property might be classified. A challenging facial property is introduced by genetic defects that have an impact on the appearance and the placement of some facial features. How well these genetic syndromes can be classified automatically is tested in Section 5.4.3.

Usually, patients that potentially suffer from these kinds of genetic defects are examined by medical doctors. To help these doctors with diagnosing a patient, a preselection of possible syndromes is automatically returned by the *facial image diagnostic aid* (FIDA) *graphical user interface* (GUI) that I implemented. FIDA is designed to manage a database of patients and their corresponding syndromes, where for each patient several images in several views, e. g., one frontal facial image and one full profile view of the face, and from several dates might be stored. FIDA incorporates the GUI written in Java, a screen-shot of which is shown in Figure 4.6(a), with the IEC classifier written in C++.

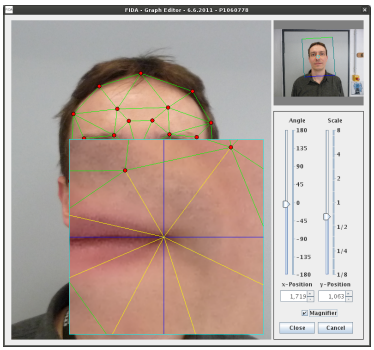
Before diagnoses can be preselected, the face graph must be placed on the image. To do this, $\mathcal{G}^{\mathrm{ML}}$ detectors as presented in Section 3.3 are trained for each different view type using hand-labeled face graphs. If for any reason the multi-scale and multi-angle face detection fails, FIDA allows for a manual rough geometrical normalization of the image in order to increase detectability. After automatic face detection and landmark localization, landmarks can be fine-tuned manually using the integrated *face graph editor* GUI. Two screen-shots of the face graph editor GUI during operation are shown in Figures 4.6(c) and 4.6(b), with and one without the magnifier tool being enabled, respectively.



(a) Facial Image Diagnostic Aid



(b) Face Graph Editor



(c) Face Graph Editor with Magnifier

Figure 4.6: Facial Image Diagnostic Aid: *This figure shows screen-shots of (a) FIDA and the face graph editor GUI during work: (b) with disabled and (c) with enabled magnifier tool.*

After hand-labeling the graphs for the faces of all view types, the Diagnosis button can be pressed. This triggers an image and graph standardization using specific node positions of the graph (i. e., the eye positions in case of frontal faces), a Gabor wavelet transform and the extraction of the Gabor jets at the each node position, and a classification of the graph using $S^{\mathrm{IEC}+c}$. Finally, the five most probable syndromes are displayed and the user can select one of these syndromes as the finally estimated diagnosis for this patient.

This procedure requires four different trained $S^{\mathrm{IEC}+c}$ classifiers, one for each combination of texture and geometry data types with frontal and profile view types. These classifiers are trained off-line on a database of patients with genetic defects that are approved by direct gene examination. To allow tests on own image databases, i. e., including on-line classifier training on this database, a leave-one-out cross-validation as described above is implemented in FIDA. This function collects all graphs of all view types, performs image standardizations and Gabor wavelet transformations, computes the classification rates for each view and data type pair, and writes all similarity values to a file. The combination of different view and data types can then be computed by the administrator of the database, e. g., using Equation (4.3 – 20). Likely, the integration of different view types is straightforward, i. e., by totalizing $S^{\mathrm{IEC}+c}$ similarity values.

Not only genetic syndromes can be classified by FIDA. In [79], we also showed that the $S^{\mathrm{IEC}+c}$ classifier is able to classify patients with acromegaly against a control group. The best results were achieved by a combination of the $S_{[P]}$ texture comparison function with the $S_{[\mathcal{E}]}$ geometry comparison, both exploiting the statistics of hand-labeled face graphs in frontal and right profile view. The overall leave-one-out classification rate of 81.9% outperformed the 64.9% CR of medical internists that were not particularly trained with acromegaly, and even beat medical experts especially trained for genetic syndrome classification, which achieved 72.1% CR on classifying the probands based on the same facial images. Notably, patients with mild features of acromegaly were classified correctly more often by FIDA than by the experts, and subjects of the control group were correctly classified in over 90% by FIDA, while experts reached only 80% CR.

Chapter 5

Experiments

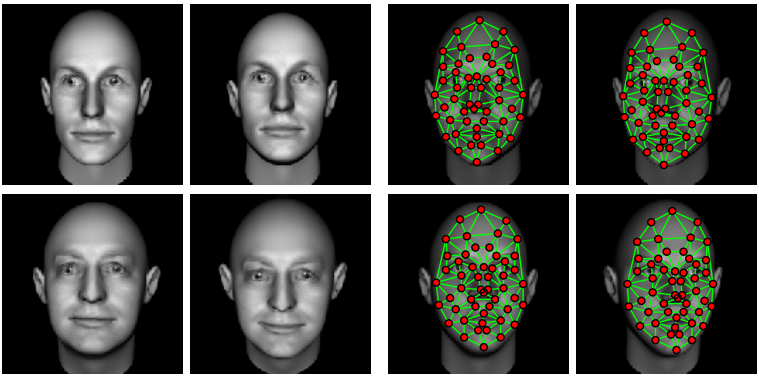
In this chapter, the proposed face detection, landmark localization, face recognition, and facial property classification algorithms are tested on some default image databases, which are introduced in Section 5.1. The face detection experiments described in Section 5.2 illustrate the quality of the proposed face detection and landmark localization algorithms. The resulting face graphs serve as the basis for the following face recognition and classification experiments that are given in Sections 5.3 and 5.4, respectively.

5.1 Face Databases

5.1.1 FaceGen

The first database that is used in the detection and recognition experiments is the FaceGen database that Müller [55] introduced. The FaceGen database consists of artificially generated facial images including *ground truth* (GT) facial landmarks, i. e., all face graphs are placed according to exact facial landmark positions. Still, some landmarks that correspond to bone structures like the cheekbone have little texture and are probably hard to detect precisely. This database is especially designed for testing face detection and landmark localization algorithms since node positioning errors can be computed. Nonetheless, a face detection algorithm performing well on these artificial data does not necessarily operate on natural data, too. Big disadvantages of these images are that they do not include facial hair and the structure of the skin is very smooth.

In the FaceGen database, 1000 identities with each two gray-scale images of size 128×128 pixel are included. The difference between two facial images of one identity is a medium pose change, which occurred in both images independently. Some exemplary image pairs and their accordant GT face graphs are shown in Figure 5.1.



(a) FaceGen Images **(b)** Ground Truth Graphs

Figure 5.1: FaceGen Images and Ground Truth Face Graphs: *This figure displays (a) some FaceGen identities (b) including their ground truth face graphs, with both images of each identity grouped horizontally.*

5.1.2 CAS-PEAL

The CAS-PEAL database [23] is a rather big database of 1040 Chinese people. The images of this database originally are of size 360×480 pixel, but I use versions downscaled to 192×256 pixel. CAS-PEAL provides different kinds of variations in the facial images. The complete list is:

1. Pose variation including 27 different poses from -90° left to 90° right profile in conjunction with three nod angles
2. Expression variations: Neutral (**N**), Laugh (**L**), Frown (**F**), Surprise (**S**), Closed eyes (**C**), and Opened mouth (**O**)
3. Lighting variations: frontal ambient lighting, fluorescent lighting from 15 different directions, and incandescent lighting from up to five locations
4. Accessories: three different types of hats and glasses (no sunglasses)
5. Distance: the size of the face in the image varies around three different scales
6. Aging: two images were taken at intervals of six month

7. Background: the color of the background changes auto white-balancing of the camera

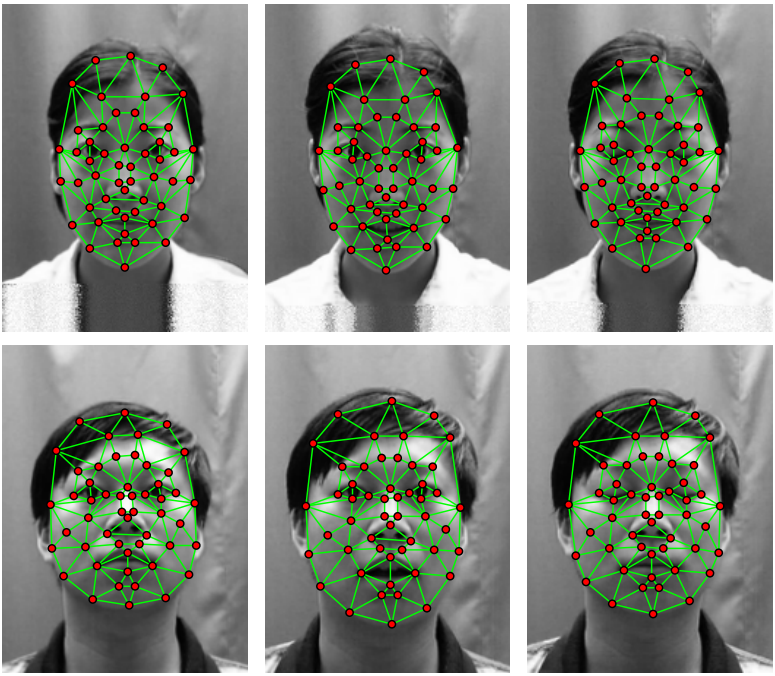
For these subsets, the CAS-PEAL database provides default experimental setups, always comparing the distorted faces to the neutral faces, and the recognition rates for some standard algorithms like PCA (cf. Section 4.2.3), PCA+LDA (Section 4.2.4), GPCA+LDA (Section 4.2.5), and LGBPHS as given in Section 4.2.7.

Since the face recognition system proposed in this thesis cannot handle non-frontal images, pose variations are skipped, and to keep this chapter a little shorter, also accessory, background or aging variations are excluded from the tests. The four treated subsets of the database are:

Neutral The neutral faces of the CAS-PEAL database serve as gallery images in all recognition experiments. Hence, for each image in one of the following experiments there is also one neutral facial image with neutral ambient illumination conditions and taken with default camera distance to the subject. Neutral facial images of 18 different people were hand-labeled and serve for training the face detectors, i. e., for building up the $\mathcal{G}^{\mathcal{B}}$ or $\mathcal{G}^{\mathrm{ML}}$ detector graphs. Two examples of standardized neutral face images including the hand-labeled graphs are given in Figure 5.2(a).

Distance The images in the Distance subset enclose faces with neutral facial expressions, but the sizes of the faces in the images vary. The pictures were taken with two different distances of the camera to the subjects, both of which are different to the distance in the Neutral images. In most cases, only one probe image per identity was taken. The number of images in the Distance subset is 307, while the number of subjects is 281. Since the face detector training standardizes the training graphs according to hand-labeled eye positions, anyway, there is no need to have hand-labeled graphs of different image sizes in the training set. Hence, the same 18 hand-labeled graphs as used in the Neutral subset are taken in the face detection experiments.

Expression For the expression subset, there are 18 hand-labeled graphs available, showing three men and three women in expressions Neutral, Surprised, and Closed eyes. One man and one woman including the hand-labeled graphs after standardization (cf. Section 2.4.3) are given in Figure 5.2. For the face detection and identity recognition experiments, 1884 images of 377 people showing expressions Laugh, Frown, Surprise, Closed eyes, and Opened mouth are used. The expression classification experiments in Section 5.4.1 additionally use the accordant Neutral graphs.



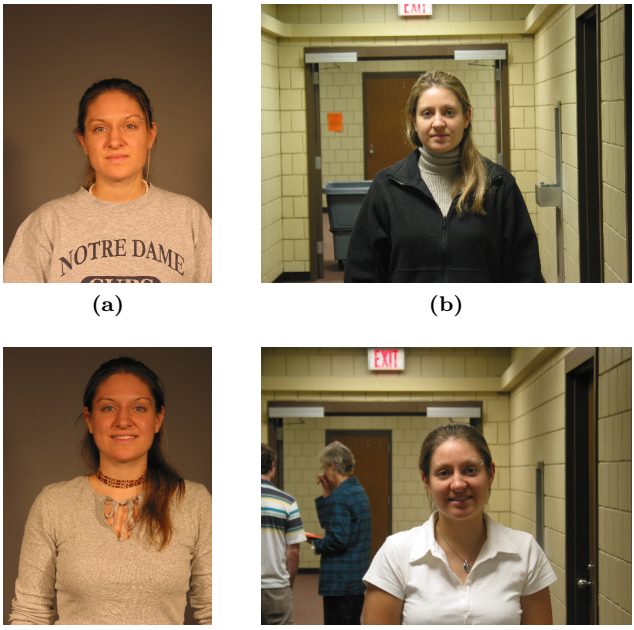
(a) Neutral **(b)** Surprise **(c)** Closed Eyes

Figure 5.2: Hand-Labeled Face Graphs of the CAS-PEAL Expression Subset: *This figure shows standardized hand-labeled face graphs of two identities from the Expression subset of the CAS-PEAL database showing three different facial expressions.*

Lighting The lighting subset includes 15 different fluorescent lighting conditions. The fluorescent light is emitted from three different vertical angles: -45° (**D**), 0° (**M**) and 45° (**U**) combined with five horizontal positions ranging from -90° to 90° in steps of 45°, see Gao *et al.* [23] for more details. Due to the low number of examples for the incandescent illumination, these types are not included in the lighting probe set. Again, a list of hand-labeled graphs are available for the lighting subset: 15 images for each of four men and four women totalize to 120 face graphs. 15 hand-labeled graphs including the standardized images of one of the women are given in Figure 5.3. The

Figure 5.3: Hand-Labeled Face Graphs of the CAS-PEAL Lighting Subset: *This figure shows all standardized hand-labeled face graphs of one identity from the Lighting subset of the CAS-PEAL database with fluorescent light from 15 directions:*
— Left to right: light from 90°, 45°, 0°, -45°, and -90° horizontally.
*— Top to bottom: light from 45° (**U**), 0° (**M**), and -45° (**D**) vertically.*

number of test images for the different lighting conditions vary between the horizontal light directions. While there are images of 233 identities including frontal light (0°), the count is 199 for light from left (-90° and -45°) and 62 for light from right (45° and 90°). For each identity with an image in left or right lighting condition, there is also one image in frontal condition. Overall, the Lighting subset includes 233 people in 2265 images.



(a) (b)

(c) (d)

Figure 5.4: Exemplary FRGC Images: *This figure displays four exemplary images under controlled and uncontrolled illumination conditions of one person from the FRGC database.*

5.1.3 FRGC

Another widely used database is the *face recognition grand challenge* (FRGC) database [65] in the version FRGC ver2.0 from 2006. It encloses 36,818 colored facial images of 466 identities taken under controlled or uncontrolled lighting conditions, and hand-labeled[1] eye, nose tip, and mouth center positions for each of them. The image resolutions are 1200×1600 pixel or 1704×2272 pixel for the controlled studio portrait images, and 1600×1200 pixel or 2272×1704 pixel for the uncontrolledly illuminated images, most

[1] The FRGC database includes some images with misplaced eye positions, which I corrected. Due to the large size of the FRGC database, the verification experiments should not be influenced much by these corrections.

of which were photographed in a hallway. Images were taken in several sessions distributed over four years, where images of each identity are taken on at least two different days. Four typical images of one person are displayed in Figure 5.4, showing each two controlled and uncontrolled images, and different facial expressions. In opposition to the CAS-PEAL database, the expression shown in an image is not labeled. In the FRGC database, also 3D-data of the faces are contained, but these data are not used here.

The database is split up into a *training* subset of 12,776 controlled and uncontrolled images of 222 identities, as well as a *target* subset containing 16,028 controlled images and a *query* subset containing 8014 uncontrolled images of all 466 people. Consequently, this means that the identities from the training subset are also present in the target and query subsets, contrary to common face database usage practice. The FRGC database provides exact experimental setup for overall six face recognition experiments, including each three different masks defining pairs of gallery and probe images that should be used for computing recognition results in term of ROC curves. These masks have varying difficulties, mask 'I' includes images pairs taken in a time span of up to six month, while mask 'III' uses image pairs with a higher time span and mask 'II' wraps up mask 'I' and mask 'III'. Furthermore, the database [65] provides an eigenface-based baseline algorithm and accordant ROC results. Since these results are based on the most difficult mask 'III', this mask is used in the face recognition experiments in Section 5.3.3 as well.

5.1.4 Human Genetics

The Human Genetics institute in Essen tries to classify human genome defects that have an impact on the facial representation from static facial images [45, 8, 91, 7]. They hand-labeled face graphs by defining landmarks that hopefully are well suited for syndrome classification, a few of these graphs were shown by Loos *et al.* [45], Böhringer *et al.* [8] and Vollmar *et al.* [91]. Since many of these syndromes affect also the profile of the face, additionally to the frontal graphs they also hand-labeled facial images taken from a full left profile of the face. For most of the patients, there are one frontal and one profile image. Overall, the database include patients with 14 different syndromes, ranging from 6 to 30 patients per syndrome. Still, the (healthy) control group is missing, so the classification task is to assign one of the 14 syndromes to the probe patient.

Loos *et al.* [45] and Böhringer *et al.* [8] first conducted experiments on the frontal faces only. They used several methods like jet-voting or PCA+LDA on Gabor wavelet responses. Since the database is rather small – at that time there were 147 frontal images from 10 syndromes – and their training

Abbreviation	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	Mucopolysacchridosis II
MPS3	8	8	Mucopolysacchridosis 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
	218	215	Overall count

Table 5.1: Number of Images per Syndrome: *This table shows the subdivision of the Human Genetic dataset into the accordant syndromes and the number of images in frontal and left profile view.*

procedure took too much time to perform a leave-one-out cross-validation, they performed a 10-fold cross-validation, using on average 130 images for training and the rest for testing the classifiers. Unfortunately, in an attempt to save computation time, they computed the principal component analysis using all available data, and performed the 10-fold cross-validation on the transformed data. Hence, information about the probe data were already included in the training sets. The best result they could achieve is 75.7% correct classification rate on the hand-labeled face graphs and 52.1% on automatically placed landmarks.

Vollmar *et al.* [91] repeated the experiments with increased number of 196 patients with 14 syndromes, again with partially training on the probe data. The classification rate for the hand-labeled graphs of the frontal images dropped slightly to 70.1%. When they included also profile view images of the same patients, the classification rate again increased to 76.1%. Performing a forward selection of the useful principal components, the classification rate further enhanced to 76.9%. The innovation of this paper was the insight that after image standardization the geometrical information of the face graphs is

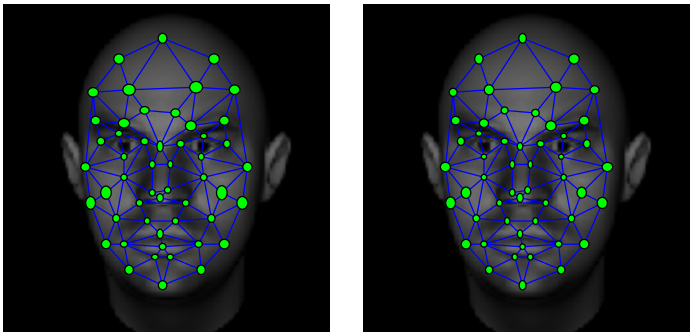
even better suited for syndrome classification than the texture. Using geometry information alone, Vollmar *et al.* [91] reached 85.7% classification rate, and the combination of texture and geometry ended in 93.1% correct classification. Still, they report that automatically detected graphs perform very poorly, ending up in 52.5% classification rate for the texture of the frontal images. After Böhringer *et al.* [7] discovered the training set inconsistency, the previously very good classification rates of 93.1% dropped remarkably, i. e., when the PCA was computed during the 10-fold cross-validation on each fold independently. Hence, Böhringer decided to perform multinomial logistic regression for classification, resulting in 60.4% classification rate on the combination of frontal and profile images including texture and geometrical information. One important notice is that the previously outstanding classification results based on geometrical information of the graphs decreased dramatically from 85.7% to only 35.1%.

Unfortunately, I do not have access to the exact dataset that was used in the experiments by Böhringer *et al.* [7]. The dataset that was given to me includes 219 patients of 14 syndromes embracing 218 frontal and 215 profile views with hand-labeled landmark positions, the complete subdivision can be looked up in Table 5.1. The images in the dataset are scaled to size 256×256 pixel. The frontal faces are aligned to the eye positions and also the profile views are geometrically normalized to upright faces.

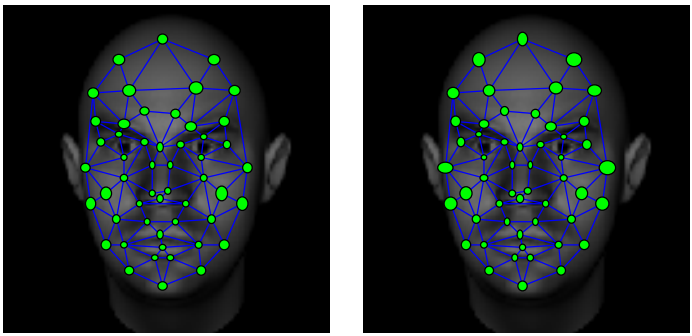
5.2 Face Detection Experiments

5.2.1 Node Positioning Errors

The first face detection experiments concern node positioning errors that occur during face detection and landmark localization. To show that the maximum likelihood face detection and landmark localization is able to detect faces and place landmarks correctly, the ML and the EBGM face detection algorithms were applied to the FaceGen database. Therefore, 100 FaceGen identities with overall 200 ground truth (GT) graphs were used to train the face detectors, i. e., to build up the $\mathcal{G}^{\mathcal{B}}$ and the $\mathcal{G}^{\mathrm{ML}}$ graphs. For the 1800 images from the remaining 900 identities, both algorithms were employed to detect the faces using the same detection schedule. This schedule used the default global, scale, and local moves that are introduced in Section 3.2, i. e., without any kind of rotation correction or Gabor jet transformation. During local move, $S_{[D]}$ used grid positions with interspaces of four pixel to estimate the disparity, while $S_{[P]}$ computed the similarity at each position, both in a range of eight pixel around the offset position. The complete detection



(a) $S^{\mathcal{B}}_{[P]}$ **(b)** $S^{\mathrm{ML}}_{[P]}$



(c) $S^{\mathcal{B}}_{[D]}$ **(d)** $S^{\mathrm{ML}}_{[D]}$

Figure 5.5: Node Positioning Errors: *This figure displays twice the standard deviations of the node positioning errors in horizontal and vertical direction that occurred during the maximum likelihood face detection, given in (b) and (d), and during common EBGM face detection, cf. (a) and (c), using either the scan local move with* $S_{[P]}$ *or the disparity scan local move employing* $S_{[D]}$.

schedules can be found in Appendix D.

The detected node positions are compared to the GT positions. Figure 5.5 displays the node-wise positioning errors for both algorithms. Each ellipse shows twice the standard deviation of the node position errors in horizontal and vertical direction independently. The landmarks of the in-

ner facial features were located well by all tested setups, with average errors of $S_{[P]} : (0.57, 0.66)$ pixel and $S_{[D]} : (0.64, 0.71)$ pixel the ML algorithm was a little more precise than EBGM with $S_{[P]} : (0.64, 0.69)$ pixel and $S_{[D]} : (0.65, 0.69)$ pixel. In opposition, the landmarks with less structure like the forehead and the cheeks were not placed as accurately, e. g., $S^{\mathrm{ML}}_{[P]}$ has an average error of $(0.85, 0.89)$ pixel. Surprisingly, also the eyebrow landmarks, which I expected to be precisely locatable features, were placed rather poorly by all tested algorithms. The nodes of the rim of the face show larger errors, most probably because the GT nodes are not located at the margin of the face, but are rather aligned to invisible landmarks used for rendering the images.

Face detection works comparably well for both algorithms. Nonetheless there is a huge difference in the time needed to complete it. All experiments were executed on a Dell Precision 670 desktop computer with a 3200 MHz Intel Xeon (32 bit) dual-core processor. While the ML face detection was able to detect all 1800 faces in only 24 minutes ($S_{[P]}$) and 13 minutes ($S_{[D]}$) – less than a second per image – the EBGM detection took 17 hours or 5 hours, respectively – over 10 seconds for one image. The reason for this issue is that the EBGM detection time scales linearly with the number $B = 200$ of training graphs, while the ML detection time is independent of B. Although the detection with $S_{[D]}$ is faster, it does not show better results than employing $S_{[P]}$ in the local moves. Furthermore, tests showed that recognition and classification accuracies drop slightly when using $S_{[D]}$ instead of $S_{[P]}$, also on natural images. Therefore, $S_{[P]}$ is employed in all local moves in the following face detection experiments.

5.2.2 Scale and Rotation Estimation

The next family of detection experiments evaluates, how well in-plane rotation angles and scales of images of natural faces can be estimated by the multi-scale and multi-angle face detection introduced in Section 3.4. To get a rough estimation of the correct scale s^* and angle α^* of the face, the hand-labeled eye positions were used to standardize the image as proposed in Section 2.4.3. Due to imprecisely and inconsistently labeled eye landmarks in the CAS-PEAL database, the estimated scales and angles deviate from the actual ones slightly, especially eye positions for the closed eye images are often hand-labeled poorly.

The quality of the detection algorithm can be estimated by the scaling error and the rotation error:

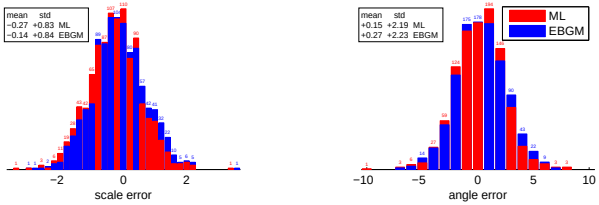
$$\epsilon(s, s^*) = 10 \log_2\left(\frac{s}{s^*}\right), \qquad \epsilon(\alpha, \alpha^*) = \alpha - \alpha^*, \tag{5.2–1}$$

with negative scale errors standing for graphs that are too small. The experiments were conducted on the Neutral, Distance, Expression, and Lighting subsets of the CAS-PEAL database. For each of the experiments, ML and EBGM algorithms were trained on the corresponding hand-labeled data for that subset. Afterwards, both algorithms were employed to detect the faces in the test images.

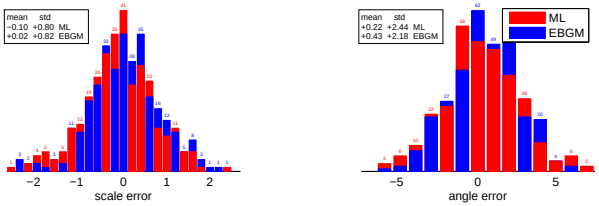
In a first trial, the face positions were determined using the multi-scale and multi-angle face detection as described in Section 3.4 without any further local move. The employed detection schedules can be obtained from Appendix D. Since the algorithms detect the face, and not the eyes directly, scale errors between about -1 and 1 and rotation errors up to 5° are acceptable. Figure 5.6 shows histogram plots of scale and rotation errors in the CAS-PEAL database that occurred during multi-scale and multi-angle face detection, including mean and standard deviation of the errors. In the Neutral and in the Distance subsets, the ML algorithm generated slightly more correct scales than EBGM, while the angle errors are distributed more or less identically. In opposition, most of the scale misdetections[2] in the Expression and Lighting subset were made by the ML algorithm, which also explains the higher standard deviations. Still, many of the scale errors in the Expression subset (cf. Figure 5.6(c)) are due to poorly hand-labeled eye positions in closed eye images, but for the Lighting subset, the assumption of the uni-modal Gaussian distribution of the Gabor wavelet responses is most probably not fulfilled. One conspicuous result for all four subsets is that the ML algorithm had the tendency to underestimate the scale, i. e., the average scale error is negative, while the EBGM algorithm more often overestimated scales. Apparently, the mode of the scale errors in the Lighting subset as given in Figure 5.6(d) is above zero, i. e., the detected face graphs on average are a little to big for the faces.

In a further trial, additional landmark localization steps were added to the detection schedule, and a second image standardization was applied, cf. Section 3.4.3 for details. The detected scales s and angles α now are computed as a combination of both normalization steps. Scale and angle errors are expected to be much smaller since now the eye positions are directly compared. Figure 5.7 displays histogram plots of the errors after face detection and landmark localization that occurred when applying ML and EBGM algorithms to the images. Since the EBGM detection algorithm seems to work slightly better, i. e., it produced fewer misdetections in the Expression and Lighting subset (cf. Figures 5.6(c) and 5.6(d)), the combination of

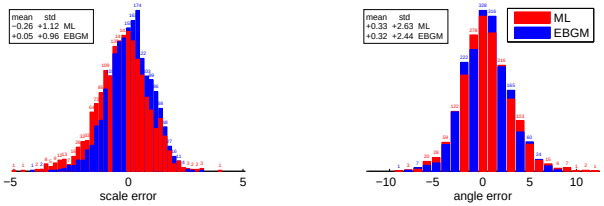
[2]The position of the face was found correctly in all cases, but scale and angle sometimes deviate.



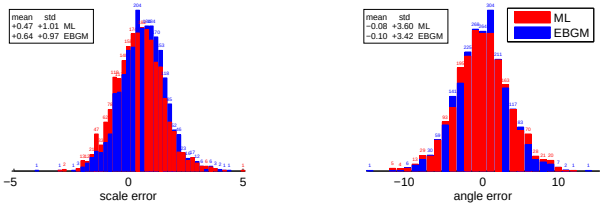
(a) Neutral Subset



(b) Distance Subset

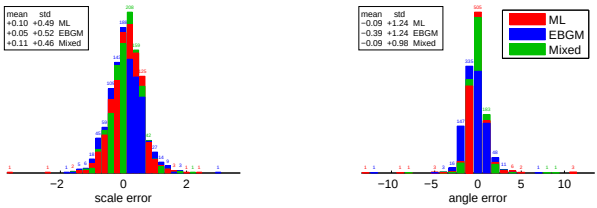


(c) Expression Subset

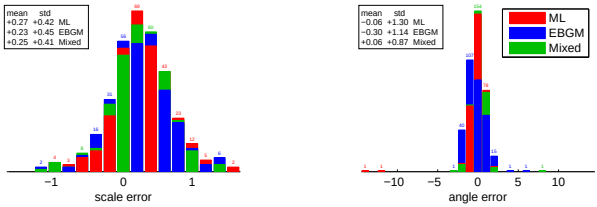


(d) Lighting Subset

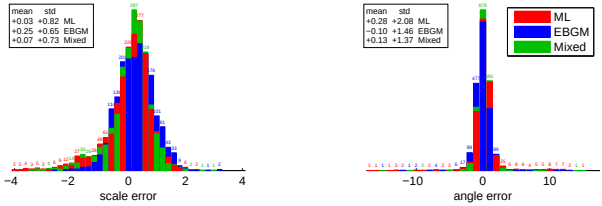
Figure 5.6: Scale and Angle Errors after Face Detection: *This figure displays histograms of $\epsilon(s, s^*)$ scale and $\epsilon(\alpha, \alpha^*)$ angle errors that occurred during face detection in the Neutral, Distance, Expression, and Lighting subsets of the CAS-PEAL database. Each plot includes mean and standard deviation of the error value for ML and EBGM detection.*



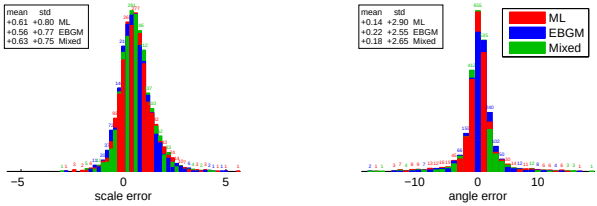
(a) Neutral Subset



(b) Distance Subset



(c) Expression Subset



(d) Lighting Subset

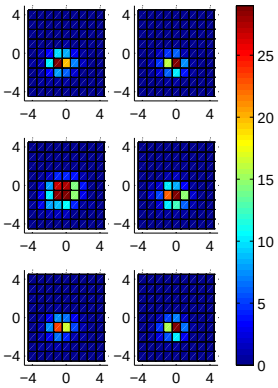
Figure 5.7: Scale and Angle Errors after Landmark Localization: *This figure displays histograms of scale and angle errors that occurred during face detection and successive landmark localization in the Neutral, Distance, Expression, and Lighting subsets of the CAS-PEAL database when employing ML and EBGM face detection and landmark localization, as well as a combination of EBGM face detection and ML landmark localization.*

EBGM face detection and ML landmark localization was executed, too. In comparison to the face detection results given in Figure 5.6, the additional local move increases the average scale and angle detection accuracy greatly. Nonetheless, the highest angle error increased, e. g., from 7° to -14° in the distance subset (cf. Figures 5.6(b) and 5.7(b)). In most of those cases, the eyes were not locatable precisely since at least one of the eyes were covered with hair[3]. In general, the evaluation of the first trial can be repeated: ML is better suited for the Neutral and Distance subsets, while EBGM performed better on the Expression and Lighting subsets. Notably, the Mixed setup could decrease errors in most cases. In summary, the best combination is the multi-scale and multi-angle EBGM face detection combined with a ML landmark localization.

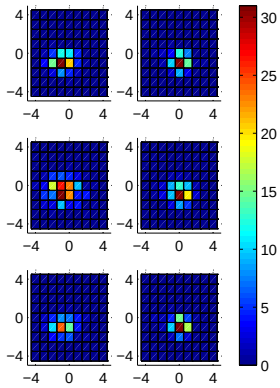
Finally, Figure 5.8 presents distributions of positioning errors of the left and right eye landmarks. Obviously, the landmarks detected with the $\mathcal{G}^{\mathrm{ML}}$ detection setup generates more stable results, while $\mathcal{G}^{\mathcal{B}}$ distributes landmarks more widely. Interestingly, most of the detected eye positions are one pixel below the hand-labeled positions given by the CAS-PEAL database, and the right eye position additionally is one pixel off to the left. This is most probably due to the fact that the average of the four eye nodes used as the center of the eye does not correspond to the true eye center, and maybe some eye landmarks of the hand-labeled graphs are placed incorrectly. This also explains that the scale errors shown in Figure 5.7 are on average greater than zero.

An interesting point is the execution time of face detection and landmark localization experiments. Table 5.2 shows the average time needed to perform the experiments described in this section. Again, the more hand-labeled graphs were used to generate $\mathcal{G}^{\mathcal{B}}$ and $\mathcal{G}^{\mathrm{ML}}$ detector graphs, the higher the gain of the ML detection time in comparison to the EBGM detection time becomes. While the Neutral, Distance, and Expression subsets relied on 18 hand-labeled training graphs, Lighting used 120 graphs. The ML detection needed most of the time for the Gabor wavelet transforms, while EBGM required a lot more in the landmark fine-tuning. The longer detection times needed in the Distance experiments are due to the higher number of tested scales.

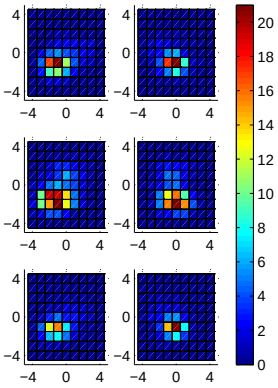
[3]The landmark localization algorithms can handle covered parts of the face only when the hand-labeled training graphs include those cases. Here, no eyes covered by hair have been in the training set.



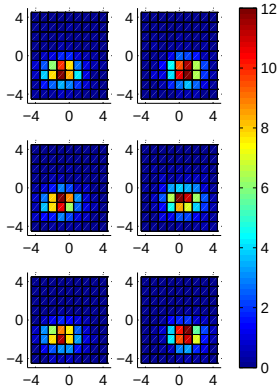
(a) Neutral



(b) Distance



(c) Expression



(d) Lighting

Figure 5.8: Eye Landmark Positioning Errors: *This figure displays histograms of positioning errors of the left and right eye landmark after landmark localization in the Neutral, Distance, Expression, and Lighting subset of the CAS-PEAL database. Each sub-figure includes left (right) and right (left) eye landmark positioning errors after applying the ML (top), EBGM (center) or the Mixed algorithm (bottom).*

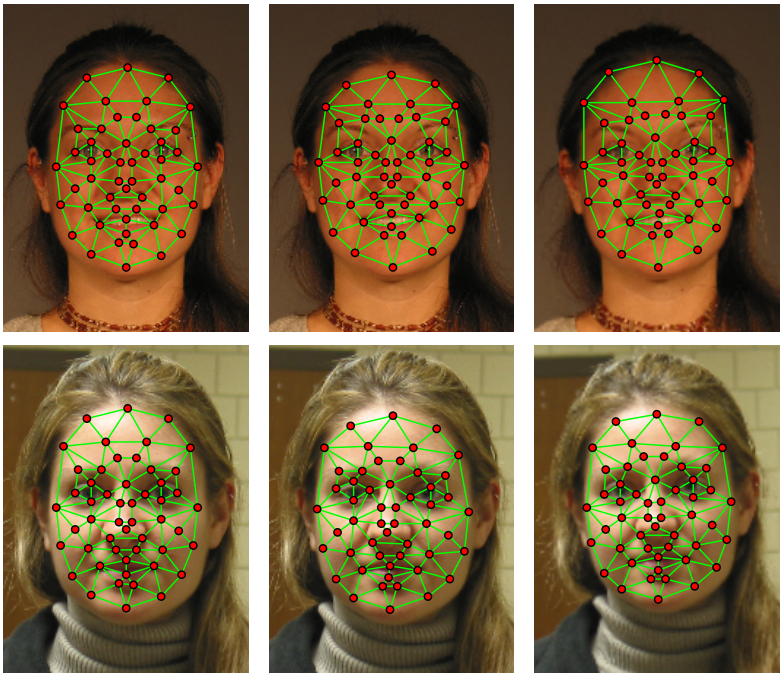
	Neutral	Expression	Distance	Lighting
ML Det.	3.7 sec.	3.7 sec.	6.8 sec.	3.8 sec.
EBGM Det.	5.0 sec.	4.6 sec.	8.8 sec.	18.7 sec.
ML	4.9 sec.	4.7 sec.	7.7 sec.	5.0 sec.
EBGM	8.9 sec.	8.2 sec.	12.5 sec.	37.9 sec.
Mixed	5.7 sec.	5.7 sec.	9.8 sec.	20.3 sec.

Table 5.2: Detection times: *This table shows the average times needed to detect the faces and place the landmarks applying the EBGM and ML algorithms to each image of the four different subsets of the CAS-PEAL database.*

5.2.3 FRGC Face Graph Extraction

The FRGC database is rather new and no hand-labeled graphs are available for this database. To circumvent hand-labeling these images, the hand-labeled face graphs of the CAS-PEAL database were used, including the Neutral and Surprise expressions as well as those parts of the lighting subset that show moderate lighting conditions. A total of $B = 80$ hand-labeled training graphs were used to build the $\mathcal{G}^{\mathrm{ML}}$ graph. Due to the size of the database and the dependence of the $\mathcal{G}^{\mathcal{B}}$ on B, detection and recognition experiments with graphs extracted with $\mathcal{G}^{\mathcal{B}}$ were omitted.

Since the CAS-PEAL database consists of Chinese people only, it is not to be expected that all faces in the FRGC images are detected correctly. To allow to find more accurate landmark positions nonetheless, the face graphs were extracted in a two-stage detection scheme that we also applied in [29]: In the first stage, the graphs of the FRGC training set were detected by standardizing the images, placing the face graphs aligned to the eye positions, performing a scale move, and doing a three-step local move, using the $\mathcal{G}^{\mathrm{ML}}$ graph from CAS-PEAL. Afterwards, the extracted training graphs were employed to train two different $\mathcal{G}^{\mathrm{ML}}$ graphs, one for the controlled and one for the uncontrolled lighting conditions. Using these graphs, three different granularities of detection sequences were used to extract the final face graphs of the FRGC database, including a re-detection of the training graphs, the complete detection schedules can be found in Appendix D:



(a) Static **(b)** Local **(c)** Global

Figure 5.9: Exemplary Face Detection Results on FRGC: *This figure shows exemplary detection results of FRGC database images when applying the (a) Static, (b) Local, or (c) Global detection sequence to the controlled image from Figure 5.4(c) and the uncontrolled image shown in Figure 5.4(b).*

Static The Static detection sequence standardized the test image according to the hand-labeled eye positions and put the $\mathcal{G}^{\mathrm{ML}}$ graph, which was also standardized to the same graph positions, onto the images. The Gabor jets were extracted directly at these static positions.

Local The Local detection sequence also standardized the image and put the according $\mathcal{G}^{\mathrm{ML}}$ graph onto the standardized eye positions. But in opposition to Static, a small global and a small scale move was performed, and additionally the three-step local move as described in Section 3.2.3 was

applied.

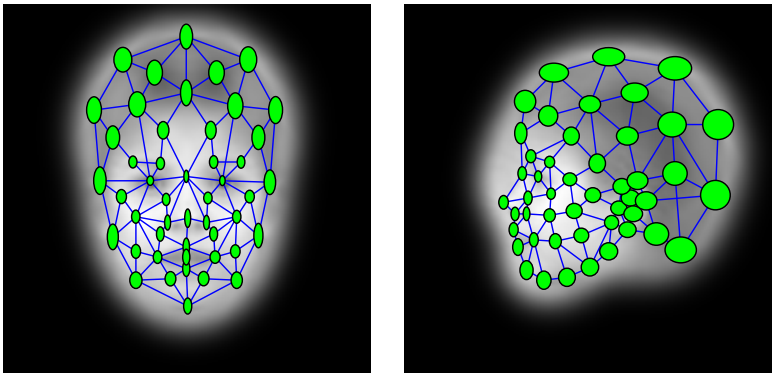
Global For the Global detection sequence, the controlled and uncontrolled target and query images were scaled down to 246×328 pixel and 560×420 pixel, respectively. Multi-scale and multi-angle face detection was applied to the target and query images, they were standardized according to the found scale and angle, and the same three-step local move is added. For the training set, the Local graphs were used to avoid later S^{IEC} training on misdetected graphs.

Exemplary results of these three detection sequences applied to one target and one query image are shown in Figure 5.9. Clearly, the Static graphs are placed very accurately at the eye positions, but show less accuracy around the nose and mouth regions. The Local graphs locate mouth and nose more precisely, while the Global graphs show some little scale or angle misdetections. The Static graphs as shown in Figure 5.9(a) are used as a baseline for my recognition experiments (see Section 5.3.3) since they are created without face detection or landmark localization. In comparison, the recognition experiments on the Local and Global graphs are used to show the impact of automatically placed landmarks and automatic face detection, respectively, on recognition accuracy.

5.2.4 Human Genetics

Although hand-labeled graphs are available for the Human Genetics database, in clinical practice it is of interests, how well automatically detected face graphs can be classified, in comparison to the hand-annotated graphs. Therefore, a leave-one-out cross-detection was performed on the Human Genetics database by successively taking one image from the database, training the $\mathcal{G}^{\mathrm{ML}}$ detector graph on the remaining hand-labeled face graphs, and extracting the face graph in the current probe image. This procedure was executed for the frontal faces and for the profile views separately. To be as comparable as possible, the detected face graphs were transformed back into the original images, which are already standardized, although not to the standard resolution of 168×224 pixel, but to 256×256 pixel.

In Figure 5.10, the node positioning errors that occurred during the ML face detection on the frontal and profile images is displayed on top of reconstructed images of average graphs for each view. Each ellipse shows the average landmark positioning error with the radius of one standard deviation in horizontal and vertical direction independently. While the inner facial landmarks at eyes, mouth, and nose were located quite reliably, the land-



(a) Frontal **(b)** Profile

Figure 5.10: Landmark Positioning Errors in the Human Genetics Database: *This figure displays node positioning errors of the ML detection for the (a) frontal and (b) profile views of the Human Genetic database, each with the radius of one standard deviation in horizontal and vertical direction, on top of reconstructed face graphs that was averaged of all syndromes.*

marks in the hair and on the ear were placed poorly. Since the ellipses in the frontal face show a vertical expansion, most probably the vertical scales of the graphs were detected incorrectly, a circumstance caused by the fact that the shapes of the faces are more irregular due to the genetic syndromes. Hence, often the average graph did not fit sufficiently well and, thus, landmark localization was impossible. Similarly, the inner facial landmarks of the profile views were located well, but due to misdetected scales and angles, the nodes of the ear and the back of the head were placed rather poorly. In general, the profile view faces were harder to detect because of the higher variance in shape, reflected in higher node positioning errors in Figure 5.10(b).

Unfortunately, I do not have the original facial images to detect the face graphs, but the faces in the images are already aligned, e. g., to their eye positions. Hence, the detection results are not comparable to the results reported by Böhringer *et al.* [8] and Vollmar *et al.* [91]. Nonetheless, since FIDA (cf. Section 4.4.3) allows manual geometrical image normalization, misdetections can be avoided with little human interaction. Thus, the classification results on the detected graphs that are detailed in Section 5.4.3 are indeed relevant for clinical practice.

5.3 Face Recognition Experiments

Subsequently, face recognition experiments are used to show whether the intrapersonal/extrapersonal classifier S^{IEC} is able to outperform the common graph similarity function $S^{\mathcal{G}}$ in terms of identity recognition accuracy. Additionally, the original a posteriori classifier of Moghaddam *et al.* [51] or, more specifically, Teixeira's [85] conversion S^{AP} (see Equation (4.3 – 4)) is reimplemented. Since neither Moghaddam *et al.* [51, 53] nor Teixeira [85] mentioned how to set the number of intrapersonal and extrapersonal principal components to keep, I made up $M_I = M_E = 30$, which are used throughout this section.

5.3.1 Recognition of FaceGen Identities

The experiments on the FaceGen dataset are split up into two parts. The first part shows the recognition results that use the ground truth (GT) graphs including exact node positions, while in the second part, the experiments are repeated using the graphs that were detected employing the $\mathcal{G}^{\mathcal{B}}$ and $\mathcal{G}^{\mathrm{ML}}$ detection algorithms (cf. Section 5.2.1).

Ground Truth Graphs

To be comparable with following experiments conducted on the detected graphs, the 100 identities that were used for face detection training were left out. Another 100 identities were used to train the IEC and AP classifiers, i. e., exploiting 100 intrapersonal and 9900 extrapersonal training pairs. For the remaining 800 identities, one of the two GT graphs was put into gallery and one into probe subset.

Recognition rates were computed for a variety of similarity functions, including both texture $S_{[\mathrm{jet}]} \in \{S_{[A]}, S_{[C]}, S_{[M]}, S_{[P]}, S_{[D]}, S_{[c]}, S_{[a]}\}$ and geometry $S_{[\mathrm{geo}]} \in \{S_{[\mathcal{E}]}, S_{[\mathcal{E}_{h/v}]}, S_{[\mathcal{L}]}, S_{[\mathcal{L}_{h/v}]}\}$ comparison functions. Figure 5.11 displays recognition rates (RR) for the $S^{\mathcal{G}}$, S^{IEC}, and S^{AP} systems, the results of the single comparison functions can be obtained in the rows or columns labeled with $\emptyset$. The $S^{\mathcal{G}}$ single function recognition results as given in Figure 5.11(a) are untrained and used as baseline. The texture comparison functions that do not include the phases ϕ_j of the Gabor jets range from 55.5% to 72.6% RR, the commonly used $S_{[A]}$ similarity function is the worst amongst those. When also including phase information, the recognition rates drop to around 45%. The geometry comparison functions also do not differ much in terms of $S^{\mathcal{G}}$ recognition accuracy, all four are in the order of 45% to 50% RR. Besides some rare cases, the combination of two comparison

	$\emptyset$	$S_{[\mathcal{E}]}$	$S_{[\mathcal{E}_{h/v}]}$	$S_{[\mathcal{L}]}$	$S_{[\mathcal{L}_{h/v}]}$
$\emptyset$		49.1	49.2	44.8	43.2
$S_{[A]}$	55.5	59.7	56.7	46.8	47.5
$S_{[C]}$	72.6	69.2	72.2	48.3	48.8
$S_{[M]}$	63.1	60.8	63.5	46.3	46.3
$S_{[P]}$	45.5	49.0	46.2	47.1	47.6
$S_{[D]}$	47.7	50.8	48.2	48.3	49.1
$S_{[c]}$	72.6	69.2	72.2	48.3	48.8
$S_{[a]}$	67.7	60.5	68.0	46.0	45.8

(a) $S^{\mathcal{G}}$

	$\emptyset$	$S_{[\mathcal{E}]}$	$S_{[\mathcal{E}_{h/v}]}$	$S_{[\mathcal{L}]}$	$S_{[\mathcal{L}_{h/v}]}$
$\emptyset$		78.1	86.1	85.7	88.1
$S_{[A]}$	88.7	91.6	94.3	94.2	94.3
$S_{[C]}$	90.7	92.1	93.8	94.2	94.6
$S_{[M]}$	88.0	90.3	93.0	93.0	93.3
$S_{[P]}$	79.3	83.1	89.1	89.7	89.3
$S_{[D]}$	84.2	88.2	92.3	92.1	91.7
$S_{[c]}$	96.7	94.2	94.0	94.1	96.0
$S_{[a]}$	97.6	96.7	95.6	95.6	97.3

(b) S^{IEC}

	$\emptyset$	$S_{[\mathcal{E}]}$	$S_{[\mathcal{E}_{h/v}]}$	$S_{[\mathcal{L}]}$	$S_{[\mathcal{L}_{h/v}]}$
$\emptyset$		85.1	72.0	94.1	97.1
$S_{[A]}$	93.2	97.3	93.8	99.0	99.1
$S_{[C]}$	94.7	99.0	92.0	99.7	99.7
$S_{[M]}$	95.8	98.8	93.7	99.7	100.
$S_{[P]}$	92.3	97.1	91.1	98.6	99.1
$S_{[D]}$	94.6	97.1	95.5	98.0	99.1
$S_{[c]}$	0.0	0.1	0.0	0.0	0.6
$S_{[a]}$	0.0	0.1	0.1	0.0	0.1

(c) S^{AP}

Figure 5.11: FaceGen Recognition Results: *This figure displays the obtained recognition rates employing (a)* $S^{\mathcal{G}}$*, (b)* S^{IEC}*, and (c)* S^{AP} *similarity measures combining each one geometry and one texture comparison function employing the GT FaceGen graphs. The green circles around the values emphasize the highest recognition rates.*

functions:

$$S^{\mathcal{G}}_{[\text{jet,geo}]}\left(\mathcal{G},\mathcal{G}'\right) = S^{\mathcal{G}}_{[\text{jet}]}\left(\vec{u}^{[\text{jet}]}\right) + S^{\mathcal{G}}_{[\text{geo}]}\left(\vec{u}^{[\text{geo}]}\right) , \tag{5.3-1}$$

which is comparable to Equation (4.3 – 20), does not improve recognition rates, at least not with this simple approach.

The recognition rates that were achieved employing S^{IEC} are given in Figure 5.11(b). In comparison to the untrained $S^{\mathcal{G}}$ results, every single recognition rate increased. Again, the texture comparison functions including phase information, i. e., $S_{[P]}$ and $S_{[D]}$ are worse than the ones without phases. The $S_{[c]}$ and $S_{[a]}$ comparison functions, which compare single Gabor wavelet responses independently, outperform the ones comparing whole Gabor jets. Due to the characteristics of the rendered FaceGen images, the texture of the faces is very smooth and, thus, high frequency Gabor wavelet responses are low in each face. Unfortunately, it is not to be expected that these functions perform that well in natural images. Stunningly, with approximately 85% RR the S^{IEC} geometry comparison functions can compete with the whole Gabor jet comparison functions. Hence, the geometry of the facial landmarks holds as much information about the identity as the texture does. The combination of texture and geometry comparison functions is able to increase recognition results perceptibly, e. g., combining $S_{[C]}$ and $S_{[\mathcal{L}_{h/v}]}$ results in 94.6% RR, while the single functions attain recognition rates of only 90.7% or 88.1%, respectively.

The S^{AP} recognition results shown in Figure 5.11(c) mostly exceed the results for the S^{IEC} and the $S^{\mathcal{G}}$ similarity measures. But weirdly, the functions $S_{[c]}$ and $S_{[a]}$, which perform best in Figure 5.11(b), do not succeed at all, both have 0% RR. Obviously, the choice of $M_I = M_E = 30$ is inappropriate in this case. The geometry comparison functions work quite nicely, especially the $S_{[\mathcal{L}_{h/v}]}$ similarity function outperforms the S^{IEC} results, in combination with $S_{[M]}$ it reaches the perfect 100% RR. Hence, the S^{AP} approach on average outperforms my S^{IEC} approach on artificially generated images.

Detected Graphs

The next step is to test whether the recognition results can be repeated when the graphs were detected automatically. Again, a variety of comparison functions were used to perform recognition experiments. The attained recognition rates can be obtained from Table 5.3. As shown in Table 5.3(a), most of the texture comparison functions are very stable against misdetections. Dependent on the recognition algorithms, the numbers for the single comparison functions are comparable between the different detection algorithms. For the

	Graph	$S_{[A]}$	$S_{[C]}$	$S_{[M]}$	$S_{[P]}$	$S_{[D]}$	$S_{[c]}$	$S_{[a]}$
$S^{\mathcal{G}}$	GT	55.5%	72.6%	63.1%	45.5%	47.7%	72.6%	67.7%
	$\mathcal{G}^{\mathcal{B}}$	54.2%	71.4%	61.1%	34.4%	46.1%	71.4%	65.5%
	$\mathcal{G}^{\mathrm{ML}}$	55.0%	73.0%	62.6%	37.7%	46.2%	73.0%	67.4%
S^{IEC}	GT	88.7%	90.7%	88.0%	79.4%	84.2%	96.7%	97.6%
	$\mathcal{G}^{\mathcal{B}}$	89.6%	91.1%	88.7%	64.7%	84.9%	96.5%	97.6%
	$\mathcal{G}^{\mathrm{ML}}$	89.9%	91.7%	89.7%	72.6%	85.4%	96.5%	97.6%
S^{AP}	GT	93.2%	94.7%	95.9%	92.4%	94.6%	0.0%	0.0%
	$\mathcal{G}^{\mathcal{B}}$	93.6%	95.9%	95.0%	50.0%	94.4%	0.0%	0.0%
	$\mathcal{G}^{\mathrm{ML}}$	92.1%	93.0%	94.2%	74.5%	95.5%	0.0%	0.0%

(a) Texture

	Graph	$S_{[\mathcal{E}]}$	$S_{[\mathcal{E}_{h/v}]}$	$S_{[\mathcal{L}]}$	$S_{[\mathcal{L}_{h/v}]}$
$S^{\mathcal{G}}$	GT	49.1%	49.2%	44.9%	43.2%
	$\mathcal{G}^{\mathcal{B}}$	12.1%	12.7%	20.1%	17.6%
	$\mathcal{G}^{\mathrm{ML}}$	13.6%	14.6%	19.9%	17.0%
S^{IEC}	GT	78.1%	86.1%	85.7%	88.1%
	$\mathcal{G}^{\mathcal{B}}$	15.2%	15.7%	30.6%	30.5%
	$\mathcal{G}^{\mathrm{ML}}$	22.5%	22.4%	40.2%	40.5%
S^{AP}	GT	85.1%	72.0%	94.1%	97.1%
	$\mathcal{G}^{\mathcal{B}}$	6.5%	0.4%	19.1%	24.6%
	$\mathcal{G}^{\mathrm{ML}}$	13.0%	1.5%	32.5%	35.9%

(b) Geometry

Table 5.3: FaceGen Recognition Results: *This table shows $S^{\mathcal{G}}$, S^{IEC}, and S^{AP} recognition results on the FaceGen graphs detected with $\mathcal{G}^{\mathrm{ML}}$ and $\mathcal{G}^{\mathcal{B}}$ in comparison to the ground truth (GT) graphs, exploiting (a) texture and (b) geometry information.*

S^{IEC} recognition algorithm, the graphs detected with $\mathcal{G}^{\mathrm{ML}}$ give even better results than GT data, i. e., the $\mathcal{G}^{\mathrm{ML}}$ detection algorithm probably induce identity-dependent detection errors (cf. [60]), which are learned by the S^{IEC} training. Another hint for a good node placement (or identity-dependently misplaced nodes) is a high recognition rate when employing $S_{[P]}$ since this function is affected by node placement errors. Obviously, the graphs detected with $\mathcal{G}^{\mathrm{ML}}$ perform better than the ones of $\mathcal{G}^{\mathcal{B}}$ in Table 5.3(a), but not as good as the GT graphs.

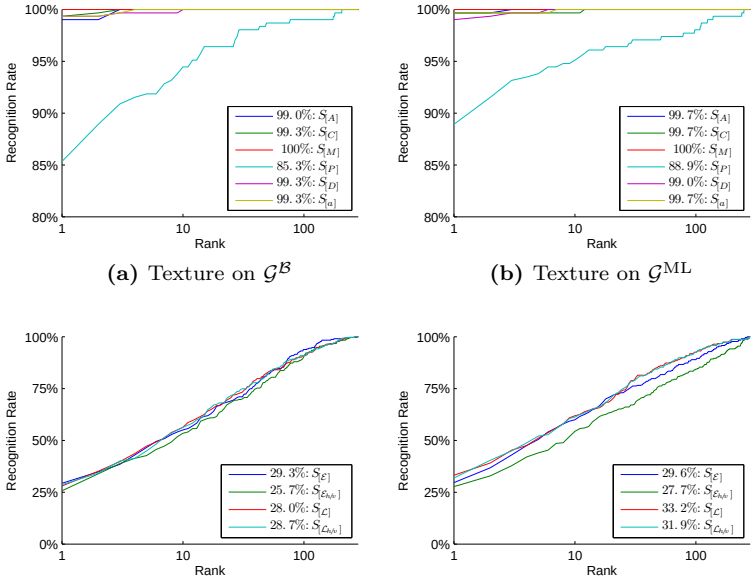
The geometry comparison functions cannot handle misdetections that well. Compared to the GT graphs, the recognition rates as given in Table 5.3(b) drop dramatically when detected graphs are used. Again, the recognition rates for the $\mathcal{G}^{\mathrm{ML}}$ graphs are better than the ones of the $\mathcal{G}^{\mathcal{B}}$ graphs, especially when the S^{IEC} recognition system is applied. The S^{AP} results are very poor, even for those comparison functions that showed good results for the GT graphs. In any case, the geometry comparison results are far from being useful, and also the combination of texture and geometry comparison functions are in general worse than using the texture functions solo (the numbers of combining texture and geometry comparison functions are not given here).

5.3.2 Recognition under Scale, Expression, and Illumination Variation

The results that are achieved in the artificial images can be extended to natural facial images as well. To prove this proposition, recognition experiments were executed on the CAS-PEAL database and the results are compared to the ones from [23, 22].

Distance Subset

The faces of the Distance experiment are easy to recognize when they were found correctly since all faces show the same neutral facial expression. Figure 5.12 shows the results of the Distance recognition experiments that were conducted on the graphs extracted with ML and EBGM, see Section 5.2.2 for details. The untrained $S^{\mathcal{G}}$ results of the texture comparison functions as given in Figures 5.12(a) and 5.12(b) exhibit the simplicity of recognition, both graph types give 100% rank 1 recognition rate for the $S_{[M]}$ similarity function, and just small errors for the other comparison functions using absolute values, although notably the commonly used $S_{[A]}$ similarity function is the weakest performer. Only $S_{[P]}$ and also $S_{[D]}$, which exploit phase information of Gabor jets, do not work that well. The geometry of the detected



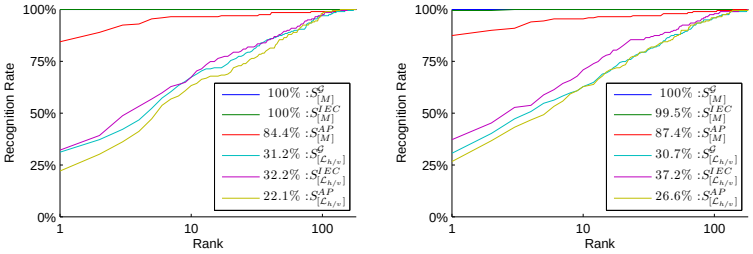
(a) Texture on $\mathcal{G}^{\mathcal{B}}$ **(b)** Texture on $\mathcal{G}^{\mathrm{ML}}$

(c) Geometry on $\mathcal{G}^{\mathcal{B}}$ **(d)** Geometry on $\mathcal{G}^{\mathrm{ML}}$

Figure 5.12: $S^{\mathcal{G}}$ Recognition Results on the Distance Subset: *This figure displays the results of the untrained Graph similarity function $S^{\mathcal{G}}$ on the Distance subset of the CAS-PEAL database. It compares the recognition results obtained on the graphs detected with $\mathcal{G}^{\mathcal{B}}$ or $\mathcal{G}^{\mathrm{ML}}$, based on the comparison of texture and geometry. For a direct comparison, each plot includes rank 1 recognition rates for each employed similarity function.*

graphs cannot be used for identification employing the $S^{\mathcal{G}}$ graph similarity function, the recognition rates shown in Figures 5.12(c) and 5.12(d) do not exceed 35%. Still, it is apparent that the ML graphs perform slightly better than the EBGM graphs.

The recognition rate of $S_{[M]}$ cannot be improved. Nevertheless, random 100 identities of the Distance subset were chosen to train the S^{IEC} and the S^{AP} recognition system. The remaining 181 identities were used for calculating the recognition rates that are given in Figure 5.13. Besides small noise, the results of the $S_{[M]}$ functions stay at the maximum when the S^{IEC} training is applied, but decrease dramatically for S^{AP}. When applying S^{IEC} training



(a) $\mathcal{G}^{\mathcal{B}}$ **(b)** $\mathcal{G}^{\mathrm{ML}}$

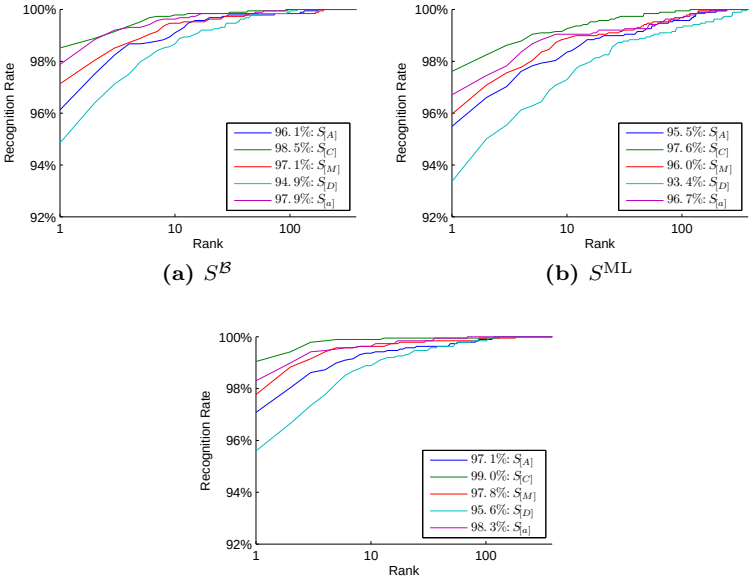
Figure 5.13: $S^{\mathrm{IEC}}/S^{\mathrm{AP}}$ **Recognition Results on the Distance Subset**: *This figure displays the recognition results of the* S^{IEC} *and the* S^{AP} *training in relation to the untrained* $S^{\mathcal{G}}$ *similarity function on the Distance subset of the CAS-PEAL database. It compares the recognition results obtained on the graphs detected with EBGM or ML, based on* $S_{[M]}$ *texture and* $S_{[\mathcal{L}_{h/v}]}$ *geometry comparison functions.*

on the geometry comparison functions, the recognition rate increases a little on the EBGM graphs, and a little more on the ML graphs, but still these rates are far from being useful. S^{AP} again decreases recognition accuracy. Due to these results, I drop the geometry comparison functions as well as the S^{AP} comparison from the further recognition experiments.

Expression Subset

Facial images are said to be difficult to identify when they show facial expressions. This section investigates, how well the comparison of face graphs is suited for recognizing faces with various facial expressions. In the first experiment, the face graphs of the images showing expressions Laugh, Frown, Surprise, Closed eyes, and Opened mouth were matched against the Neutral expression. Figure 5.14 provides the results[4] for the recognition tests on the EBGM and ML graphs when no S^{IEC} training was applied. In opposition to the prevalent belief, the recognition rates are very high, $S_{[C]}$ (and identically $S_{[c]}$) reaches 98.5% RR on the EBGM graphs. The ML results are a little lower, probably due to the already discussed misdetections of scale or angle, but still 97.6% of the images were identified correctly. To show the impact of

[4]The recognition rates of the $S_{[P]}$ similarity function were below 65% and are not included in Figure 5.14



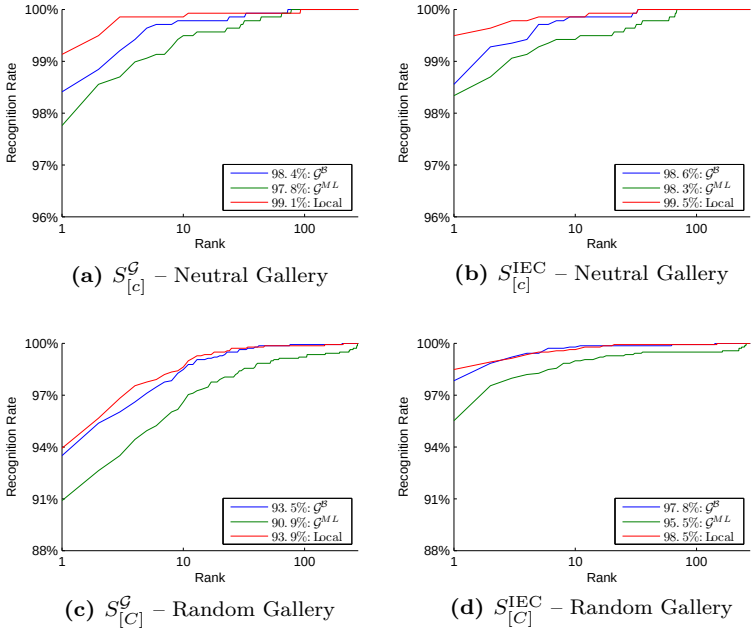
(a) $S^{\mathcal{B}}$ **(b)** S^{ML}

(c) Local

Figure 5.14: $S^{\mathcal{G}}$ Recognition Results on Expression Variations: *This figure displays the results of the untrained Graph similarity function $S^{\mathcal{G}}$ on the Expression subset of the CAS-PEAL database. It compares the recognition results obtained on the graphs detected with (a) EBGM, (b) ML, or (c) ML landmark localization on standardized images, based on texture comparison.*

the misdetections on the recognition rates, the approach similar to the Local set of the FRGC database (cf. Section 5.2.3) is used by standardizing the images according to their hand-labeled eye positions and performing a local adaptation of the landmark positions (using the $\mathcal{G}^{\mathrm{ML}}$ local move as given in Section 3.3.5). Employing these graphs, the recognition accuracy can even be improved to 99.0% RR.

To see if these rates can further be improved by training the S^{IEC} recognition system, a random subset of 100 identities was used as training set, while the remaining 277 identities were used for testing, again putting the Neutral graphs into gallery and the remaining expressions into probe set. The recognition results employing the $S_{[c]}$ similarity function, which performed best



(a) $S^{\mathcal{G}}_{[c]}$ – Neutral Gallery **(b)** $S^{\text{IEC}}_{[c]}$ – Neutral Gallery

(c) $S^{\mathcal{G}}_{[C]}$ – Random Gallery **(d)** $S^{\text{IEC}}_{[C]}$ – Random Gallery

Figure 5.15: Recognition Results on Expression Variations: *This figure shows the $S^{\mathcal{G}}$ and S^{IEC} recognition results of the Expression subset of the CAS-PEAL database, based on the $S_{[c]}$ or the $S_{[C]}$ texture comparison function. In (a) and (b), the Neutral faces were used as gallery, while in (c) and (d), for each identity, an image with random facial expression was put into gallery.*

in this experiment, are given in Figure 5.15(b). Since the results are not directly comparable to the results of Figure 5.14, the replicated corresponding $S^{\mathcal{G}}$ similarities are shown in Figure 5.15(a). Conspicuously, all recognition rates increased and with 99.5% RR the perfect recognition of the Distance experiments is missed only marginally. Fortunately, with approximately 98.5% RR the automatically detected graphs perform nearly as well.

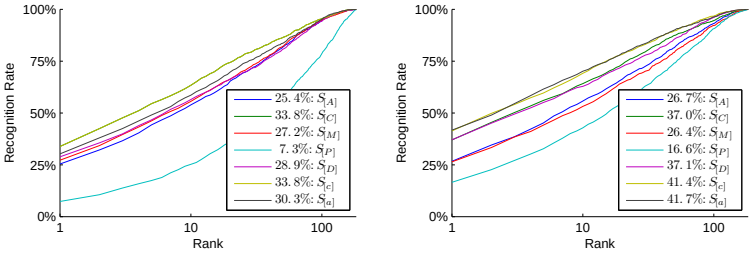
A more difficult experiment is to take one of the six expressions randomly as gallery and use the remaining five expressions as probe. Hence, the task was to discriminate between similarities based on identity and similarities

based on similar facial expressions. For example, if the probe image showed a Closed eye expression, the similarities to other Closed eye expressions in the gallery needed to be lower than the similarity to the, e. g., Laugh expression of the correct identity in the gallery. The identities taken for training and testing were identical to the ones in the previous experiment. The results of $S^{\mathcal{G}}$ and S^{IEC} can be obtained from Figures 5.15(c) and 5.15(d), in this case the $S_{[C]}$ function, which generates the best results, is reported, throughout. Obviously, the S^{IEC} training learns to disregard expressions quite well. Overall, the recognition rates on the Local graphs increased from 94% of $S^{\mathcal{G}}_{[C]}$ to 98.5% of $S^{\mathrm{IEC}}_{[C]}$, but these very good results can unfortunately not be reached on the graphs detected with $\mathcal{G}^{\mathrm{ML}}$.

Lighting Subset

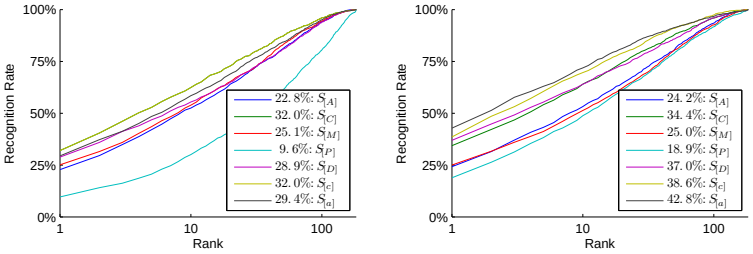
Facial images are indeed hard to identify when the illumination changes. The experiments on the Lighting subset match probe images taken under fluorescent light sources from different directions to gallery images with ambient light. Figure 5.16 shows the recognition results on this subset. As expected, the recognition rates are very low, the best untrained $S^{\mathcal{G}}$ results, which are generated by the $S_{[c]}$-function, throughout, do not exceed 40% RR on the standardized images and are below 35% RR on the detected faces.

The IEC training was executed on a subset of random 50 identities, the remaining 183 identities are used as probe. Since the task was to compare ambiently illuminated gallery images with fluorescently illuminated probe images, only graph pairs including one ambient and one fluorescent illumination were used for IEC training. The recognition results of the trained IEC classifiers are given in the second column of Figure 5.16. There are three major points to note: Firstly, recognition accuracy moderately increased in each detection setup, the gain is in the order of 10 percentage points. Secondly, the comparison functions comparing individual Gabor wavelet responses, i. e., $S_{[c]}$ and $S_{[a]}$ give the highest rates. Likely, it can be learned quite well, which variations in Gabor wavelet responses stem from identity and which are introduced by lighting. The third point is that the disparity similarity function $S_{[D]}$, which performed poorly in the previous experiments, is amongst the best functions, at least in the Local images. Hence, the phase values of Gabor wavelet responses seem to be more stable against illumination changes than the absolute values, at least including these information can increase recognition rates. In his MSc thesis [31], Haufe conducted experiments exploiting phases of Gabor wavelet responses for recognition. Although he still used the simple disparity estimation as given in Appendix B.1, his recognition rates on the lighting subset outrivaled the results reported in [23, 22].



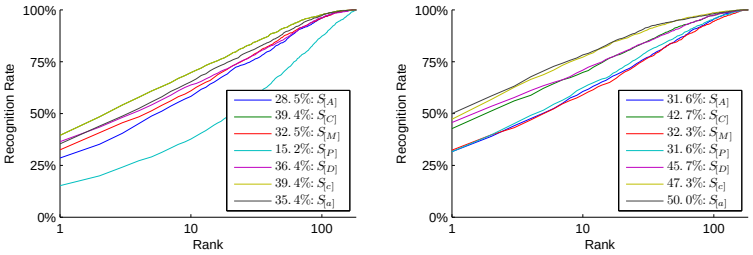
(a) $S^{\mathcal{G}}$ on EBGM

(b) S^{IEC} on EBGM



(c) $S^{\mathcal{G}}$ on ML

(d) S^{IEC} on ML



(e) $S^{\mathcal{G}}$ on Local

(f) S^{IEC} on Local

Figure 5.16: Recognition Results on Illumination Variations: *This figure displays $S^{\mathcal{G}}$ (see (a), (c), and (e)) and S^{IEC} (cf. (b), (d), and (f)) face recognition results of the Lighting subset of the CAS-PEAL database, employing several texture comparison functions on face graphs detected with EBGM, ML, and ML landmark localization on standardized images.*

Algorithm	Distance	Expression	Lighting
PCA	74.2%	53.7%	8.2%
PCA+LDA	93.5%	71.3%	21.8%
GPCA+LDA	100%	90.6%	44.8%
LGBPHS[5]	99%	95%	51%
$S^{\mathcal{G}}$ on Local	100% ($S_{[C]}$)	98.8% ($S_{[C]}$)	32.3% ($S_{[c]}$)
S^{IEC} on Local	100% ($S_{[C]}$)	99.3% ($S_{[C]}$)	38.3% ($S_{[c]}$)
$S^{\mathcal{G}}$ on $\mathcal{G}^{\text{ML}}$	100% ($S_{[M]}$)	96.8% ($S_{[C]}$)	26.2% ($S_{[c]}$)
S^{IEC} on $\mathcal{G}^{\text{ML}}$	99.5% ($S_{[M]}$)	97.7% ($S_{[C]}$)	28.5% ($S_{[c]}$)
$S^{\text{IEC}}_{[c,d]}$ on Local	96.0%	98.4%	56.8%
$S^{\text{IEC}}_{[c,d]}$ on $\mathcal{G}^{\text{ML}}$	93.5%	96.7%	42.0%

Table 5.4: Result Comparison on CAS-PEAL Database: *This table shows the baseline results of the CAS-PEAL experiments reported by Gao et al. [23, 22] and the results of my recognition experiments on half- and fully-automatically detected face graphs.*

Parts of his experiments are repeated in the following (here I use the disparity estimation as given in Appendix B.4):

Comparison to Other Results

The CAS-PEAL database [23, 22] provides recognition results for default algorithms like PCA or PCA+LDA (see Sections 4.2.3 and 4.2.4, respectively) and more recent algorithms like GPCA+LDA and LGBPHS[5] (cf. Sections 4.2.5 and 4.2.7, respectively). All these results were generated with a gallery size of 1040 and, therefore, including also persons that were not in the accordant probe sets. The results in the previous experiments used only reduced gallery sizes by removing the gallery identities that are not in the probe sets. For a more fair comparison to the results reported in [23, 22], some experiments were recomputed on the large gallery. In case of the Lighting subset, also the 185 images with incandescent lighting were included into the probe set.

Table 5.4 shows the results as reported in [23, 22], all of which rely on

[5]The recognition rates for LGBPHS were estimated from the graphic in [22].

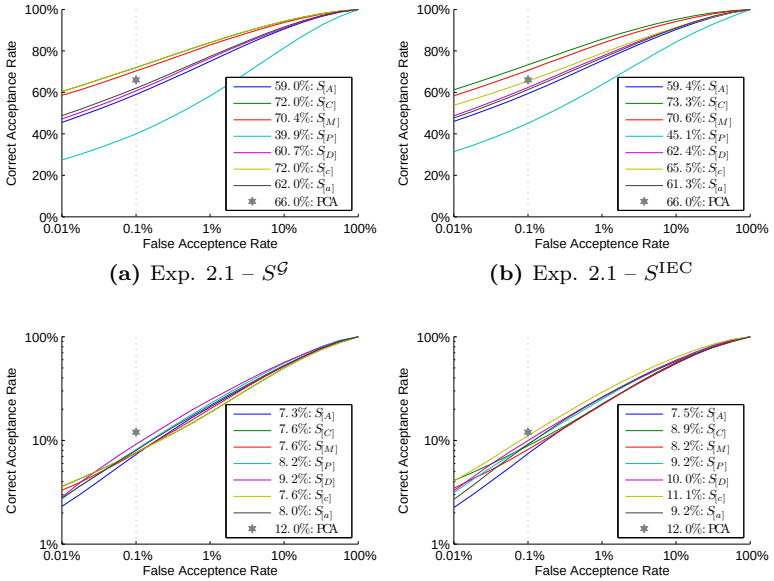
half-automatically extracted image features, i. e., using images standardized according to their hand-labeled eye positions. Additionally, the $S^{\mathcal{G}}$ and the S^{IEC} results from my experiments and the results on the $S_{[c,d]}$ function proposed by Haufe [31] are included in Table 5.4. In any of the subsets, the recognition rates of the eigenfaces (PCA) and the Fisher faces (PCA+LDA) are far below any Gabor wavelet based recognition function like Gabor graph comparison, GPCA+LDA, or LGBPHS. Using pixel gray values for face recognition directly seems not to be a good idea.

For the Distance subset, my results replicate the very high recognition rates of GPCA+LDA and LGBPHS, only the trained classifier on the detected graphs decay slightly by misidentifying one single face graph. On the Expression subset, the results of the untrained Graph similarity function already outperform the best reported numbers, both on the half-automatically generated Local graphs as well as on the graphs that are detected fully-automatically. Employing an additional IEC training, the resulting 99.3% RR is not only outperforming the 95% RR reported by Gao *et al.* [22], with 0.7% the error rate is one order of magnitude below the 5% error rate of LGBPHS. Although the probe sizes differ since the IEC training used parts of the test set, the recognition results are directly comparable since the gallery is identical (cf. Section 4.1.1).

The increase of gallery size most severely impacts recognition accuracy in the Lighting subset, the recognition rates drop perceptibly (cf. Figure 5.16 and Table 5.4). When utilizing only absolute values of Gabor wavelet responses, the reported 51% of the LGBPHS cannot be reached in any experiment. Nonetheless, it is to note that the S^{IEC} similarity always outperforms the untrained $S^{\mathcal{G}}$ similarity. When exploiting the phase values of the Gabor wavelet responses, the $S_{[d]}$ similarity function already reaches 45.9% RR (number not given in Table 5.4), and in combination with $S_{[c]}$, 56.8% RR is accomplished. As Haufe [31] showed, this value can be further increased by using grid graphs with more nodes. Still, this recognition rate is not sufficient for real-life applications, but it shows the potential of both the Gabor phases and the IEC training. Unfortunately, the high recognition rates for the Local set can only partly be replicated with the automatically detected face graphs.

5.3.3 Large Scale Verification Results

To test how well IEC recognition scales to large databases, verification tests were performed on the FRGC database, too. All tests were repeated on the Static and on the Local graphs, both of which rely on images standardized to the hand-labeled eye positions, as well as on the Global graphs, which

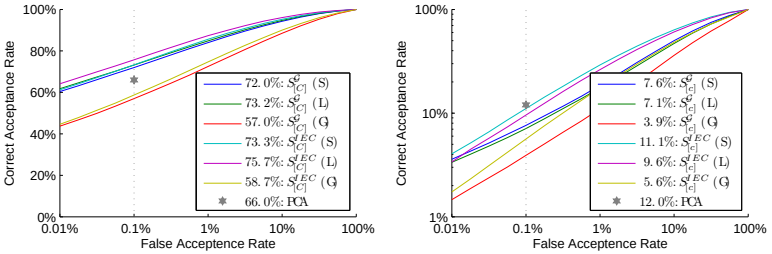


(a) Exp. 2.1 – $S^{\mathcal{G}}$ **(b)** Exp. 2.1 – S^{IEC}

(c) Exp. 2.4 – $S^{\mathcal{G}}$ **(d)** Exp. 2.4 – S^{IEC}

Figure 5.17: Verification Results on Static FRGC Graphs: *This figure shows ROC curves for experiment 2.1 and 2.4 of the FRGC database using $S^{\mathcal{G}}$ and S^{IEC} on several texture comparison functions. For easy comparison, each plot includes verification rates at 0.1% FAR for each employed comparison function and the PCA baseline algorithm. The ROC curves in (c) and (d) for experiment 2.4 are plotted with logarithmical axes of ordinate.*

are detected by the $\mathcal{G}^{\mathrm{ML}}$ algorithm as described in Section 3.3. For the experiments, the FRGC database [65] give baseline results in terms of receiver operating characteristics (ROC), the verification rates (VR) at FAR = 0.1% are reported to be 66% for experiment 2.1, i. e., comparing faces under controlled lighting conditions, and 12% for experiment 2.4, i. e., comparing images under controlled with images under uncontrolled illuminations. Since the experiments were repeated unmodifiedly, the results are also given in terms of ROC and VR. As a baseline test for the different face detection granularities, Figure 5.17 shows ROC curves for the Static graphs, employing a variety of texture comparison functions. On experiment 2.1, once more

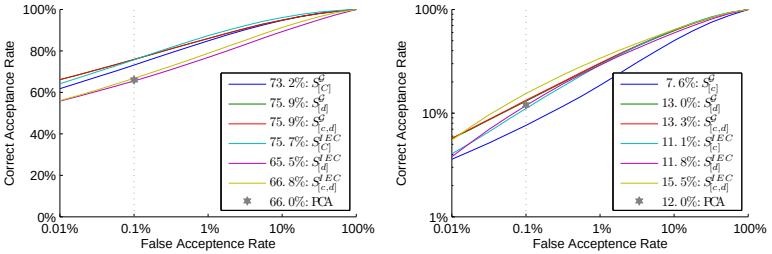


(a) Experiment 2.1 **(b)** Experiment 2.4

Figure 5.18: Detection Setup Impact on Verification: *This figure shows the impact of face detection and landmark localization on verification accuracy based on the FRGC experiments 2.1 and 2.4. The results are given for three the different face detection schedules Static (S), Local (L), and Global (G), and for $S^{\mathcal{G}}$ and S^{IEC}.*

the $S_{[C]}$ comparison function works best, followed by $S_{[M]}$, both are outperforming the 66% VR of the eigenfaces. $S_{[D]}$ and $S_{[c]}$ are better suited for experiment 2.4, but still both are missing the 12% VR mark. Conspicuously, $S_{[c]}$ and $S_{[a]}$ dropped accuracies when training on experiment 2.1, compare Figures 5.17(a) and 5.17(b). Obviously, the precondition of unimodal distributions of the T_I and T_E classes are not fulfilled, and the assumption of linear independence of the elements of $\vec{u}_I^{[c]}$ and $\vec{u}_E^{[c]}$ is violated. Although this is also true for experiment 4, in this case $S_{[c]}$ and $S_{[a]}$ are well trainable.

To test how landmark positioning and face detection influence recognition accuracy, the best functions $S_{[C]}$ and $S_{[c]}$ on experiment 2.1 and 2.4, respectively, are tested on the Static, Local and Global detection setup (see Section 5.2.3). The results are shown in Figure 5.18. In any case, S^{IEC} training improved verification rates, but unfortunately only little. Because the number of training data is huge, probably there are training algorithms that perform better than S^{IEC}. For experiment 2.1 as shown in Figure 5.18(a), the result attained with the Local graphs is better than using the Static ones. Hence, applying a local move helps recognizing controlledly illuminated faces. Still, there are misdetected scales and angles so that the results on the Global graphs drop remarkably. Apparently, the very good results reported in [29] could not be repeated, caused by the fact that a different detection strategy was employed (cf. Section 3.4). For the uncontrolled images in experiment 2.4 (see Figure 5.18(b)), the local move does not improve verification accuracy,



(a) Experiment 2.1 **(b)** Experiment 2.4

Figure 5.19: FRGC Verification Results of Haufe's Functions: *This figure shows the $S^{\mathcal{G}}$ and S^{IEC} verification rates generated with the similarity functions proposed by Haufe [31] (a) on the Local graphs for experiment 2.1 and (b) on the Static graphs for experiment 2.4.*

here the Static graphs are suited better for recognition.

Employing the similarity functions that Haufe [31] introduced, the verification rates can further be improved, the resulting ROC curves are presented in Figure 5.19. For experiment 2.1, $S_{[d]}$ and $S_{[c,d]}$ give 75.9% VR for the untrained $S^{\mathcal{G}}$ similarity function based on the Local graphs, while $S^{\text{IEC}}_{[c,d]}$ reaches 15.5% VR on experiment 2.4 based on the Static graphs. Unfortunately, IEC training on the $S_{[d]}$ function does not improve recognition accuracy, in both experiments the verification results dropped remarkably when comparing $S^{\mathcal{G}}_{[d]}$ and $S^{\text{IEC}}_{[d]}$.

5.4 Classification Experiments

This section shows the ability of the IEC recognition system to learn, how to classify image properties on the basis of a small amount of training data. The first classification experiments estimate, which facial expression was made by the person shown in the image, or under which lighting condition the images was taken. Afterwards, the classified lighting condition is used to increase identification accuracies. Later on, the same classifier is used to classify genetic syndromes.

5.4.1 Facial Expression Classification

Classifying facial expression might be of interest for the human-robot-communications [26, 48] since the robot could interpret the facial expression and perform actions according to these. The CAS-PEAL database provides facial images with different facial expressions and with different lighting conditions, however it does not include combinations of both as they would appear in an uncontrolled or outdoor scenario. Automatic classification of facial expressions based on single static images is a tough problem, even for humans [113, 95]. Usually, face databases do not store real facial expressions, but the probands most often perform facial expressions as they think they should look like since they were told to show these expressions and the expressions did not come naturally [113].

Certainly, facial expressions can be determined best when the texture and the geometry of the facial features are regarded (cf. also [48]). Some expressions like Opened mouth and Frown change the arrangement of the facial features, in this case mainly the nodes of the mouth, while in the Closed eyes expression, primarily the texture of the eyes varies. Figure 5.20 presents the classification rates when combining one texture and one geometry comparison function. Apparently, the texture similarity functions that include phase information, i. e., $S_{[P]}$ and $S_{[D]}$, which perform poorly in most identification tasks, are much better suited for expression classification. Another point is that computing the similarities $S^{\mathcal{G}+c}$ or $S^{\mathrm{IEC}+c}$ of the probe graphs to category centers $\mathcal{G}_c$ (reconstructions of the category centers are given in Figure 5.22(a)) is on average better than holding all training graphs in the gallery. Furthermore, combining texture comparison function $S_{[P]}$ and geometry comparison function $S_{[\mathcal{E}]}$ increases classification accuracy in the $S^{\mathrm{IEC}+c}$ setup given in Figure 5.20(d), although detected node positions usually are not precise enough, e. g., to be used for identification (cf. Section 5.3.2).

Figure 5.21 shows confusion matrices of expression classification errors, using the best combination of functions from Figure 5.20(d). For each category, the number of correctly classified probe images is higher than for any other category. Especially the Neutral[6] and the Closed eyes expressions seem to be classifiable easily, while the other expressions are more often mixed up. In Figure 5.21(c), the confusion matrix for the geometry similarity function $S_{[\mathcal{E}]}$ is presented. Obviously, the detected node positions are not that well suited for classification,[6] but still the correct category was chosen most often for each facial expression. Nonetheless, combining geometry and tex-

[6]The graphs for the Neutral expression were extracted employing a different $\mathcal{G}^{\mathrm{ML}}$ graph. Certainly, the IEC classifier was able to learn this interrelation, especially when only geometrical information is used, cf. Figure 5.21(c).

	$\emptyset$	$S_{[\mathcal{E}]}$	$S_{[\mathcal{E}_{h/v}]}$	$S_{[\mathcal{L}]}$	$S_{[\mathcal{L}_{h/v}]}$
$\emptyset$		48.8	48.0	46.7	45.6
$S_{[A]}$	56.8	65.6	64.4	47.7	48.3
$S_{[C]}$	61.7	65.4	65.6	47.2	47.7
$S_{[M]}$	61.2	63.1	67.1	46.8	47.0
$S_{[P]}$	63.3	62.9	63.5	48.3	49.6
$S_{[D]}$	61.2	67.3	63.6	50.2	53.1
$S_{[c]}$	61.7	65.4	65.6	47.2	47.7
$S_{[a]}$	59.3	59.4	64.4	46.7	46.3

(a) $S^{\mathcal{G}}$

	$\emptyset$	$S_{[\mathcal{E}]}$	$S_{[\mathcal{E}_{h/v}]}$	$S_{[\mathcal{L}]}$	$S_{[\mathcal{L}_{h/v}]}$
$\emptyset$		49.2	46.6	49.1	49.3
$S_{[A]}$	60.2	65.8	67.1	65.3	67.3
$S_{[C]}$	61.1	65.2	66.5	63.7	65.6
$S_{[M]}$	63.6	67.2	67.9	65.8	67.8
$S_{[P]}$	66.1	64.6	63.8	63.2	65.2
$S_{[D]}$	64.7	69.2	68.2	68.2	68.8
$S_{[c]}$	61.5	60.2	60.0	59.7	60.5
$S_{[a]}$	63.3	62.5	61.7	61.3	63.0

(b) S^{IEC}

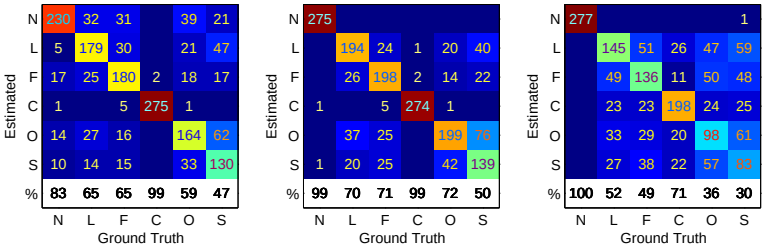
	$\emptyset$	$S_{[\mathcal{E}]}$	$S_{[\mathcal{E}_{h/v}]}$	$S_{[\mathcal{L}]}$	$S_{[\mathcal{L}_{h/v}]}$
$\emptyset$		54.7	56.1	52.7	52.2
$S_{[A]}$	64.2	73.0	69.1	54.0	54.4
$S_{[C]}$	45.5	64.2	50.9	54.1	54.6
$S_{[M]}$	62.6	71.6	68.6	53.4	53.2
$S_{[P]}$	74.7	75.0	74.8	54.9	56.0
$S_{[D]}$	67.6	74.7	69.5	56.0	58.4
$S_{[c]}$	45.5	64.2	50.9	54.1	54.6
$S_{[a]}$	67.7	66.4	73.9	53.0	52.8

(c) $S^{\mathcal{G}+c}$

	$\emptyset$	$S_{[\mathcal{E}]}$	$S_{[\mathcal{E}_{h/v}]}$	$S_{[\mathcal{L}]}$	$S_{[\mathcal{L}_{h/v}]}$
$\emptyset$		56.4	53.2	52.8	54.7
$S_{[A]}$	65.2	74.5	72.2	71.2	73.0
$S_{[C]}$	58.6	69.5	67.6	66.8	69.1
$S_{[M]}$	64.5	74.1	72.4	71.2	73.2
$S_{[P]}$	74.5	76.6	73.8	72.6	75.6
$S_{[D]}$	69.7	77.0	75.4	74.1	76.0
$S_{[c]}$	71.7	66.7	63.6	60.6	63.9
$S_{[a]}$	70.3	72.1	67.0	64.5	68.0

(d) $S^{\text{IEC}+c}$

Figure 5.20: Facial Expression Classification: *This figure shows facial expression classification rates when combining each one texture and one geometry comparison function. In (a) and (b), probe graphs were compared to gallery graphs, while in (c) and (d), probe graphs were compared to averaged graphs $\mathcal{G}_c$ of the categories.*



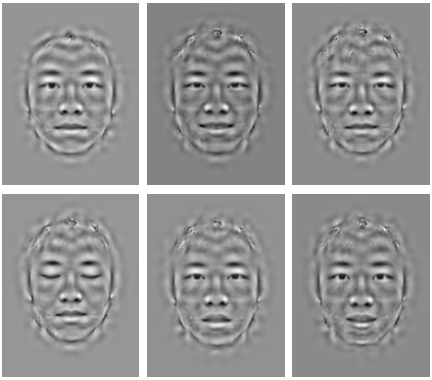
(a) $S^{\mathrm{IEC}+c}_{[D]}$ **(b)** $S^{\mathrm{IEC}+c}_{[D,\mathcal{E}]}$ **(c)** $S^{\mathrm{IEC}+c}_{[\mathcal{E}]}$

Figure 5.21: Facial Expression Confusion: *This figure shows matrices of facial expression classification confusion based on (a) the* $S^{\mathrm{IEC}+c}_{[D]}$ *texture and (c) the* $S^{\mathrm{IEC}+c}_{[\mathcal{E}]}$ *geometry comparison function, and (b) the combination* $S^{\mathrm{IEC}+c}_{[D,\mathcal{E}]}$ *of both functions. The last row of each matrix shows classification rates of the single expressions in percent.*

ture decreases misclassifications in nearly all categories (cf. Figures 5.21(a) and 5.21(b)).

The results presented in Figures 5.20 and 5.21 rely on the Local graphs, i. e., including image standardization according to hand-labeled eye positions. Fortunately, the classification results drop only marginally when the graphs detected with the $\mathcal{G}^{\mathrm{ML}}$ or $\mathcal{G}^{\mathcal{B}}$ algorithm are used. The highest classification rates of 75.6% and 74.0% are reached by $S^{\mathrm{IEC}+c}_{[D,\mathcal{E}]}$ and $S^{\mathrm{IEC}+c}_{[D,\mathcal{E}_{h/v}]}$, respectively (the complete results are not reported here). Notably, no hand-labeled graphs of the Laugh, Frown, and Open mouth expressions were used during face detection and landmark localization.

The achieved classification rates of 75% are below rates that are reported in literature. For example, Martin *et al.* [48] reported expression classification accuracies of over 90% using *active appearance models*. This rather big gap can be explained by differences in the experimental setup: First, the employed database differs. While Martin *et al.* [48] used the FEEDTUM mimic database [92] with video sequences of 18 people, I used the CAS-PEAL database with still images of 377 identities. Second, Martin *et al.* [48] used hand-labeled graphs of 1438 images of all expressions to train their active appearance model, while here graphs where detected on the basis of only 18 hand-labeled graphs including only three of six expressions. The last and and most important point is that the identities Martin *et al.* [48] used for training



(a) Averaged Expressions

(b) Averaged Illuminations

Figure 5.22: Reconstructed Average Category Graphs: *This figure displays images reconstructed from average Local face graphs $\mathcal{G}_c$ of (a) six facial expression categories (from top left to bottom right:* ***N****,* ***L****,* ***F****,* ***C****,* ***O****, and* ***S****) and (b) fifteen lighting condition categories (from left to right: +90°, +45°, 0°, -45°, and -90°; from top to bottom:* ***U****,* ***M****, and* ***D****).*

Setup	$S_{[A]}$	$S_{[C]}$	$S_{[M]}$	$S_{[P]}$	$S_{[D]}$	$S_{[c]}$	$S_{[a]}$	$S_{[a,d]}$	$S_{[c,d]}$
$S^{\mathcal{G}}$	91.9%	94.3%	92.8%	83.2%	94.5%	94.3%	94.8%	96.0%	96.5%
S^{IEC}	92.6%	95.8%	93.4%	90.9%	95.7%	95.8%	95.3%	97.2%	97.5%
$S^{\mathcal{G}+c}$	89.9%	81.8%	87.7%	86.8%	94.2%	81.8%	92.7%	95.2%	95.8%
$S^{\mathrm{IEC}+c}$	89.7%	86.9%	88.6%	89.9%	94.2%	89.9%	92.8%	96.2%	95.4%

(a) Local

Setup	$S_{[A]}$	$S_{[C]}$	$S_{[M]}$	$S_{[P]}$	$S_{[D]}$	$S_{[c]}$	$S_{[a]}$	$S_{[a,d]}$	$S_{[c,d]}$
$S^{\mathcal{G}}$	92.0%	94.3%	92.3%	78.4%	93.0%	94.3%	93.7%	94.9%	95.4%
S^{IEC}	92.8%	95.0%	91.7%	87.0%	94.9%	94.8%	93.0%	96.4%	96.3%
$S^{\mathcal{G}+c}$	86.0%	72.4%	81.6%	83.0%	91.5%	72.4%	89.5%	92.4%	92.8%
$S^{\mathrm{IEC}+c}$	88.6%	82.5%	86.5%	83.5%	91.8%	88.6%	92.5%	94.5%	93.7%

(b) $\mathcal{G}^{\mathrm{ML}}$

Table 5.5: Classification of Lighting Conditions: *This table displays S^{IEC} lighting condition classification results for the CAS-PEAL database utilizing (a) the graphs extracted with the Local setup and (b) the graphs detected by $\mathcal{G}^{\mathrm{ML}}$, based on texture comparison functions.*

their classifiers were identical to the identities in their test set, training and test images even stem from the same video sequences. In opposition, the identities used for training and testing were disjoint in the tests reported in this section.

5.4.2 Lighting Condition Classification

The classification of the direction of the light source can be done identically. Again, both approaches, i. e., comparing two graphs directly or comparing graphs with averaged lighting condition graphs $\mathcal{G}_c$ (see Figure 5.22(b) for reconstructions of the $\mathcal{G}_c$ graphs for the different illuminations) were employed. Since landmark positions should not be affected by the lighting condition, geometry comparison functions were left out in these experiments. The classification results are given in Table 5.5. Obviously, the lighting conditions can be classified easily. In opposition to the expression classification, the comparison to average graphs, i. e., with $S^{\mathcal{G}+c}$ or $S^{\mathrm{IEC}+c}$ is not as accurate as comparing two graphs with $S^{\mathcal{G}}$ or S^{IEC}. Seemingly, valuable information

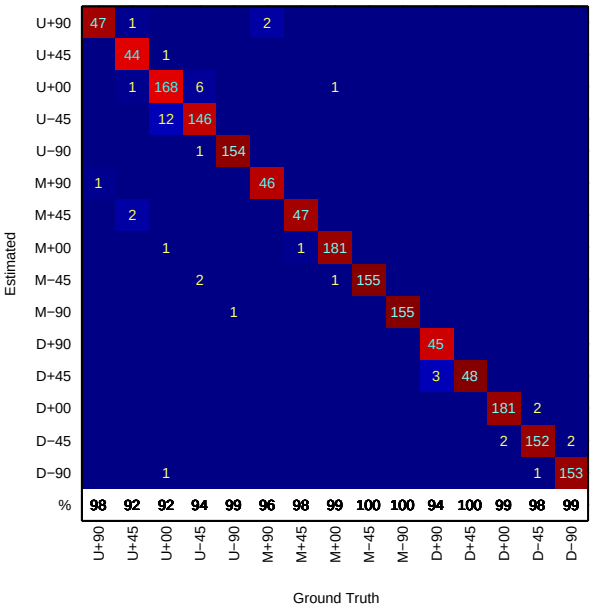


Figure 5.23: Lighting Classification Confusions: *This figure displays the confusion matrix of the $S^{\mathrm{IEC}}_{[c,d]}$ comparison function in the lighting condition classification experiment on the Local graphs of the CAS-PEAL Lighting subset.*

about the lighting condition was lost during graph averaging, possibly due to the fact that node positions have been placed automatically. Especially, the $S_{[C]}$ comparison function that is one of the best functions in S^{IEC} cannot handle averaged graphs well, the $S^{\mathrm{IEC}+c}$ classification rates drop considerably. Again, the classification results decrease marginally when the graphs detected with the $\mathcal{G}^{\mathrm{ML}}$ algorithm are used instead of the Local graphs, for the $S_{[A]}$ similarity function the classification accuracy even increased. Still, with over 95% CR the $S_{[C]}$ comparison function suites best for this classification task, but also $S_{[D]}$ and $S_{[c]}$ results are valuable. When applying the $S^{\mathrm{IEC}}_{[c,d]}$ function introduced by Haufe [31], the result can even be increased to 97.5% CR.

Figure 5.23 shows the single classification confusions done by the $S^{\mathrm{IEC}}_{[c,d]}$

classifier. In almost all cases, misclassifications missed the correct lighting condition only by one step, either in horizontal or vertical direction. Most of the errors occur in the **U** conditions with light from above, the frontal **M** conditions are classified near-to-perfect. Note that the number of training examples differ for the lighting conditions. While the 0° frontal condition used 50 identities for S^{IEC} training, the -45° and -90° conditions got 44 identities, and the +45° and +90° conditions had to manage on only 14 training identities.

One possible application of the lighting condition classification is the improvement of recognition. This is achieved by using IEC recognizers specifically trained for this kind of illumination, in turn simply by selecting appropriate training image pairs. The recognition procedure is as follows: Given the training set of 50 identities, the S^{IEC} classifier is trained to classify the lighting condition. Afterwards, the same training set is split up into the 15 different lighting conditions and for each, a single S^{IEC} classifier is trained to recognize the face by comparing the graphs to ambiently illuminated gallery graphs. For a given probe graph, first its lighting condition is classified, and afterwards the correspondingly trained IEC classifier is used to identify the face.

Table 5.6 embraces recognition results for some combinations of texture comparison functions used for classification and for recognition, as well as the correct classification rates (CR) for the single comparison functions. Furthermore, the recognition results that are achieved when the recognizer was chosen according to the ground truth (GT) lighting condition is added. Obviously, the $S_{[a]}$ and $S_{[a,d]}$ recognition functions result in the highest recognition rates, while $S_{[c]}$ and $S_{[c,d]}$ are best suited for classification. Fortunately, the estimated lighting conditions are good enough for recognition rates not to drop severely compared to GT conditions. The results of the Local graphs as given in Table 5.6(a) are as expected higher than the ones from the graphs detected with $\mathcal{G}^{\mathrm{ML}}$ in Table 5.6(b). For the latter, the recognition rates are already lower when the GT lighting condition information is utilized, but again misclassification does not harm identification. Apparently, highest classification rates do not necessarily coincide with highest recognition accuracy. In some cases the recognition results using the estimated lighting condition are even better than exploiting GT illumination information. Note that the resulting recognition rates are comparable to the results shown in Figure 5.16, but not to the results reported in [23, 22] and Table 5.4.

Classification	Recognition $S_{[C]}$	$S_{[D]}$	$S_{[c]}$	$S_{[a]}$	$S_{[a,d]}$	$S_{[c,d]}$	CR
$S_{[C]}$	44.5%	48.1%	52.1%	58.0%	61.8%	54.8%	95.8%
$S_{[D]}$	44.7%	47.9%	52.4%	58.0%	62.1%	55.2%	95.7%
$S_{[c]}$	44.7%	48.0%	52.2%	58.1%	61.8%	55.0%	95.8%
$S_{[a]}$	44.5%	48.0%	52.0%	57.8%	61.8%	55.1%	95.3%
$S_{[a,d]}$	44.8%	47.8%	52.6%	58.0%	61.8%	55.0%	97.2%
$S_{[c,d]}$	44.7%	47.9%	52.5%	58.0%	61.9%	55.0%	97.5%
GT	44.9%	48.0%	52.6%	58.1%	62.3%	55.4%	

(a) Local

Classification	Recognition $S_{[C]}$	$S_{[D]}$	$S_{[c]}$	$S_{[a]}$	$S_{[a,d]}$	$S_{[c,d]}$	CR
$S_{[C]}$	36.3%	39.2%	44.4%	51.3%	53.0%	47.9%	95.0%
$S_{[D]}$	36.3%	39.3%	44.8%	51.4%	53.0%	47.9%	94.9%
$S_{[c]}$	36.5%	39.3%	44.6%	51.4%	53.0%	47.7%	94.8%
$S_{[a]}$	36.3%	39.2%	44.5%	51.7%	53.1%	47.8%	93.0%
$S_{[a,d]}$	36.5%	39.3%	44.7%	51.6%	52.9%	47.9%	96.4%
$S_{[c,d]}$	36.2%	39.3%	44.6%	51.6%	52.9%	47.9%	96.3%
GT	36.6%	39.4%	44.7%	51.6%	53.1%	48.0%	

(b) $\mathcal{G}^{\mathrm{ML}}$

Table 5.6: Face Recognition after Illumination Classification: *This table shows the recognition results of the Lighting experiment that classified the lighting condition and used accordingly trained recognizers, employing different similarity functions for classification and recognition. In the last columns, the classification rates (CR) are given, while the last rows show recognition rates when recognizers where chosen according to ground truth (GT) lighting conditions.*

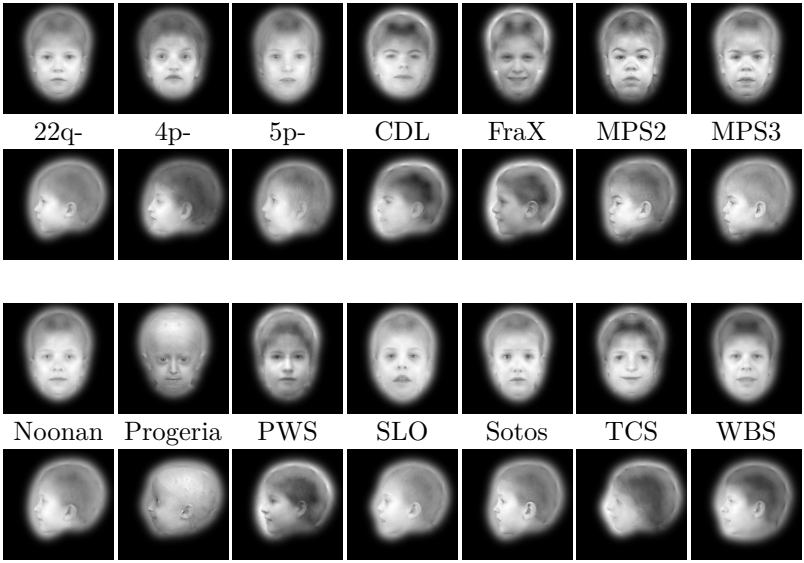


Figure 5.24: Reconstructions of Average Syndrome Graphs: *This figure displays reconstructed averaged hand-labeled face graphs in frontal and profile view from within the Human Genetics database. For displaying purposes, Gabor jets of the graphs were generated with Gabor wavelet set* $\Gamma^{\{g\}}$ *(cf. Chapter 6).*

5.4.3 Genetic Syndrome Classification

The very same classification approach can be used for classifying genetic syndromes that have an effect on the facial phenotype. Since the Human Genetic database includes facial images in frontal and profile views for most patients, the combination of both should be exploited during classification. Fortunately, in the IEC approach this is easily possible by adding up S^{IEC} or $S^{\mathrm{IEC}+c}$ similarity values of different types:

$$\begin{aligned} S^{\mathrm{IEC}+c}_{[\mathrm{total}]}(\mathcal{G}_c, \mathcal{G}) = & S^{\mathrm{IEC}+c}_{[\mathrm{jet,front}]}(\mathcal{G}_c, \mathcal{G}) + S^{\mathrm{IEC}+c}_{[\mathrm{geo,front}]}(\mathcal{G}_c, \mathcal{G}) + \\ & S^{\mathrm{IEC}+c}_{[\mathrm{jet,profile}]}(\mathcal{G}_c, \mathcal{G}) + S^{\mathrm{IEC}+c}_{[\mathrm{geo,profile}]}(\mathcal{G}_c, \mathcal{G}) \ . \end{aligned} \tag{5.4–1}$$

Additionally, when one of the two views is not available, it can simply be left out in the totalization and still the resulting $S^{\mathrm{IEC}+c}_{[\mathrm{total}]}$ values are comparable.

Furthermore, weights for the different data or view types could be defined in order to increase classification accuracy, but here no weights are used.

Figure 5.24 shows reconstructions of the average graphs $\mathcal{G}_c$ for the frontal and profile views of the single syndromes. The corresponding syndromes can be classified correctly from these images by medical experts [7]. Obviously, the syndrome is encoded in the averaged graphs and, hence, testing probe graphs to the category centers might be a good idea. The results of the LOOCV experiments executed on the hand-labeled graphs can be obtained from Table 5.7. In this case, the texture and geometry comparison functions were applied independently, and tests were executed on the frontal and the profile graphs, and on the combination of both. Independent of the comparison function or the comparison setup, frontal faces always performed better than the profile views, but they were always outperformed by the combination both. The classification rates of the $S^{\mathcal{G}}$ comparison are given in Table 5.7(a). This function can be seen as a 1-nearest neighbor classifier, employing graph similarity functions instead of Euclidean distances. Already this simple classification seems to outperform the multinomial logistic regression reported by Böhringer *et al.* [7]. With 60.7% or 57.5% CR, texture or geometry comparison is better than the reported 55.4% or 35.1% CR, respectively.

When comparing probe graphs to average graphs $\mathcal{G}_c$ (cf. Table 5.7(b)), the classification results of most texture comparison functions drop, while the geometry comparison function results further raise. One exception of the texture comparison is the $S_{[P]}$ comparison function, which includes both the absolute values and the phase values of the Gabor wavelet responses. When comparing to average graphs, the $S_{[P]}$ function outperforms every other texture comparison function by far. In turn this means that the Gabor phases, which are neglected by most state-of-the-art recognition and classification algorithms, boost classification rates, especially when they are compared to averaged phases. Employing $S^{\mathcal{G}}_{[c,d]}$ or $S^{\mathcal{G}+c}_{[c,p]}$ from Haufe [31], the resulting 66.2% CR is outperforming any of the "traditional" texture comparison functions (the numbers are not given in Table 5.7).

As Tables 5.7(c) and 5.7(d) show, executing IEC training further improves classification rates in most cases, only the $S_{[c]}$ and $S_{[a]}$ comparison functions perform worse. Obviously, the number of training examples is too low in comparison to the quite large number of categories and, hence, the number of variables to be estimated for the $S^{\text{IEC}}_{[a]}$ or $S^{\text{IEC}}_{[c]}$ comparison functions is too high. All other functions, for which only a couple of variables need to be estimated by the S^{IEC} training, perform well and most of the observations comparing $S^{\mathcal{G}}$ with $S^{\mathcal{G}+c}$ can be repeated. Briefly, $S_{[P]}$ and $S_{[\mathcal{E}]}$ or $S_{[\mathcal{E}_{h/v}]}$ achieve highest classification accuracies in the $S^{\text{IEC}+c}$ test, while $S_{[C]}$ and $S_{[D]}$ perform best in the S^{IEC} test.

	$S_{[A]}$	$S_{[C]}$	$S_{[M]}$	$S_{[P]}$	$S_{[D]}$	$S_{[c]}$	$S_{[a]}$	$S_{[\mathcal{E}]}$	$S_{[\mathcal{E}_{h/v}]}$	$S_{[\mathcal{L}]}$	$S_{[\mathcal{L}_{h/v}]}$
Front	48.2%	56.4%	52.3%	47.7%	51.8%	56.4%	56.0%	46.8%	46.3%	42.7%	44.5%
Left	48.4%	44.7%	42.3%	39.5%	54.4%	44.7%	50.7%	37.7%	39.5%	28.4%	28.4%
Both	58.9%	60.7%	58.4%	52.1%	60.3%	60.7%	58.0%	56.6%	57.5%	45.2%	45.2%

(a) $S^{\mathcal{G}}$

	$S_{[A]}$	$S_{[C]}$	$S_{[M]}$	$S_{[P]}$	$S_{[D]}$	$S_{[c]}$	$S_{[a]}$	$S_{[\mathcal{E}]}$	$S_{[\mathcal{E}_{h/v}]}$	$S_{[\mathcal{L}]}$	$S_{[\mathcal{L}_{h/v}]}$
Front	47.7%	37.6%	41.7%	54.6%	52.8%	37.6%	50.9%	55.0%	53.7%	42.2%	42.7%
Left	41.4%	36.3%	34.9%	58.1%	50.7%	36.3%	45.6%	47.0%	54.9%	40.9%	38.6%
Both	50.7%	45.7%	45.7%	64.4%	58.0%	45.7%	59.4%	65.8%	67.1%	53.4%	53.9%

(b) $S^{\mathcal{G}+c}$

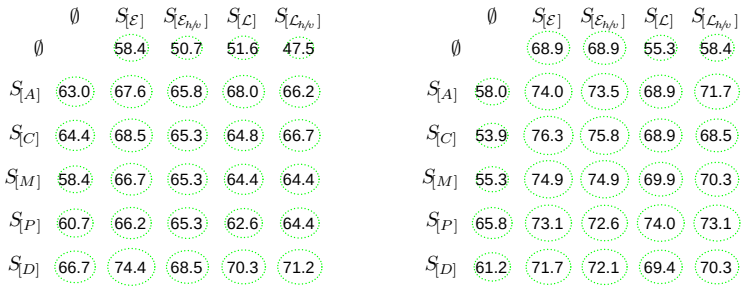
	$S_{[A]}$	$S_{[C]}$	$S_{[M]}$	$S_{[P]}$	$S_{[D]}$	$S_{[c]}$	$S_{[a]}$	$S_{[\mathcal{E}]}$	$S_{[\mathcal{E}_{h/v}]}$	$S_{[\mathcal{L}]}$	$S_{[\mathcal{L}_{h/v}]}$
Front	50.0%	58.3%	51.4%	54.6%	63.3%	41.3%	24.3%	49.5%	46.8%	41.3%	44.0%
Left	49.8%	51.2%	49.3%	44.7%	57.7%	35.8%	23.3%	43.3%	37.2%	33.0%	32.6%
Both	63.0%	64.4%	58.0%	60.7%	66.7%	41.6%	23.3%	58.4%	50.7%	51.6%	47.5%

(c) S^{IEC}

	$S_{[A]}$	$S_{[C]}$	$S_{[M]}$	$S_{[P]}$	$S_{[D]}$	$S_{[c]}$	$S_{[a]}$	$S_{[\mathcal{E}]}$	$S_{[\mathcal{E}_{h/v}]}$	$S_{[\mathcal{L}]}$	$S_{[\mathcal{L}_{h/v}]}$
Front	45.9%	39.9%	40.4%	54.1%	53.7%	21.6%	44.0%	57.8%	62.4%	45.4%	47.2%
Left	48.4%	45.1%	45.6%	47.4%	49.3%	20.5%	40.5%	52.1%	52.6%	38.6%	40.9%
Both	58.0%	53.9%	55.3%	65.8%	61.2%	20.1%	39.3%	68.9%	69.4%	55.3%	57.5%

(d) $S^{\mathrm{IEC}+c}$

Table 5.7: Syndrome Classification: *This table shows the leave-one-out cross-validation results employing a single similarity function on either the frontal faces, the profile views, or the combination of both. The classification rates are given for* $S^{\mathcal{G}}$, $S^{\mathcal{G}+c}$, S^{IEC}, *and* $S^{\mathrm{IEC}+c}$, *executed on the hand-labeled graphs of the Human Genetic dataset.*



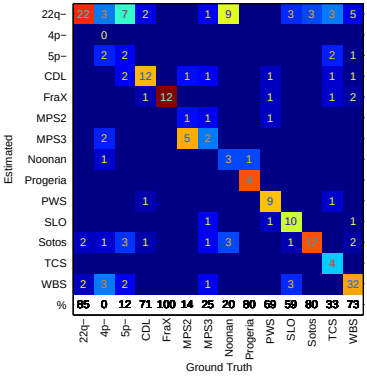
(a) S^{IEC} **(b)** $S^{\mathrm{IEC}+c}$

Figure 5.25: Texture and Geometry for Syndrome Classification: *This figure displays the classification rates for the leave-one-out cross-validation experiments on the hand-labeled graphs of the Human Genetic database for S^{IEC} and $S^{\mathrm{IEC}+c}$ setups combining each one texture and one geometry comparison function.*

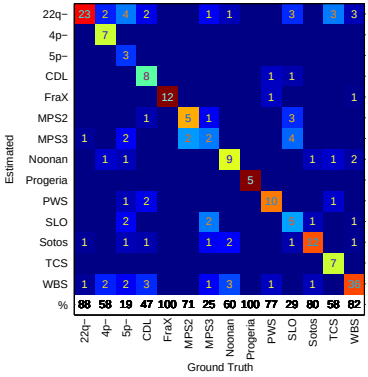
Combining each one of these texture and geometry functions using Equation (5.4 – 1) in turn increases classification accuracy. Figure 5.25 displays the classification results for S^{IEC} and $S^{\mathrm{IEC}+c}$ setup. For S^{IEC}, with 74.4% CR the combination of $S_{[D]}$ and $S_{[\mathcal{E}]}$ works best, while the highest rate of 76.3% CR was achieved by $S^{\mathrm{IEC}+c}_{[C,\mathcal{E}]}$. This rate is more than 15% higher than the 60.4% CR that was reported by Böhringer [7]. Interestingly, when employing $S^{\mathrm{IEC}+c}$, the combination of the geometry functions with the $S_{[P]}$ function, which performs best solo, is exceeded by the combination of $S_{[\mathcal{E}]}$ with $S_{[C]}$, which has a low solo accuracy. Seemingly, the phases of the Gabor wavelet responses exploited by $S_{[P]}$ code approximately the same information as the edges exploited by $S_{[\mathcal{E}]}$ and, hence, combining both does not help much.

The classification results using the automatically detected graphs are given in Table 5.8. As before, all texture comparison functions perform nearly as good as for the hand-labeled graphs. For the $S^{\mathcal{G}+c}$ and $S^{\mathrm{IEC}+c}$ classifiers, $S_{[A]}$, $S_{[C]}$, and $S_{[M]}$ work even better on detected graphs than on the hand-labeled ones (compare Table 5.8 with Tables 5.7(b) and 5.7(d)). Obviously, syndrome-dependent detection errors improved classification. Unfortunately, the geometry comparison functions do not work that well on the detected graphs, and combining them with texture comparison does not expedite.

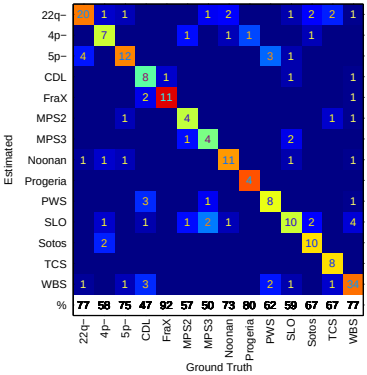
The confusion matrices for the best performing texture ($S^{\mathrm{IEC}+c}_{[P]}$) and geometry ($S^{\mathrm{IEC}+c}_{[\mathcal{E}]}$) comparison functions and the best combination ($S^{\mathrm{IEC}+c}_{[C,\mathcal{E}]}$)



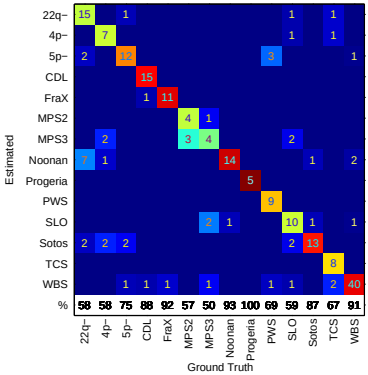
(a) $S^{\mathrm{IEC}+c}_{[D]}$ on Detected Graphs



(b) $S^{\mathrm{IEC}+c}_{[P]}$ on hand-labeled Graphs



(c) $S^{\mathrm{IEC}+c}_{[\mathcal{E}]}$ on hand-labeled Graphs



(d) $S^{\mathrm{IEC}+c}_{[C,\mathcal{E}]}$ on hand-labeled Graphs

Figure 5.26: Confusion Matrices for Syndrome Classification: *This figure shows matrices of syndrome classification confusions employing combinations of frontal and profile facial images. In (a) the graphs detected by the* $\mathcal{G}^{\mathrm{ML}}$ *algorithm were used, while (b) - (d) are based on hand-labeled graphs. (a) and (b) exploit texture comparisons, whereas (c) use geometrical information and (d) a combination of both.*

	$S_{[A]}$	$S_{[C]}$	$S_{[M]}$	$S_{[P]}$	$S_{[D]}$	$S_{[c]}$	$S_{[a]}$	$S_{[\mathcal{E}]}$	$S_{[\mathcal{E}_{h/v}]}$	$S_{[\mathcal{L}]}$	$S_{[\mathcal{L}_{h/v}]}$
$S^{\mathcal{G}}$	56.6%	54.8%	55.3%	43.8%	53.4%	54.8%	58.0%	28.3%	27.4%	20.1%	21.0%
S^{IEC}	59.8%	59.4%	59.4%	57.1%	66.2%	38.8%	23.7%	22.8%	21.5%	19.2%	14.6%
$S^{\mathcal{G}+c}$	61.2%	56.6%	59.4%	53.9%	58.0%	56.6%	65.3%	26.5%	32.9%	18.3%	18.3%
$S^{\mathrm{IEC}+c}$	61.6%	58.0%	55.7%	48.4%	57.1%	24.7%	42.5%	27.9%	30.1%	20.1%	18.7%

Table 5.8: Syndrome Classification on Detected Graphs: *This table shows the classification rates of the combination of detected graphs in frontal and profile view from the Human Genetic database. The experiments were executed with the classification methods $S^{\mathcal{G}}$, S^{IEC}, $S^{\mathcal{G}+c}$, and $S^{\mathrm{IEC}+c}$, and several texture and geometry comparison functions were employed.*

are given in Figure 5.26. These matrices also include the classification results for each single syndrome. Basically, the single syndrome classification errors are compliant with the ones reported by Vollmar *et al.* [91], also here FraX and Progeria have highest accuracies, while MPS3 could not be classified well. Clearly, there are different preferred comparison techniques for the different syndromes. Texture comparison in terms of $S^{\mathrm{IEC}+c}_{[P]}$ similarities play an important role for some syndromes like 22q-, FraX, and Progeria, while failing for 5p-, MPS3, or Noonan. On the other hand 5p-, Noonan, and SLO are classified well by geometrical $S^{\mathrm{IEC}+c}_{[\mathcal{E}]}$ graph comparison, whereas MPS2, PWS, TCS, and WBS perform best on the combination $S^{\mathrm{IEC}+c}_{[P,\mathcal{E}]}$ of both graph comparison types.

In summary, $S^{\mathrm{IEC}+c}_{[C,\mathcal{E}]}$ seems to be the best combination, but obviously not for all syndromes. Actually, every syndrome might have its own comparison function that give best classification accuracy, e. g., combining $S^{\mathrm{IEC}+c}_{[P,\mathcal{E}]}$ (cf. Figures 5.26(b) and 5.26(c)) for the classification of 22q- works better as $S^{\mathrm{IEC}+c}_{[C,\mathcal{E}]}$ given in Figure 5.26(d). Also in [79], we reported that the combination $S^{\mathrm{IEC}+c}_{[P,\mathcal{E}]}$ to be most accurate for the classification of acromegaly, although $S^{\mathrm{IEC}+c}_{[D,\mathcal{E}]}$ gave good results, too. Still, we used $S_{[D]}$ with the old disparity estimation in [79], while disparities are estimated including phase correction (cf. Appendix B.4) in this thesis.

Chapter 6

Reconstruction from Gabor Graphs

The aim of this chapter is the reconstruction of an image from a Gabor graph using amplitude and phase information of the Gabor jets. Previous attempts of reconstructing from Gabor wavelet responses were, amongst others, made by Wundrich [103, 101, 102, 104] and Pötzsch [71, 70].

Wundrich *et al.* [101, 102] used only the amplitudes of the Gabor wavelet responses to reconstruct an image. They took the Gabor jets extracted at all positions of the Gabor transformed image $\mathcal{T}$ and reconstructed them with an iterative algorithm [101]. In his diploma thesis, Wundrich [103] investigated the theoretical background of the reconstruction from sampled Gabor wavelet responses, with the sampling being applied in all 4 dimensions, i. e., $\vec{x}$ and $\vec{k}$ of the Gabor wavelets. He presented the framework for my reconstruction algorithm by using dual Gabor wavelets ψ^d, although he calculated the dual Gabor wavelets differently than I do in Section 6.1.3.

On the other hand, Pötzsch [71] reconstructed single Gabor jets with full absolute and phase information and removed background information from them. He [70] also established an algorithm for reconstructing Gabor graphs by defining Voronoi areas around the nodes and reconstructing the Gabor jets locally. Although the resulting reconstructions of face graphs look quite well, sometimes the gray value jumps at the borderlines between the Voronoi areas. With this method, Pötzsch reconstructed a *phantom face* (cf. Section 6.6.1) of a probe image by using the Gabor jet of each bunch (i. e. the darker jets in Figure 2.7(c) on page 32) that best matches the image. In the phantom face that Pötzsch *et al.* [70] showed, gray value jumps are developed strongly.

This chapter presents an algorithm that reconstructs natural images from Gabor graphs, which, in contrast to Pötzsch *et al.* [70], processes Gabor jets at few landmark positions in a global fashion. After settling the theoretical background, specifying the dual Gabor wavelets, and extending the Gabor wavelet family to include gray level and color information, an iterative algorithm that is inspired by the one presented by Wundrich *et al.* [104, 101] is constituted. Subsequently, an advanced initial condition for the iterative

algorithm is engendered by approximating the layers $\mathcal{T}_{\vec{k}_j}$ of the Gabor transformed image from the given Gabor graph. At the end of this chapter, some examples of reconstructed Gabor graphs are shown and the time performance of the different parts of the reconstruction algorithm is discussed.

6.1 Inverse Gabor Wavelet Transform

6.1.1 Inverse 2D Wavelet Transformation

To come closer to the goal of a reconstruction from Gabor graphs, the general two-dimensional inverse wavelet transform that is introduced in Section 2.1.2 should be recapitulated. The signal $\mathcal{I}(\vec{x})$ can be reconstructed from the wavelet responses $\mathcal{T}\left(s, \vartheta, \vec{t}\right)$ using:

$$\mathcal{I}(\vec{x}) = \frac{1}{C_\chi} \int\limits_{\mathbb{R}^+} \int\limits_{0}^{2\pi} \int\limits_{\mathbb{R}^2} \mathcal{T}\left(s, \vartheta, \vec{t}\right) \, \chi_{s,\vartheta,\vec{t}}(\vec{x}) \, \frac{d^2t \, d\vartheta \, ds}{s^3} \,, \tag{6.1-1}$$

with the admissibility constant C_χ that is dependent on the wavelet χ:

$$C_\chi = \int\limits_{\mathbb{R}^2 \setminus \{0\}} \frac{\left|\check{\chi}\left(\vec{\xi}\right)\right|^2}{\left|\vec{\xi}\right|^2} d^2\xi \,. \tag{6.1-2}$$

Using the substitutions (the calculation is based on [103], the missing proof is added in Appendix A.1):

$$\vec{\xi} = s \, Q(\vartheta) \, \vec{\omega} \,, \qquad \frac{d^2\xi}{ds \, d\vartheta} = \frac{\left|\vec{\xi}\right|^2}{s} \,, \tag{6.1-3}$$

the factor $1/s^3$ in the reconstruction formula from Equation (6.1 – 1) can be explained:

$$\begin{aligned} C_\chi &= \int_0^{2\pi} \int_{\mathbb{R}^+} \frac{|\check{\chi}(s\,Q(\vartheta)\;\vec{\omega})\,|^2}{\left|\vec{\xi}\right|^2}\;\frac{d^2\xi}{ds\,d\vartheta}\;ds\,d\vartheta \\ &= \int_0^{2\pi} \int_{\mathbb{R}^+} \left|\frac{1}{s}\;\check{\chi}_{s,\vartheta}(\vec{\omega})\right|^2\;\frac{ds\,d\vartheta}{s} \\ &= \int_0^{2\pi} \int_{\mathbb{R}^+} |\check{\chi}_{s,\vartheta}(\vec{\omega})|^2\;\frac{ds\,d\vartheta}{s^3} \\ &= C_\chi(\vec{\omega}) \end{aligned} \tag{6.1-4}$$

since it emerges in the admissibility constant C_χ, too. As obtainable from Equation (6.1 – 4), C_χ is the sum of the squared wavelet family in frequency domain, normalized by s^3. This constant is identical for every frequency $\vec{\omega} \neq \vec{\omega}_0$, but only when all daughter wavelets are available.

In the reconstruction procedure, it is possible to reconstruct the signal $\mathcal{I}(\vec{x})$ from the wavelet coefficients $\mathcal{T}\left(s, \vartheta, \vec{t}\right)$ by using a different wavelet family χ', called the family of *synthesis wavelets* [6, 78]. Thus, Equation (6.1 – 1) can be converted to:

$$\mathcal{I}(\vec{x}) = \frac{1}{C_{\chi,\chi'}} \int_{\mathbb{R}^+} \int_0^{2\pi} \int_{\mathbb{R}^2} \mathcal{T}\left(s, \vartheta, \vec{t}\right)\;\chi'_{s,\vartheta,\vec{t}}(\vec{x})\;\frac{d^2t\,d\vartheta\,ds}{s^3}\,, \tag{6.1-5}$$

with the cross-admissibility condition [6, 78]:

$$0 < C_{\chi,\chi'} = \int_{\mathbb{R}^2\setminus\{\vec{0}\}} \frac{\overline{\chi\left(\vec{\xi}\right)}\,\chi'\left(\vec{\xi}\right)}{\left|\vec{\xi}\right|^2} d^2\xi < \infty\,. \tag{6.1-6}$$

The special family of *dual wavelets* $\chi^d = \frac{\chi}{C_\chi}$ is chosen such that this constant is unity (for the proof of $C_{\chi,\chi^d} = 1$ see Appendix A.2).

6.1.2 Continuous Dual Gabor Wavelet Family

In case of reconstructing from Gabor wavelet responses, the reconstruction formula is similar to Equation (6.1 – 1):

$$\mathcal{I}(\vec{x}) = \frac{1}{C_\psi} \int_0^{2\pi} \int_{\mathbb{R}^+} \int_{\mathbb{R}^2} \mathcal{T}\big(k, \vartheta, \vec{t}\big)\ \psi_{k,\vartheta}\big(\vec{x} - \vec{t}\big)\ \frac{d^2t\, dk\, d\vartheta}{k}\,, \tag{6.1-7}$$

with the constant:

$$C_\psi = C_\psi(\vec{\omega}) = \int_0^{2\pi} \int_{\mathbb{R}^+} \left|\check{\psi}_{k,\vartheta}(\vec{\omega})\right|^2 \frac{dk\, d\vartheta}{k} = \text{const}\,. \tag{6.1-8}$$

In both Equations (6.1 – 7) and (6.1 – 8), the factor $\frac{1}{k}$ arises. This issue can be explained by repeating the calculations from Equation (6.1 – 4) with the Gabor wavelets ψ:

$$\vec{\xi} = \frac{1}{k}\, Q(\vartheta)\, \vec{\omega}\,, \qquad\qquad \frac{d^2\xi}{dk\, d\vartheta} = \frac{\left|\vec{\xi}\right|^2}{k} \tag{6.1-9}$$

(the proof is, again, given in Appendix A.1). When the transition from polar coordinates to Cartesian coordinates:

$$\vec{k} = k\, Q(\vartheta)\ \vec{e}_h\,, \tag{6.1-10}$$

with:

$$\frac{d^2k}{dk\, d\vartheta} = \begin{vmatrix} \frac{\partial k_1}{\partial k} & \frac{\partial k_1}{\partial \vartheta} \\ \frac{\partial k_2}{\partial k} & \frac{\partial k_2}{\partial \vartheta} \end{vmatrix} = \begin{vmatrix} \cos(\vartheta) & -k\, \sin(\vartheta) \\ \sin(\vartheta) & k\, \cos(\vartheta) \end{vmatrix} = k \tag{6.1-11}$$

is made, the calculation of the constant changes to:

$$C_\psi(\vec{\omega}) = \int_{\mathbb{R}^2 \setminus \{\vec{0}\}} \left|\check{\psi}_{\vec{k}}(\vec{\omega})\right|^2 d^2k \tag{6.1-12}$$

and the reconstruction in Equation (6.1 – 7) is rearranged to:

$$\begin{aligned} \mathcal{I}(\vec{x}) &= \frac{1}{C_\psi} \int_{\mathbb{R}^2 \setminus \{\vec{0}\}} \int_{\mathbb{R}^2} \mathcal{T}_{\vec{k}}\big(\vec{t}\big)\ \psi_{\vec{k}}\big(\vec{x} - \vec{t}\big)\ d^2t\, d^2k \\ &= \int_{\mathbb{R}^2 \setminus \{\vec{0}\}} \int_{\mathbb{R}^2} \mathcal{T}_{\vec{k}}\big(\vec{t}\big)\ \psi^d_{\vec{k}}\big(\vec{x} - \vec{t}\big)\ d^2t\, d^2k\,. \end{aligned} \tag{6.1-13}$$

The continuous dual Gabor wavelet ψ^d can be generated identically to the dual wavelets χ^d:

$$\psi^d_{\vec{k}}(\vec{x}) = \frac{\psi_{\vec{k}}(\vec{x})}{C_\psi}\,, \qquad \check{\psi}^d_{\vec{k}}(\vec{\omega}) = \frac{\check{\psi}_{\vec{k}}(\vec{\omega})}{C_\psi}\,. \tag{6.1–14}$$

6.1.3 Discrete Dual Gabor Wavelet Family

When using the discrete Gabor wavelet family Γ (cf. Section 2.2.3), there are, of course, some differences to the continuous reconstruction. The first and most obvious part is that the integral over $\vec{k}$ is converted to the sum over $\vec{k}_j$ ($j = 0, \ldots, J\text{-}1$). Another point is that C_ψ is no longer constant, but a function of $\vec{\omega}$:

$$\check{C}_\psi(\vec{\omega}) = \sum_{j=0}^{J\text{-}1} \left| \check{\psi}_{\vec{k}_j}(\vec{\omega}) \right|^2 . \tag{6.1–15}$$

Recalling that the discrete family of Gabor wavelets covers only a sub-band in half the frequency domain (cf. Figure 2.2 on page 19), some more problems arise. Firstly, the second half of the frequency domain has to be filled, too. This is easily done by using the symmetry condition from Equation (2.2–15), i. e., $\check{\psi}_{\text{-}\vec{k}}(\vec{\omega}) = \check{\psi}_{\vec{k}}(\text{-}\vec{\omega})$ and calculating:

$$\check{C}_\psi(\vec{\omega}) = \sum_{j=0}^{J\text{-}1} \left[\check{\psi}_{\vec{k}_j}(\vec{\omega})^2 + \check{\psi}_{\vec{k}_j}(\text{-}\vec{\omega})^2 \right] . \tag{6.1–16}$$

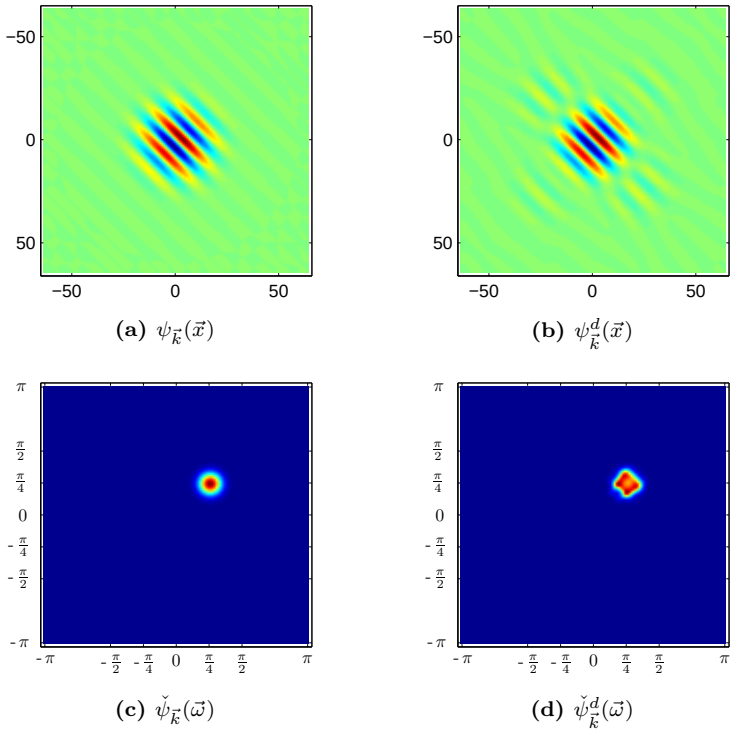
Since $\max_{\vec{\omega}}\ \check{\psi}_{\vec{k}_j}(\vec{\omega}) \approx 1$ and the Gabor wavelet family discretization is sparse enough, the value of C_ψ has a maximum: $\max_{\vec{\omega}}\ \check{C}_\psi(\vec{\omega}) \approx 1$. The second issue is that low and high frequencies are not covered by the discrete Gabor wavelet family and, thus, $\check{C}_\psi(\vec{\omega})$ vanishes at $\vec{\omega} \approx \vec{\omega}_0$ and $|\vec{\omega}| \approx \pi$. Hence, the dual Gabor wavelet would grow infinitely at those frequencies. To solve this, $\check{C}_\psi(\vec{\omega})$ of Equation (6.1–16) is limited below to the constant minimum value[1]:

$$C_{\min} = \frac{1}{4} \approx \frac{1}{4} \max_{\vec{\omega}} \check{C}_\psi(\vec{\omega}) \tag{6.1–17}$$

and, therefore, the dual Gabor wavelet:

$$\check{\psi}^d_{\vec{k}_j}(\vec{\omega}) = \frac{\check{\psi}_{\vec{k}_j}(\vec{\omega})}{\max\{C_{\min}, \check{C}_\psi(\vec{\omega})\}} \tag{6.1–18}$$

[1] The $C_{\min}$ value was adjusted such that the sub-band that is covered by the Gaborwavelets is not altered, but only the high and low frequencies are truncated.



(a) $\psi_{\vec{k}}(\vec{x})$ **(b)** $\psi^d_{\vec{k}}(\vec{x})$

(c) $\check{\psi}_{\vec{k}}(\vec{\omega})$ **(d)** $\check{\psi}^d_{\vec{k}}(\vec{\omega})$

Figure 6.1: Dual Gabor Wavelets: *This figure displays the real parts of (a) a two dimensional Gabor wavelet and (b) its corresponding dual Gabor wavelet in spatial domain, using* $\vec{k} = \left(\frac{\pi}{8}, \frac{\pi}{8}\right)^{\mathrm{T}}$*, as well as a pair of (c) Gabor wavelet and (d) corresponding dual Gabor wavelet in frequency domain, this time with* $\vec{k} = \left(\frac{\pi}{4}, \frac{\pi}{4}\right)^{\mathrm{T}}$*.*

decays similarly to the corresponding Gabor wavelet $\check{\psi}_{\vec{k}_j}$. Two pairs of Gabor wavelet and corresponding dual Gabor wavelet in spatial and in frequency domain are displayed in Figure 6.1. Due to the limitations of $\check{C}_\psi$, the dual Gabor wavelet is only producible in frequency domain. In opposition to the continuous dual Gabor wavelet from Equation (6.1 – 14), an algebraic description of $\psi^d(\vec{x})$ in spatial domain is not available.

6.2 Iterative Reconstruction

Summarizing last section, image $\mathcal{I}$ can be reconstructed from Gabor transformed image $\mathcal{T}$ by:

$$\begin{aligned}\check{\mathcal{I}}(\vec{\omega}) &= \sum_{j=0}^{J-1} \left[\check{\psi}^d_{\vec{k}_j}(\vec{\omega})\ \check{\mathcal{T}}_{\vec{k}_j}(\vec{\omega}) + \check{\psi}^d_{-\vec{k}_j}(\vec{\omega})\ \check{\mathcal{T}}_{-\vec{k}_j}(\vec{\omega}) \right] \\ &= \sum_{j=0}^{J-1} \left[\check{\psi}^d_{\vec{k}_j}(\vec{\omega})\ \check{\mathcal{T}}_{\vec{k}_j}(\vec{\omega}) + \check{\psi}^d_{\vec{k}_j}(\text{-}\vec{\omega})\ \overline{\check{\mathcal{T}}_{\vec{k}_j}(\text{-}\vec{\omega})} \right] ,\end{aligned} \tag{6.2-1}$$

using the two symmetry conditions from Equations (2.2 – 15) and (2.2 – 22).

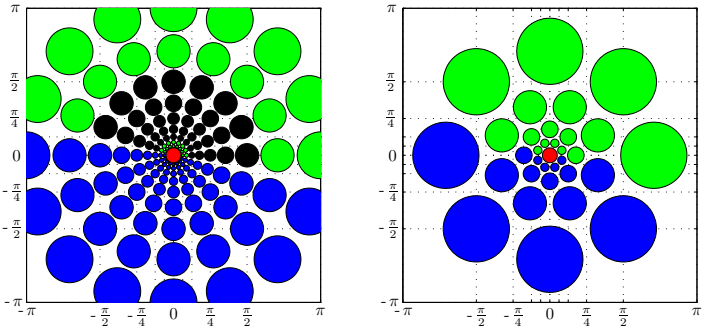
6.2.1 Reconstruction of Gray Level and Color Images

Due to the nature of the Gabor transformed image $\mathcal{T}$, it is not possible to reconstruct a full gray image from $\mathcal{T}$ since the common family of Gabor wavelets Γ cover only a sub-band of the frequency domain. Especially, the low frequencies and specifically the mean gray value are not available. To be able to reconstruct a gray image from $\mathcal{T}$, it must include this information.

To achieve this, the set of Gabor wavelets is extended by changing the parameter set Γ from Equation (2.2 – 16) to the set of *gray level kernels* $\Gamma^{\{g\}} = \left(\nu_{\max}, \zeta^{\{g\}}_{\max}, k^{\{g\}}_{\max}, k_{\text{fac}}, \sigma, \sigma^{\{g\}}_0 \right)$ so that it also contains high and low frequency information:

$$\zeta^{\{g\}}_{\max} = \zeta_{\max} + 4 = 9\,, \quad k^{\{g\}}_{\max} = 2k_{\max} = \pi\,, \quad \sigma^{\{g\}}_0 = \sigma = 2\pi\,. \tag{6.2-2}$$

The additional parameter $\sigma^{\{g\}}_0$ belongs to a Gaussian, called $\check{\psi}_0$ for convenience, that is placed in the center of the frequency domain to cover the mean gray value and the lowest frequencies. Figure 6.2(a) illustrates the frequency *kernels*, i. e., the Gabor wavelets and the Gaussian. Each kernel is depicted by a circle with the center at $\vec{k}$ (or $\vec{0}$ for the Gaussian) and the radius of one (effective) standard deviation. The common set of $\zeta_{\max} = 5$ levels of Gabor wavelets is crayoned in black, the additional two levels of higher and two levels of lower frequency Gabor wavelets are shown in green, and the Gaussian kernel is colored red. Finally, the blue circles in Figure 6.2(a) indicate the second half of the frequency domain, which does not need to be covered with kernels.



(a) Gray Kernels $\Gamma^{\{g\}}$ **(b)** Color Kernels $\Gamma^{\{c\}}$

Figure 6.2: Extended Families of Frequency Kernels: *This figure displays the frequency kernels that are used (a) for gray image layer transforms and (b) for color image layer transforms. The common family Γ of Gabor wavelets is colored black, the additional Gabor wavelets are green and the Gaussian has a red color. The blue colored Gabor wavelets can be omitted in the Gabor wavelet transform.*

The reconstruction itself does not need to be altered much. Of course, the Gabor transformed image $\mathcal{T}$ must have one layer for each kernel. In the reconstruction, the Gaussian kernel needs special attention and, hence, Equation (6.2 – 1) has to be extended to:

$$\check{\mathcal{I}}(\vec{\omega}) = \check{\psi}_0^d(\vec{\omega})\ \check{\mathcal{T}}_0(\vec{\omega}) + \sum_{j=0}^{J^{\{g\}}-1} \left[\check{\psi}_{\vec{k}_j}^d(\vec{\omega})\ \check{\mathcal{T}}_{\vec{k}_j}(\vec{\omega}) + \check{\psi}_{\vec{k}_j}^d(\text{-}\vec{\omega})\ \overline{\check{\mathcal{T}}_{\vec{k}_j}(\text{-}\vec{\omega})} \right] , \quad (6.2-3)$$

where $\check{\psi}_0^d$ depicts the dual Gaussian kernel that is calculated similarly to the dual Gabor wavelets (see Equation (6.1 – 18)) and $J^{\{g\}} = \nu_{\max}\, \zeta_{\max}^{\{g\}} = 72$ is the number of Gabor wavelets generated by $\Gamma^{\{g\}}$.

It is also possible to go one step further and include color information into the Gabor transformed image. With this color information, also colored images can be reconstructed from it. Color information are included using the YUV color space. The Y-plane, i. e., the gray image layer is transformed with the gray level kernels and their generating parameter set $\Gamma^{\{g\}}$ from Figure 6.2(a), while the U- and V-layers are transformed with a different set of kernels, which is shown in Figure 6.2(b) and generated with the parameter

set $\Gamma^{\{c\}} = \left(\nu_{\max}^{\{c\}}, \zeta_{\max}^{\{c\}}, k_{\max}^{\{c\}}, k_{\text{fac}}^{\{c\}}, \sigma^{\{c\}}, \sigma_0^{\{c\}}\right)$:

$$\begin{aligned} \nu_{\max}^{\{c\}} &= 4\,, & k_{\max}^{\{c\}} &= \frac{\pi}{\sqrt{2}}\,, & \sigma^{\{c\}} &= \pi\,, \\ \zeta_{\max}^{\{c\}} &= 4\,, & k_{\text{fac}}^{\{c\}} &= \frac{1}{2}\,, & \sigma_0^{\{c\}} &= 2\pi\,. \end{aligned} \tag{6.2–4}$$

This set includes four levels and four directions of Gabor wavelets, where every second level of Gabor wavelets is shifted by one half distance in angular direction. The Gaussian in the center of the frequency domain is equal to that from the gray kernel set $\Gamma^{\{g\}}$. These settings of transformation and reconstruction kernels are chosen as a compromise between quality and the amount of computed data. The gray image layer, i. e., the Y-layer is, as usual in color image compression, sampled with a higher density, whereas the U- and V-layers are scanned more sparsely. Overall, color images are transformed with 107 gray and color kernels.

The color transform from RGB to YUV color space generates U- and V-layers that scale between values from 0 to 255, where the neutral color, i. e., gray is in the center of this range at value 128. This neutral color value is subtracted from the U and V values before executing the Gabor wavelet transform. The further transformation and reconstruction of the images color layers is identical to the gray image method, besides the different set of kernels. At the end of the reconstruction, the neutral color value 128 is added to the U and V layer pixels and the image layers are color transformed to RGB color space.

6.2.2 Reconstruction from Gabor Graphs

When reconstructing an image from a Gabor graph, the Gabor jets store only the responses of the Gabor wavelets at the node positions $\mathcal{L}_l$ of the graph and, hence, the information for most parts of the Gabor transformed image $\mathcal{T}$ is not available. The iterative reconstruction algorithm proposed in this section, which is depicted in Figure 6.3, tries to spread out the information from the node positions to fill the whole Gabor transformed image.

In the initialization, the Gabor transformed image $\mathcal{T}$ is emptied, i. e., filled with (complex-valued) 0. The first step of the iteration, which is depicted by $\mathcal{T} \stackrel{+}{=} \mathcal{G}$ in Figure 6.3, puts the available information of the Gabor graph $\mathcal{G}$ into the Gabor transformed image $\mathcal{T}$, without changing the other parts of $\mathcal{T}$:

$$\forall l, j\colon \mathcal{T}_{\vec{k}_j}(\mathcal{L}_l) = (\mathcal{J}_l)_j\ . \tag{6.2–5}$$

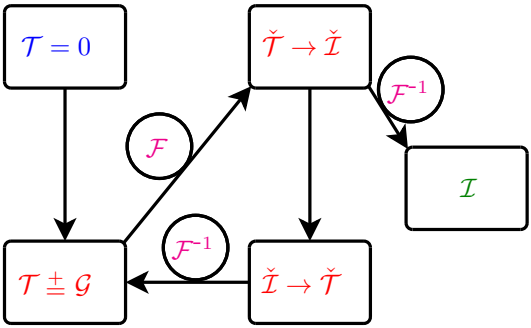
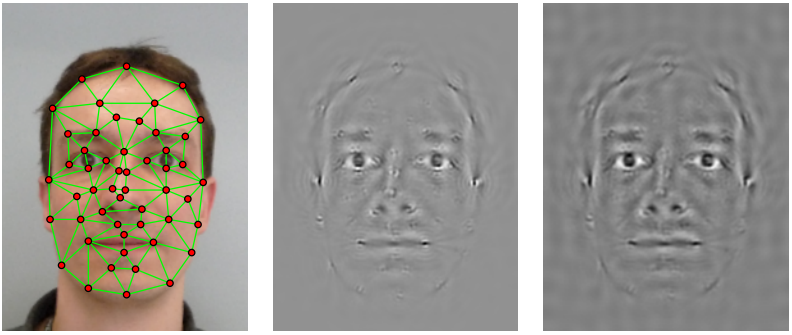


Figure 6.3: Iterative Reconstruction: *This figure displays the steps of the iterative reconstruction algorithm. Starting with an empty Gabor transformed image $\mathcal{T}$, the Graph $\mathcal{G}$ is filled into $\mathcal{T}$ and a pair of reconstruction and Gabor wavelet transform is applied to it. After several iterations, the final image $\mathcal{I}$ is resulting. $\mathcal{F}$ and $\mathcal{F}^{-1}$ indicate Fourier transforms into or from frequency domain, respectively.*

The image $\check{\mathcal{I}}$ is reconstructed in frequency domain ($\check{\mathcal{T}} \to \check{\mathcal{I}}$) using reconstruction Equation (6.2 – 1) or Equation (6.2 – 3). To enable this, the complete Gabor transformed image $\mathcal{T}$ needs to be transformed into frequency domain, i. e., by Fourier transforming each of its layers $\mathcal{T}_{\vec{k}_j}$ separately. Finalizing the iteration cycle, the image in frequency domain is again multiplied with the Gabor wavelets ($\check{\mathcal{I}} \to \check{\mathcal{T}}$), as described in Section 2.2.4, and the resulting Gabor transformed image $\check{\mathcal{T}}$ is transformed back to spatial domain so that the texture information of the graph can be filled in ($\mathcal{T} \stackrel{+}{=} \mathcal{G}$) during the next iteration. Since the Gabor wavelet transformed image can only be filled in spatial domain, but the reconstruction works in frequency domain, $2J$ (inverse) Fourier transforms need to be applied in each iteration. After the last iteration, the image $\check{\mathcal{I}}$ needs another inverse Fourier transform to the final reconstructed image $\mathcal{I}$ in spatial domain.

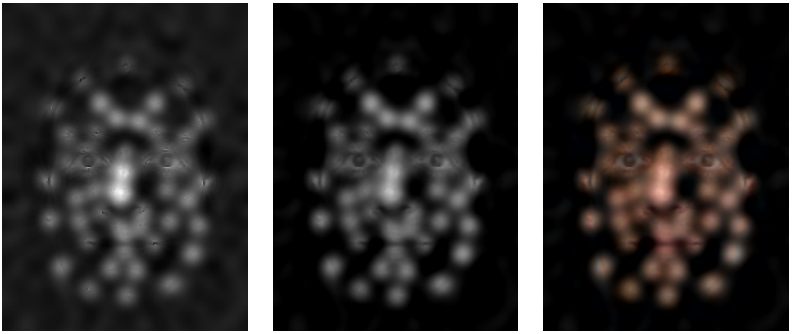
The iterative reconstruction algorithm from Wundrich *et al.* [101], which reconstructs from absolute Gabor wavelet responses, needs additional Fourier transform steps to transform the reconstructed image to spatial domain, cut off the imaginary part of the image, and transform the result back to frequency domain. Since my reconstruction algorithm considers the symmetry characteristics and uses absolute and phase values of the Gabor jets, the image is always real valued and no imaginary part need not to be cut off.



(a) Original Image with Graph

(b) Γ after 0 Iterations

(c) Γ after 200 Iterations



(d) $\Gamma^{\{g\}}$ after 0 Iterations

(e) $\Gamma^{\{g\}}$ after 200 Iterations

(f) $\Gamma^{\{c\}}$ after 200 Iterations

Figure 6.4: Reconstructions from Gabor Graph: *This figure displays reconstructions of the graph shown in (a) with different kernel sets* Γ, $\Gamma^{\{g\}}$, *and* $\Gamma^{\{c\}}$ *after 0 (cf. (b) and (d)) and after 200 iterations (see (c), (e), and (f)). The gray values of the images shown in (b), (c), and (d) are auto-scaled between respective minimum and maximum values.*

Exemplary reconstructions of a face graph are shown in Figure 6.4. The original image including the graph is given in Figure 6.4(a), while Figures 6.4(b) and 6.4(c) display the reconstructed graph including common Gabor jet information after 0 and 200 iterations, respectively. Hence, these images visualize the information that is usually taken for face detection, landmark localization (cf. Chapter 3), identity recognition, and classification (see Chapter 4). The gray values in the images were scaled linearly so that the lowest (usually negative) value was assigned to black, while the highest value became white. The gray level image reconstructions given in Figures 6.4(d) and 6.4(e) were also generated with 0 and 200 iterations, respectively. For the former, the result was again auto-scaled to fit to $[0, 255]$ gray value range. The white spots at the nose in Figure 6.4(d) emerged because the density of the nodes was quite high in that region and, thus, the information spread during the reconstruction step was higher than, e. g., at cheek regions. The absolute gray values after a single reconstruction step is in the order of [-0.5, 2] and, thus, far away from the $[0, 255]$ gray value range from the original image. For Figure 6.4(e), the gray values were not rescaled, but negative values were cut off. Nonetheless, the image looks very blistered. This is also the case for the reconstructed colored image shown in Figure 6.4(f), which employed 200 iterations using the $\Gamma^{\{c\}}$ parameter set for Gabor wavelet transformation and reconstruction. The regions with dense node sampling, e. g., the eyes or the bridge of the nose are reconstructed well, but areas with sparse node sampling are understaffed. Unfortunately, applying more iterations does not solve the problem, so another solution must be found:

6.3 Approximation of the Gabor Transformed Image

The initialization step of the iterative reconstruction algorithm shown in Figure 6.3 is very crude since the layers of the Gabor transformed image $\mathcal{T}$ are initialized with 0 at all unknown positions. Therefore, the amount of information that is spread throughout one reconstruction/transformation cycle is very low. A better initialization $\mathcal{T}^*$ can be attained by interpolating the available information of the graph $\mathcal{G}$, the process is described in this section.

To approximate the value of a certain position $\vec{t}$ in the interpolated Gabor transformed image $\mathcal{T}^*$, where $\vec{t}$ is not located at a landmark position $\mathcal{L}_l$, three surrounding landmarks need to be found and the Gabor jets that are attached to the landmarks have to be interpolated, after the weights for these landmarks are calculated.

6.3.1 Delaunay Triangulation

The most common triangulation algorithm, which is frequently used in 3D computer graphic applications, is the Delaunay triangulation. It generates a tessellation of the area that is optimal in the sense that the circumcircle of each generated triangle does not contain other nodes of the graph. Furthermore, the nodes of the convex hull, which are needed later on, are identified automatically.

I implemented an iteratively growing Delaunay triangulation algorithm. It starts with the two nodes that have the shortest Euclidean distance of all nodes of the Graph, which are called $\mathcal{L}_A$ and $\mathcal{L}_B$ for brevity, and uses them as the first start edge $\overline{\mathcal{L}_A\mathcal{L}_B}$. Iteratively, the algorithm searches for the node $\mathcal{L}_C$ that has the maximum angle:

$$\mathcal{L}_C = \underset{\mathcal{L}\in\{\mathcal{L}_l|l=0,\ldots,L\text{-}1\}\setminus\{\mathcal{L}_A,\mathcal{L}_B\}}{\arg\max} \angle(\overline{\mathcal{L}\mathcal{L}_A}, \overline{\mathcal{L}\mathcal{L}_B})\,, \tag{6.3-1}$$

on both sides of the edge $\overline{\mathcal{L}_A\mathcal{L}_B}$ independently. If only at one side such a node $\mathcal{L}_C$ is found, the edge $\overline{\mathcal{L}_A\mathcal{L}_B}$ is a convex hull edge and, thus, $\mathcal{L}_A$ and $\mathcal{L}_B$ are convex hull nodes. If the found triangle $\triangle\mathcal{L}_A\mathcal{L}_B\mathcal{L}_C$ is new and does not intersect any already existing triangle, edges $\overline{\mathcal{L}_A\mathcal{L}_C}$ and $\overline{\mathcal{L}_B\mathcal{L}_C}$ are put into the set of start edges and the search starts again by using the shortest start edge. The search ends when all start edges are processed.

The resulting triangles tessellate only the area inside of the convex hull of the nodes. Hence, not all positions in $\mathcal{T}$ can be approximated yet and, thus, the remaining area has to be tessellated, too. This is done with an adaptation of the triangulation algorithm that uses only the nodes of the convex hull and crosses the borders of the image boundaries. An exemplary result is displayed in Figure 6.5(a). The green lines show the triangles inside of the convex hull, which in turn is connected with magenta lines. Finally, the blue lines belong to the triangles that cross the image borders.

6.3.2 Limited Linear Weights

The approximation of the Gabor transformed image $\mathcal{T}^*$ from the Gabor graph is done pixel-wise and in spatial domain. To be able to approximate $\mathcal{T}^*(\vec{t})$ from the surrounding landmarks, first there is the need to identify the three nodes surrounding $\vec{t}$, say $\mathcal{L}_A = (A_h, A_v)$, $\mathcal{L}_B = (B_h, B_v)$, and $\mathcal{L}_C = (C_h, C_v)$, which are defined by the Delaunay triangulation. The weights w_A, w_B, and w_C for these landmarks are set up such that they fulfill the linear equation:

$$\vec{t} = w_A\,\mathcal{L}_A + w_B\,\mathcal{L}_B + w_C\,\mathcal{L}_C\,. \tag{6.3-2}$$

Since the weights sum up to unity: $w_A + w_B + w_C = 1$, they can be calculated by analytically solving the system of linear equations:

$$\begin{pmatrix} t_h \\ t_v \\ 1 \end{pmatrix} = \begin{pmatrix} A_h & B_h & C_h \\ A_v & B_v & C_v \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} w_A \\ w_B \\ w_C \end{pmatrix} . \qquad (6.3-3)$$

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. To tackle this issue, the range of the linear interpolation is limited by calculating weights w_j for each Gabor wavelet response[2] independently. The limiting distance is defined by the absolute value of the Gabor wavelet in spatial domain, i. e., the enveloping Gaussian without the prefactor:

$$\left| \tilde{\psi}_{\vec{k}_j}(\vec{x}) \right| = \mathrm{e}^{-\frac{\vec{k}_j^2 \, \vec{x}^2}{2\sigma^2}} . \qquad (6.3-4)$$

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

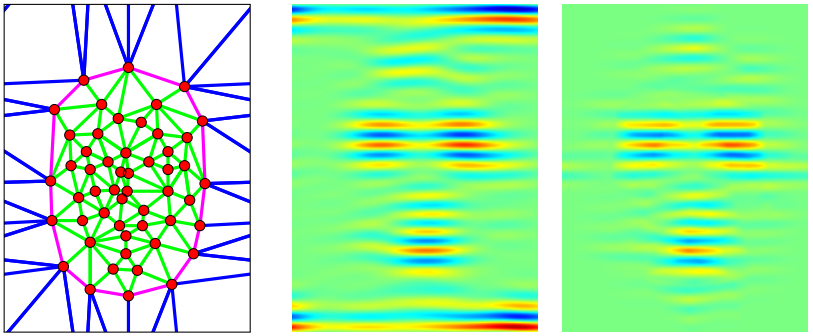
$$w_{X;j} = \left| \tilde{\psi}_{\vec{k}_j}\left(\mathcal{L}_X - \vec{t}\right) \right| \qquad X \in \{A, B, C\} . \qquad (6.3-5)$$

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 to long
3. two distances are short enough, but one is not
4. one distance is short enough, but two are not

where "short enough" means that $|\tilde{\psi}_{\vec{k}_j}(\mathcal{L} - \mathcal{L}')|$ of the two landmarks is dropping below a certain threshold, say 10^{-4}. For possibility 1, the linear weights $w_{A;j} = w_A$, $w_{B;j} = w_B$, and $w_{C;j} = w_C$ are used, where possibility 2 uses weights as given in Equation (6.3–5). The remaining possibilities 3 and 4 are handled by a mixture model that includes both linear and Gaussian weights, which is not explained in detail here.

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



(a) Triangulated Landmarks

(b) Original $\mathcal{T}_{\vec{k}_j}$

(c) Approximated $\mathcal{T}^*_{\vec{k}_j}$

Figure 6.5: Approximation of Gabor Transformed Image: *This figure displays (a) the triangulation result, as well as the real parts of an exemplary layer of (b) the original Gabor transformed image $\mathcal{T}_{\vec{k}}$ and (c) the approximated Gabor transformed image $\mathcal{T}^*_{\vec{k}}$ with $\vec{k} = \left(0, \frac{\pi}{8}\right)^{\mathrm{T}}$.*

6.3.3 Interpolation of Gabor Jets

Finally, the Gabor jets $\mathcal{J}_A$, $\mathcal{J}_B$, and $\mathcal{J}_C$ that are attached to the nodes of the triangle have to be interpolated to calculate the complex pixel value of $\mathcal{T}^*_{\vec{k}_j}\left(\vec{t}\right)$ at the current position $\vec{t}$. To get the correct complex value, the phase of the Gabor jet entry $(\mathcal{J}_X)_j$ $(X = A, B, C)$ has to be shifted according to the disparity between $\vec{t}$ and the node $\mathcal{L}_X$:

$$\left(\mathcal{J}'_X\right)_j = \left(\mathcal{J}_X\right)_j \, \mathrm{e}^{i\,\vec{k}_j^{\mathrm{T}}\left(\vec{t}-\mathcal{L}_X\right)} \,. \tag{6.3-6}$$

These phase shifted Gabor jet elements are summed up weightedly by means of Equation (6.3 – 7) using the weights that have been calculated in the last section. Again, real and imaginary parts of the complex values are treated independently:

$$\begin{aligned} \mathcal{T}^*_{\vec{k}_j}\left(\vec{t}\right) &= \sum_{X\in\{A,B,C\}} w_{X;j}\left(\mathcal{J}'_X\right)_j \\ &= \sum_{X\in\{A,B,C\}} w_{X;j}\left(\mathcal{J}_X\right)_j \, \mathrm{e}^{i\,\vec{k}_j^{\mathrm{T}}\left(\vec{t}-\mathcal{L}_X\right)} \,. \end{aligned} \tag{6.3-7}$$

Figure 6.5 shows an example of the original layer $\mathcal{T}_{\vec{k}_j}$, i. e., the result of the Gabor wavelet transform (see Chapter 2 for details) with Gabor wavelet $\psi_{\vec{k}_j}$, and the approximated layer $\mathcal{T}^*_{\vec{k}_j}$ generated by this algorithm, both using $\vec{k}_j = \left(0, \frac{\pi}{8}\right)^{\mathrm{T}}$. The high responses at top and bottom border of Figure 6.5(b) are not (and need not be) estimated properly since the nodes of the graph are too far away from this background region and, thus, the Gabor jets of the graph do not contain that information.

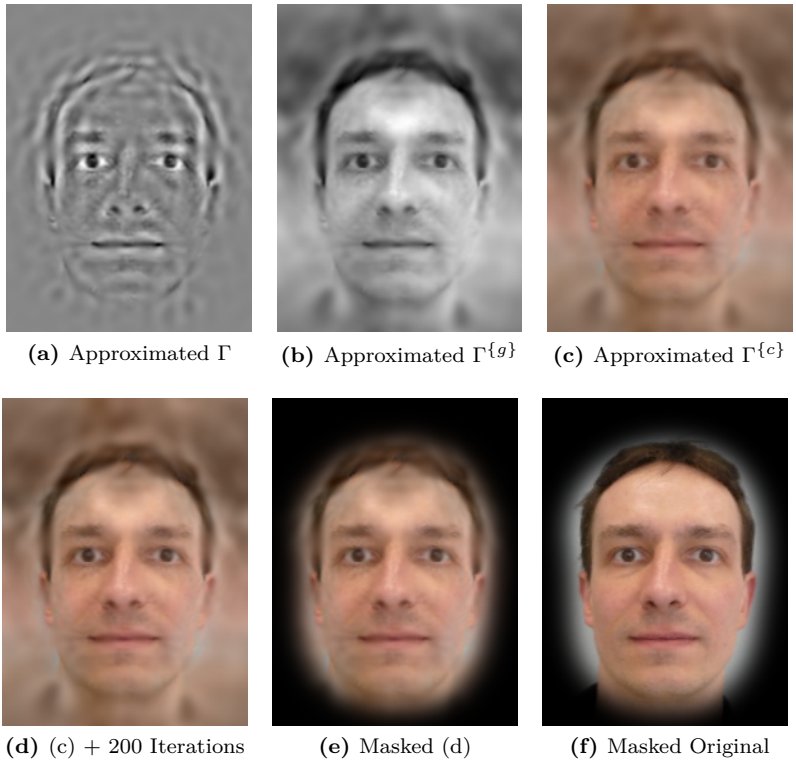
6.4 Background Removal

Exemplary reconstructions of approximated Gabor transformed images can be found in Figure 6.6. The first row shows the reconstructions employing the three different sets of Gabor kernels, i. e., $\Gamma, \Gamma^{\{g\}}$, and $\Gamma^{\{c\}}$, respectively, using no iterations. Clearly, all three images look much better than the corresponding images of Figure 6.4 that were generated with 200 iterations. Nonetheless, combining both approaches results in images that have a higher quality in terms of sharpness, an example is given in Figure 6.6(d).

The optional last step of the image reconstruction is background removal, where *background* is defined as everything outside of the convex hull of the graph. These parts of the image usually do not show reasonable information, but look *cloudy*, especially if the background is far away from the face. To remove this background, the convex hull nodes of the graph are used, which are provided during triangulation in Section 6.3.1. Connecting the convex hull directly would lead to straight lines, the resulting image would look unhandsomely. For this reason, a cubic B-Spline $\mathcal{V}$, an implementation of which was already available in the institute, is put through the convex hull nodes and the pixel values that are outside of $\mathcal{V}$ are smoothed out. Therefore, the resulting image is weighted with a mask image $W_{\mathcal{V}}$:

$$W_{\mathcal{V}}(\vec{x}) = \begin{cases} 1 & \text{if } \vec{x} \text{ inside } \mathcal{V} \\ \mathrm{e}^{-\frac{D(\vec{x},\mathcal{V})}{2\sigma_{\mathcal{V}}^2}} & \text{otherwise} \end{cases}, \qquad (6.4-1)$$

where $D(\vec{x}, \mathcal{V})$ is the distance from $\vec{x}$ to the nearest point on the B-Spline $\mathcal{V}$ and $\sigma_{\mathcal{V}} = 4\pi$ is twice the σ of the Gaussian kernel. The result of the background removal step is shown in Figure 6.6(e). For comparison, the original image as shown in Figure 6.4(a) masked with $W_{\mathcal{V}}$ is displayed in Figure 6.6(f). Clearly, the forehead area is not reconstructed perfectly, but the person shown in Figure 6.6(e) is clearly identifiable as me.



(a) Approximated Γ **(b)** Approximated $\Gamma^{\{g\}}$ **(c)** Approximated $\Gamma^{\{c\}}$

(d) (c) + 200 Iterations **(e)** Masked (d) **(f)** Masked Original

Figure 6.6: Reconstruction of Approximated Gabor Transformed Image: *This figure shows image reconstructions of approximated Gabor transformed images $\mathcal{T}^*$ employing the different Gabor kernel sets Γ, $\Gamma^{\{g\}}$, and $\Gamma^{\{c\}}$. In (a), (b), and (c) no additional reconstruction iterations were used, whereas (d) was generated using both $\mathcal{T}^*$ and additional 200 iterations. In (e), the image from (d) was masked with mask $W_{\mathcal{V}}$. The original image masked with mask $W_{\mathcal{V}}$ is given in (f).*

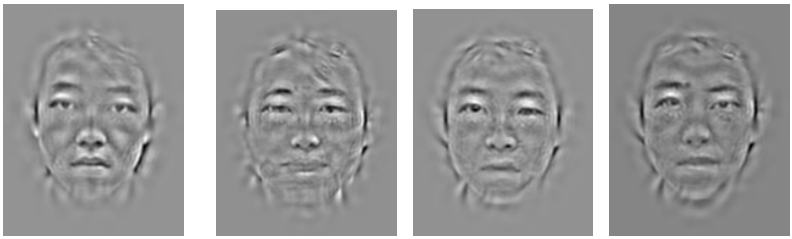
6.5 Time Performance

Finally, the time consumption of the reconstruction is assessed. To give a rough estimation of the reconstruction duration, the times that were needed to generate the images shown in Figures 6.4 and 6.6 were recorded. The produced images have the resolution 192×256 pixel and the graph to be reconstructed has 52 nodes. The computer that is used for this tasks is a Dell Precision 670 with a 3200 MHz Intel Xeon (32 bit) dual-core processor. The application is written in C++, and the implementation of the Intel® *math kernel library* (MKL) is employed for Fourier transform.

The consumed time is very different between the different approaches. The iterative reconstruction of a common Gabor graph (using parameter set Γ) with 200 iterations took 1 minute and 51 seconds to compute, which were split up into 3 seconds for Gabor wavelet initialization, and 14 seconds or 7 seconds for the forward Gabor wavelet transform or the reconstruction steps, respectively. The most expensive parts of the iteration are the Fourier transforms of the Gabor transformed image, which altogether needed 1 minute and 24 seconds. The overall times raised to 3 minutes and 22 seconds or 4 minutes and 59 seconds when employing $\Gamma^{\{g\}}$ or $\Gamma^{\{c\}}$, respectively, where again the Fourier transforms took the biggest parts.

In opposition, the time needed to approximate the Gabor transformed image amounts to 6 seconds, 12 seconds, and 13 seconds for Γ, $\Gamma^{\{g\}}$, and $\Gamma^{\{c\}}$, respectively. Hence, the results shown in the first row of Figure 6.6 look much better *and* can be generated much faster than the results of the iterative reconstruction algorithm from Figure 6.4. The reason for that is surely the non-necessity of Fourier transforms for approximating $\mathcal{T}^*$.

For the final background removal step, the mask $W_{\mathcal{V}}$ including the calculation of the B-Spline $\mathcal{V}$ took 9 seconds to be computed and applied. Overall, the image shown in Figure 6.6(e) needed 5 minutes and 18 seconds to be generated. Of course, for real-time applications this time period is unacceptable, but in those cases it might be sufficient to skip the 200 iterations and take the images from Figure 6.6(c), which took "only" 14 seconds.



(a) Detected Graph **(b)** Global Move **(c)** Scan Local Move **(d)** Disparity Scan Local Move

Figure 6.7: Phantom Faces: *This figure displays phantom faces for different stages of the EBGM detection schedule: (b) the global move employing $S_{[A]}$, (c) the scan local move employing $S_{[P]}$, and (d) the disparity scan local move employing $S_{[D]}$. For comparison, the graph with Gabor jets of the detected person is visualized in (a).*

6.6 Reconstruction Examples

To show that the reconstruction works well for face graphs that are not directly extracted from an image, this section presents two other applications of the reconstruction.

6.6.1 Phantom Faces

Phantom faces visualize the detection process in the EBGM schedule. They are reconstructed from *phantom graphs* $\mathcal{G}^{\#}$ that hold the most similar Gabor jet $\mathcal{J}_l^{\#}$ of the bunch $\mathcal{J}_l^{\mathcal{B}}$ for each landmark:

$$\mathcal{J}_l^{\#} = \underset{\mathcal{J}_l{}^{(b)} \in \mathcal{J}_l^{\mathcal{B}}}{\arg\max} \; S_{[\cdot]}\Big(\mathcal{J}_l{}^{(b)}, \mathcal{J}_l^{\mathcal{I}}\Big) \;, \tag{6.6-1}$$

where $\mathcal{J}_l^{\mathcal{I}}$ is the Gabor jet of the currently investigated image graph $\mathcal{G}^{\mathcal{I}}$. Thus, the phantom graph $\mathcal{G}^{\#}$ is a conglomerate of Gabor jets from the bunch graph $\mathcal{G}^{\mathcal{B}}$, i. e., a mixture of texture information from different people.

Figure 6.7 displays phantom faces for different stages of the EBGM detection schedule. For comparison, Figure 6.7(a) shows the reconstruction of the image graph that was extracted from the probe image, i. e., the image graph shown in Figure 3.2(b). The bunch graph $\mathcal{G}^{\mathcal{B}}$ integrated Gabor jets from hand-labeled face graphs of 18 people, not including the identity shown

in the probe image. The phantom face that is shown in Figure 6.7(b) was generated after the global move. The Gabor jets in this phantom graph are those with the highest $S_{[A]}$ similarities. Since this similarity function does not respect phases of the Gabor jets and the node positions are not yet aligned to the landmarks, the reconstructed image has some distortions. Figures 6.7(c) and 6.7(d) show phantom faces after scan local move and disparity scan local move employing $S_{[P]}$ and $S_{[D]}$, respectively. In comparison, both phantom faces include facial features from different persons, e. g., the nose differs between the two images, and both are different from Figure 6.7(a).

The visualizations in Figure 6.7 show that the reconstruction algorithm is able to reconstruct graphs incorporating facial features of different persons. Possibly, this reconstruction algorithm could be used for constructing facial composite images integrating facial parts of different people. As a practical application, these composite images could help the police to capture criminals. In this case, composite images, of course, should be reconstructed using kernel set $\Gamma^{\{g\}}$ or $\Gamma^{\{c\}}$.

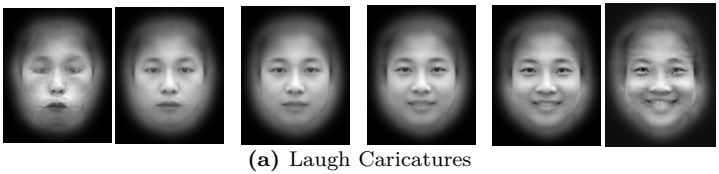
6.6.2 Caricatures

Caricatures are facial images with exaggerated facial properties. *Caricature graphs* $\mathcal{G}^\circ$ are face graphs with exaggerated texture and geometrical features. They are created using a base graph $\mathcal{G}$ and a target graph $\mathcal{G}'$. Usually, the target graph contains the properties that should be caricatured, while the base graph is an average of several of these properties. Caricatures can be created in different magnitudes, defined by weight w for both graphs. They are computed by adding w times the difference of base and target graph to the base graph:

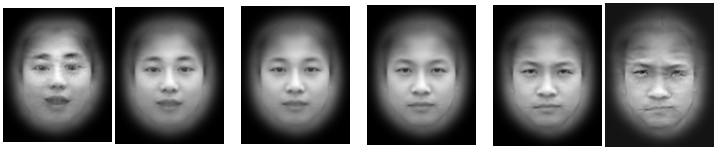
$$\mathcal{G}^\circ_w = \mathcal{G} + w\left(\mathcal{G}' - \mathcal{G}\right) . \qquad (6.6-2)$$

Caricatures graphs are built by modifying both the landmark positions and the texture in terms of Gabor jets, where the Gabor wavelet responses are, in turn, averaged in algebraic form.

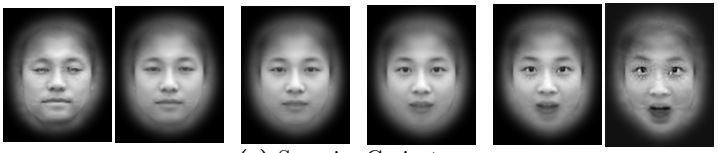
Figure 6.8 shows facial expression caricatures, the images were reconstructed from face graphs using the $\Gamma^{\{g\}}$ kernel set. Since the geometrical structure of the graphs is altered, the sizes of the reconstructed images vary. The base graph $\mathcal{G}$, which is shown in the third column, is generated by averaging 2261 semi-automatically detected face graphs showing six different facial expressions, i. e., the same set that was used for expression classification in Section 5.4.1. The target graphs $\mathcal{G}'$, which hold averaged graphs for each of the six expressions, are visualized in the fourth column. The last



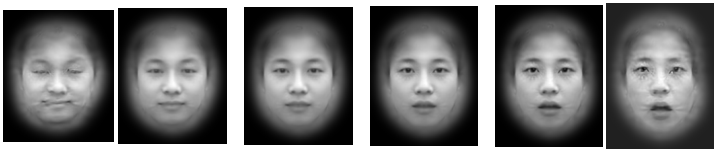
(a) Laugh Caricatures



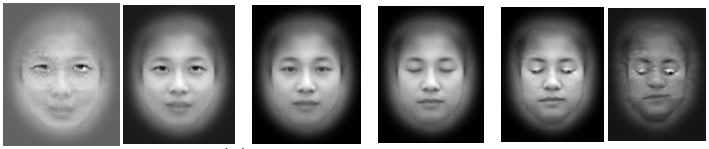
(b) Frown Caricatures



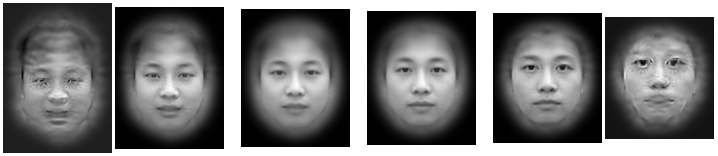
(c) Surprise Caricatures



(d) Open Mouth Caricatures



(e) Closed Eye Caricatures



(f) Neutral Caricatures

Figure 6.8: Facial Expression Caricatures: *This figure displays caricatures and anti-caricatures for averaged facial expressions. The weights for the caricatures are (from left to right):* w = -3, w = -1, $w = 0$ *(base graph* $\mathcal{G}$*),* $w = 1$ *(target graph* $\mathcal{G}'$*),* $w = 2$ *and* $w = 4$.

two columns show caricatures of the expressions, with weights $w = 2$ and $w = 4$, respectively, while the first two columns show *anti-caricatures* with weights w = -3 and w = -1, respectively. Note that weight w = -1 for the anti-caricature is equivalent to weight $w = 2$ for the caricature, cf. Equation (6.6 – 2). The Laugh, Frown, Surprise, and Open mouth caricatures are clearly showing these expressions, and for $w = 2$ also the Closed eye and the Neutral expression caricatures are identifiable. Facial expression anti-caricatures emphasize, how facial expressions do not look like. For example, it is unlikely to have the mouth closed while laughing or being surprised and, hence, the Laugh and Surprise anti-caricatures have closed mouths and closed eyes. In opposition, the anti-caricature of the Closed eyes expression emphasizes opened eyes. Notably, the anti-caricature of the Laugh expression looks sad, although "sad" is not part of the expressions in the CAS-PEAL database.

A possible use case for caricatures is the reconstruction of caricatured genetic syndromes (see Section 5.4.3 for exemplary averaged syndrome graphs) that emphasize the characteristics of the specific syndrome. These images could be used to train medical doctors to diagnose syndromes and to discriminate between them.

Chapter 7

Summary

7.1 Conclusion

This thesis introduces a simple parameter-free statistical model, which is able to learn the statistics of any data vectors from few training examples with a time complexity linear in the number of data vectors. It is shown, how this model can be used for face detection, exact feature point localization, face recognition, and facial property classification. The elements of the data vectors are expected to be linearly independent and approximately Gaussian distributed, knowing that these expectations are not met in most of cases. Nonetheless, few training examples are not sufficient to reliably compute transformation matrices that generate linearly independent data vectors. Assuming linear independence has another advantage: the elements in the vector do not need to be comparable in size or distribution. Hence, it is easily possible to integrate data vectors of different kinds, e. g., texture and geometrical information in the case of the face graph, especially since the model generates dimensionless data.

In opposition to many common face detection systems, in the face detection task only positive training examples are needed and no definition of "non-faces" is required. As an exemplary application, the *elastic bunch graph matching* (EBGM) algorithm [96] is modified using the proposed model. It is shown that this model is able to learn the distribution of texture and geometry of the training faces and to detect novel faces reliably. Furthermore, a fast method of transforming the texture descriptors is added to be used in multi-scale and multi-angle face detection. It is empirically demonstrated that both extensions of the EBGM face detection work well and that faces in different sizes and in-plane rotation angles are found dependably. Unfortunately, in the combination of the multi-scale and multi-angle face detection with the model, it showed performance slightly below the combination with the EBGM approach. Since the model is trained on proper texture features, it may have issues classifying interpolated texture descriptors. During feature point localization, statistics of texture and geometrical information are integrated to allow texture information to be divergent, while keeping the geometrical configuration stable. An iterative landmark position refining al-

gorithm is proposed that allows facial features to be aligned locally with higher accuracy. The detected node positions are accurate enough to help classification algorithms, although they are not as precise as hand-labeled positions.

For face recognition, the proposed model is extended to a classifier that learns how to compare two faces by choosing appropriate training image pairs. For the comparison of a gallery and a probe image, this classifier estimates the probability that these two facial images are from the same person. The advantage of using this approach is that the classifier learns these comparisons independent of the persons. Unlike other classification methods, the persons in the gallery do not need to be in the training set and, thus, the proposed classifier is able to recognize faces that it has not seen during training. By applying this classifier to face graphs, it is shown that it can learn from few training data to dependably recognize faces in different sizes and with different facial expressions. Recognition rates drop remarkably when the illumination conditions change. One important attainment of this thesis is the realization that those parts of the texture descriptors that are usually ignored in face recognition, i. e., the Gabor phases are indeed well suited for identification in case of illumination variations. Although Zhang *et al.* [111] also lately used Gabor phases to build *enhanced local Gabor binary pattern* texture descriptors, together with Haufe [31] I propagate a function that allows to use the direct comparison of Gabor phases for recognition. Additionally, when ground truth node positions are available, it is shown that the geometrical information is also useful for recognition and the combination of texture and geometrical information increases recognizability. When applied to large scale databases, the classifier is able to increase recognition accuracies moderately.

The very same classifier that is used for face recognition is employed for the classification of facial image properties, like facial expressions or illumination conditions. This is achieved by exchanging the training image pairs. Since the classifier is parameter-free, no further adaptations need to be done. Using a face graph to classify facial expression, it is demonstrated that the integration of the geometrical information of automatically detected feature positions can improve texture-based classification accuracy. Classification of lighting condition proves to be reliable, and using the estimated lighting condition, recognition accuracy can be improved. In this case, recognition is even robust against misclassification of the lighting condition. The most important success of the proposed model is the classification of genetic syndromes based on hand-labeled facial images of the patients. Without the need of setting up any parameters, this model outperforms current classification algorithms that have many hand-crafted parameters and are highly

optimized to the syndrome classification task. Additionally, it is argued that the texture of automatically detected faces can be used for syndrome classification as well, but the landmark positions show not to be precise enough to enhance classification accuracy. In another experiment [79], we indicate that this classifier is even able to exceed human expert performance in classifying acromegaly versus a control group based on hand-labeled facial images.

In all recognition and classification experiments performed in this thesis, the normalized scalar product texture comparison function, which is commonly used for face detection and face recognition, does not perform best. In most of the simple cases, i. e., when comparing images with identical illumination conditions, the Canberra and the modified Manhattan texture similarity functions outperform the normalized scalar product (see also [36]). In more challenging recognition tasks, the disparity similarity texture comparison function, which includes Gabor phases and which is usually not used for identification turn out to be very valuable. After optimizing this comparison function, in the sense that positioning errors, which occur during landmark localization, are canceled out more reliably, it operates even better on face recognition and facial property classification.

Finally, the reconstruction of images from face graphs is solved by integrating a solid theoretical foundation with well-known computer graphics algorithms. One innovation is the interpolation of texture descriptors, limiting highly localized parts of the descriptors to local regions. As an extension, different kinds of texture descriptors are proposed, e. g., one that includes color information. It is shown that image reconstructions can not only be computed from face graphs that were extracted from one image, but also mixtures of face graphs, e. g., combinations of facial parts of different people or averages of face graphs show to be soundly reconstructible. Notably, reconstructions of averaged face graphs of genetic syndromes clearly include the prominent characteristics of these syndromes [7].

7.2 Outlook

Unfortunately, the time for this thesis is limited, while I still have tons of ideas that might further improve detection, recognition, or classification accuracy. In this thesis, the statistical model is applied to face graphs with Gabor jets as texture descriptors, but it is able to deal with any kind of image descriptors. For example, *scale invariant feature transform* (SIFT) features are used to reliably locate texture positions in different scales and angles and, hence, replacing the Gabor jets by SIFT features in the EBGM algorithm might be a good idea. For face recognition, Müller [56] already ex-

ecuted some experiments with local binary patterns, extending them to the *graph based local Gabor binary pattern histogram sequence* (GBLGBPHS). It seemed to be expedient and performed better than the *local Gabor binary pattern histogram sequence*, which is based on regular grid node positions. The classifier that is proposed in this thesis is flexible enough to use comparisons of these histograms as data vectors. It would be interesting to try if GBLGBPHS feature comparisons can be learned.

Face detection and landmark localization under different poses, facial expressions, or illuminations is still an issue. The *flexible object model*, which Tewes [87] introduced, allows more specific node positioning tests. In combination with the proposed statistical model, it could solve the problem. For face recognition, Tewes proposed to use the estimated expression or pose to transform texture information from expressions or poses into the neutral images. Beyond that, it should also be possible to use my classifier to learn the distributions of the texture under each expression or pose. In case of expression, the person could be recognized by simple graph comparison, the proposed classifier already showed stability against facial expressions. In case of poses, face graphs are no longer comparable since the geometry of the graphs may have changed and some landmarks are invisible. Furthermore, the texture of the face has completely changed due to head rotation. The model-based ranking list solution to that problem that was introduced by Müller [58, 57, 56] is already very stable, but both his graph comparison function and his rank list comparison function are untrained. Maybe statistical learning of these functions might further improve his already outstanding recognition results.

In this thesis as well as in literature [96], face detection (including scale and angle estimation) is done with a Gabor jet comparison function that ignores the phases of the Gabor wavelet responses. Usually, this was a good choice since the phases were unusable when the nodes of the graph do not perfectly hit the landmarks, which is never the case during face detection. In this thesis, I introduce a procedure that greatly enhances the estimation of node positioning errors based on the Gabor jets. Hence, the disparity similarity texture comparison function that uses the phases of the Gabor wavelet responses after displacement correction could increase face detection accuracy, although it would slow down the detection process remarkably.

The nodes of the face graphs are located at positions corresponding to facial landmarks. Although detection and recognition accuracies are good, it is completely unclear whether these positions are really best suited for these tasks. In my Diploma thesis [28], I conducted experiments applying evolutionary algorithms to find out which node positions are useful for face detection and recognition and which parts of the texture descriptors, i. e., the

responses of which directions and scales of Gabor wavelets are important. I showed that the data required for reliable facial feature detection is not identical to the data that allows more robust face recognition. These results can be explained by the fact that these two applications are somewhat opposite: while facial feature detection exploit consistencies between faces, recognition needs to discriminate between different faces. Furthermore, it turned out that the node positions that are useful for face recognition are highly dependent on the dataset, e. g., the eye centers proved to be nearly useless in face recognition under different facial expressions, while the eyebrow region is always valuable [28]. It is still unclear whether the nodes in the margin of the face should be included, or if the influence of the background makes them void and only the inner facial features are utilizable.

There is another ongoing discussion into that direction. Currently, two main opinions about general node placement exists: one of them is believing that grid graphs and regular positions are better suited for recognition, while the other one thinks that face graphs with very precise landmark positions are more useful. Current findings [83, 31] suggest that the former opinion may be the right one. In combination with the classifier proposed in this thesis, the grid graph can outperform the face graph [31] in terms of recognition accuracy, even when the number of nodes in the grid is lower than in the face graph. Still, face detection with grid bunch graphs is inferior to face detection with face bunch graphs [83].

The Gabor jet texture descriptor I am using is said to be motivated by the visual path in humans, but this is only partially true. The current local size, i. e., the σ of the Gabor wavelet was taken to be 2π, while the pictures presented by Daugman [14] and Jones and Palmer [37] rather vote for a much smaller value like $\sigma = 2$, which was, e. g., used in my diploma thesis [28]. Following from that, the complete parametrization of the Gabor wavelet family should be reviewed. Another issue is the normalization prefactor of the Gabor wavelets. Latest investigations indicate that the currently used prefactor indeed performs well on natural images, but might be inappropriate for facial images.

Throughout my thesis, I use normalized texture descriptors for detection, recognition, and classification because (after some experiments that are not reported in this thesis) I strongly believe that unnormalized Gabor jets do not work that well, even when illumination conditions are controlled. Nonetheless, I might be wrong in that belief and further test need to be invoked. Still, misdetections in the background are usually provoked by boosting texture noise by texture normalization. It might be a good choice to apply the normalization only when the normalization factor exceeds a certain threshold. Potentially, this normalization factor could also be given as one input for

the proposed statistical model so that no fixed threshold has to be assessed.

The next step should test the proposed face detection, face recognition, and facial property classification systems under real-life conditions, e. g., on an interactive shopping guide robot [26]. In the investigated databases, all conditions like facial expressions, poses, and illuminations are varied individually, but on the robot all these conditions appear concurrently. Fortunately, the number of identities the robot needs to recognize is usually low, most often it should be sufficient to verify that it is still talking to the same person. Furthermore, instead of having single static images, the mobile camera supplies video data that allows for stabilizing identification and classification decisions over several frames, requiring the invention of algorithms to cope with that. But, as the results of this thesis indicate, classifying facial expressions is a big challenge.

Appendix A

Proofs

A.1 Proof of Coordinate Transformation

2D Wavelets

For the 2D wavelets admissibility constant C_χ, the following coordinate transformations must be made:

$$\vec{\xi} = s \begin{pmatrix} \omega_h \cos(\vartheta) - \omega_v \sin(\vartheta) \\ \omega_h \sin(\vartheta) + \omega_v \cos(\vartheta) \end{pmatrix} .$$

Thus, $d^2\xi$ in Equation (6.1 – 2) has to be substituted by $ds\, d\vartheta$. Therefore, the determinant of the Jacobean matrix is needed:

$$\begin{aligned}
\frac{d^2\xi}{ds\, d\vartheta} &= \begin{vmatrix} \frac{\partial \xi_h}{\partial s} & \frac{\partial \xi_h}{\partial \vartheta} \\ \frac{\partial \xi_v}{\partial s} & \frac{\partial \xi_v}{\partial \vartheta} \end{vmatrix} \\
&= \begin{vmatrix} \omega_h \cos(\vartheta) - \omega_v \sin(\vartheta) & s\,(\text{-}\omega_h \sin(\vartheta) - \omega_v \cos(\vartheta)) \\ \omega_h \sin(\vartheta) + \omega_v \cos(\vartheta) & s\,(\omega_h \cos(\vartheta) - \omega_v \sin(\vartheta)) \end{vmatrix} \\
&= |(\omega_h \cos(\vartheta) - \omega_v \sin(\vartheta))\, s\,(\omega_h \cos(\vartheta) - \omega_v \sin(\vartheta)) - \\
&\quad (\omega_h \sin(\vartheta) + \omega_v \cos(\vartheta))\, s\,(\text{-}\omega_h \sin(\vartheta) - \omega_v \cos(\vartheta))| \\
&= |s\,(\omega_h^2 \cos^2(\vartheta) + \omega_v^2 \sin^2(\vartheta) - 2\,\omega_h\,\omega_v \sin(\vartheta) \cos(\vartheta)) - \\
&\quad s\,(\text{-}\omega_h^2 \sin^2(\vartheta) - \omega_v^2 \cos^2 \vartheta - 2\omega_h\,\omega_v \sin(\vartheta) \cos(\vartheta))| \\
&= |s\,(\omega_h^2\,(\cos^2(\vartheta) + \sin^2(\vartheta)) + \omega_v^2\,(\cos^2(\vartheta) + \sin^2(\vartheta)))| \\
&= s\,|\vec{\omega}|^2 .
\end{aligned}$$

With $|\vec{\omega}| = \frac{1}{s}|\vec{\xi}|$ (cf. Equation (6.1 – 3)) it follows that $\frac{d^2\xi}{ds\, d\vartheta} = s\frac{|\vec{\xi}|^2}{s^2} = \frac{|\vec{\xi}|^2}{s}$.

Gabor Wavelets

For the Gabor wavelets admissibility constant C_ψ in Equation (6.1 – 8), the coordinate transformation is similar, but the prefactor changes from s to $1/k$:

$$\vec{\xi} = \frac{1}{k} \begin{pmatrix} \omega_h \cos(\vartheta) - \omega_v \sin(\vartheta) \\ \omega_h \sin(\vartheta) + \omega_v \cos(\vartheta) \; . \end{pmatrix}$$

The calculation of the determinant of the Jacobean matrix:

$$\begin{aligned}
\frac{d^2\xi}{dk\, d\vartheta} &= \begin{vmatrix} \frac{\partial \xi_h}{\partial k} & \frac{\partial \xi_h}{\partial \vartheta} \\ \frac{\partial \xi_v}{\partial k} & \frac{\partial \xi_v}{\partial \vartheta} \end{vmatrix} \\
&= \begin{vmatrix} -\frac{1}{k^2} \left(\omega_h \cos(\vartheta) - \omega_v \sin(\vartheta)\right) & \frac{1}{k} \left(-\omega_h \sin(\vartheta) - \omega_v \cos(\vartheta)\right) \\ -\frac{1}{k^2} \left(\omega_h \sin(\vartheta) + \omega_v \cos(\vartheta)\right) & \frac{1}{k} \left(\omega_h \cos(\vartheta) - \omega_v \sin(\vartheta)\right) \end{vmatrix} \\
&= \left| -\frac{1}{k^3} \Big[\left(\omega_h \cos(\vartheta) - \omega_v \sin(\vartheta)\right) \left(\omega_h \cos(\vartheta) - \omega_v \sin(\vartheta)\right) - \right. \\
&\qquad \left. \left(\omega_h \sin(\vartheta) + \omega_v \cos(\vartheta)\right) \left(-\omega_h \sin(\vartheta) - \omega_v \cos(\vartheta)\right) \Big] \right| \\
&= \left| -\frac{1}{k^3} \Big[\left(\omega_h^2 \cos^2(\vartheta) + \omega_v^2 \sin^2(\vartheta) - 2\,\omega_h\,\omega_v \sin(\vartheta) \cos(\vartheta)\right) - \right. \\
&\qquad \left. \left(-\omega_h^2 \sin^2(\vartheta) - \omega_v^2 \cos^2(\vartheta) - 2\,\omega_h\,\omega_v \sin(\vartheta) \cos(\vartheta)\right) \Big] \right| \\
&= \left| -\frac{1}{k^3} \left(\omega_h^2 \left(\cos^2(\vartheta) + \sin^2(\vartheta)\right) + \omega_v^2 \left(\cos^2(\vartheta) + \sin^2(\vartheta)\right)\right) \right| \\
&= \frac{|\vec{\omega}|^2}{k^3}
\end{aligned}$$

is slightly different. With $|\vec{\omega}| = k\,|\vec{\xi}|$ (cf. Equation (6.1 – 9)) it follows that $\frac{d^2\xi}{dk\, d\vartheta} = \frac{k^2\,|\vec{\xi}|^2}{k^3} = \frac{|\vec{\xi}|^2}{k}$.

A.2 Proof of Cross-Admissibility-Condition

To be able to use a different synthesis wavelet χ' for reconstruction than the wavelet χ that was employed for the wavelet transform, the cross-admissibility condition $0 < C_{\chi,\chi'} < \infty$ from Equation (6.1 – 6) has to be fulfilled.

The most simple synthesis wavelet is the dual wavelet $\chi^d = \frac{\chi}{C_\chi}$ that removes the prefactor $\frac{1}{C_{\chi,\chi'}}$ of the reconstruction Equation (6.1 – 5). Therefore, C_{χ,χ^d} must be unity:

$$\begin{aligned} C_{\chi,\chi^d} &= \int\limits_{\mathbb{R}^2\setminus\{\vec{0}\}} \frac{\overline{\chi\left(\vec{\xi}\right)}\,\chi^d\left(\vec{\xi}\right)}{|\vec{\xi}|^2} d^2\xi \\ &= \int\limits_{\mathbb{R}^2\setminus\{\vec{0}\}} \frac{\overline{\frac{\chi(\vec{\xi})}{C_\chi}}\,\chi(\vec{\xi})}{|\vec{\xi}|^2} d^2\xi \\ &= \frac{1}{C_\chi} \int\limits_{\mathbb{R}^2\setminus\{\vec{0}\}} \frac{\left|\chi\left(\vec{\xi}\right)\right|^2}{|\vec{\xi}|^2} d^2\xi \\ &= \frac{C_\chi}{C_\chi}\,. \end{aligned}$$

A.3 Proof of Rotation Direction Change

The calculation of the daughter Gabor wavelets can be done in several ways. The difference between Equations (2.2 – 5) and (2.2 – 11) is that in the former, the position $\vec{x}$ in spatial domain is rotated clockwise, while in the latter, the parameter vector $\vec{k}$ is rotated counterclockwise. When leaving out the unaffected mean-free part of Equation (2.2 – 12) and the offset point $\vec{t} = \vec{0}$ from Equation (2.2 – 6), it is easy to see that both ways lead to the same Gabor wavelet:

$$
\begin{aligned}
\psi_{\vec{k}}(\vec{x}) &= \frac{\vec{k}^2}{\sigma^2} \mathrm{e}^{-\frac{\vec{x}^2}{2\sigma^2}} \, \mathrm{e}^{i \vec{k}^{\mathrm{T}} \vec{x}} \\
&= \frac{k^2}{\sigma^2} \mathrm{e}^{-\frac{\vec{x}^2}{2\sigma^2}} \, \mathrm{e}^{i \, k (Q(\vartheta) \vec{e}_h)^{\mathrm{T}} \vec{x}} && \vec{k} = k \, Q(\vartheta) \, \vec{e}_h \\
&= \frac{k^2}{\sigma^2} \mathrm{e}^{-\frac{\vec{x}^2}{2\sigma^2}} \, \mathrm{e}^{i \, k \, \vec{e}_h^{\mathrm{T}} Q(\vartheta)^{\mathrm{T}} \vec{x}} && (Q(\vartheta) \, \vec{e}_h)^{\mathrm{T}} = e_h^{\mathrm{T}} Q(\vartheta)^{\mathrm{T}} \\
&= \frac{k^2}{\sigma^2} \mathrm{e}^{-\frac{\vec{x}^2}{2\sigma^2}} \, \mathrm{e}^{i \, \vec{e}_h^{\mathrm{T}} \left(k \, Q(\vartheta)^{\mathrm{T}} \vec{x} \right)} \\
&= \psi_{k,\vartheta}(\vec{x}) \; .
\end{aligned}
$$

Appendix B

Disparity Estimation

B.1 Disparity Estimation Between Gabor Jets

The disparity of two Gabor jets are estimated by using the Tailor expansion[1] $\cos(x) \approx 1 - \frac{1}{2}x^2$ of the cosine in the $S_{[D]}$ function:

$$\begin{aligned} S_{[D]}\big(\mathcal{J},\mathcal{J}'\big) &= \frac{\sum\limits_{j=0}^{J-1} a_j\, a'_j\, \cos\big(\phi_j - \phi'_j - \vec{k}_j^{\mathrm{T}}\vec{d}\big)}{\sqrt{\left(\sum\limits_{j=0}^{J-1} a_j^2\right)\left(\sum\limits_{j=0}^{J-1} a_j'^2\right)}} \\ &\approx \frac{\sum\limits_{j=0}^{J-1} a_j\, a'_j \left[1 - \frac{1}{2}\big(\phi_j - \phi'_j - \vec{k}_j^{\mathrm{T}}\vec{d}\big)^2\right]}{\sqrt{\left(\sum\limits_{j=0}^{J-1} a_j^2\right)\left(\sum\limits_{j=0}^{J-1} a_j'^2\right)}}\,. \end{aligned} \tag{B.1-1}$$

Since the disparity $\vec{d}$ with the maximal $S_{[D]}$ similarity value should be estimated, $S_{[D]}$ is maximized by calculating the gradient $\nabla S_{[D]}$ and setting it to zero in both directions d_h and d_v:

$$\nabla S_{[D]}\big(\mathcal{J},\mathcal{J}'\big) \approx \begin{pmatrix} \frac{\partial S_{[D]}}{\partial d_h} \\ \frac{\partial S_{[D]}}{\partial d_v} \end{pmatrix} \stackrel{!}{=} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{B.1-2}$$

[1] Theimer and Mallot [88] used the quadratic error function $\epsilon^2(\mathcal{J},\mathcal{J}') = \sum\limits_{j=0}^{J-1} a_j a'_j (\phi_j - \phi'_j - \vec{k}_j^{\mathrm{T}}\vec{d})^2$ directly, the resulting estimation of $\vec{d}$ is, of course, identical.

with:

$$\frac{\partial S_{[D]}}{\partial d_h} = -\frac{1}{2}\sum_{j=0}^{J-1} a_j\, a'_j\, \left(\phi_j - \phi'_j - d_h\, k_{j;h} - d_v\, k_{j;v}\right) 2\,(\text{-}k_{j;h}) \stackrel{!}{=} 0\,,$$
$$\frac{\partial S_{[D]}}{\partial d_v} = -\frac{1}{2}\sum_{j=0}^{J-1} a_j\, a'_j\, \left(\phi_j - \phi'_j - d_h\, k_{j;h} - d_v\, k_{j;v}\right) 2\,(\text{-}k_{j;v}) \stackrel{!}{=} 0\,. \tag{B.1–3}$$

Factoring out d_h and d_v in both equations leads to the system of linear equations:

$$\left(\sum_{j=0}^{J-1} a_j\, a'_j\, k_{j;h}^2\right) d_h + \left(\sum_{j=0}^{J-1} a_j\, a'_j\, k_{j;h}\, k_{j;v}\right) d_v = \sum_{j=0}^{J-1} a_j\, a_j{}'(\phi_j - \phi'_j)\, k_{j;h}\,,$$
$$\left(\sum_{j=0}^{J-1} a_j\, a'_j\, k_{j;h}\, k_{j;v}\right) d_h + \left(\sum_{j=0}^{J-1} a_j\, a'_j\, k_{j;v}^2\right) d_v = \sum_{j=0}^{J-1} a_j\, a'_j\, (\phi_j - \phi'_j)\, k_{j;v}\,, \tag{B.1–4}$$

which can easily be solved for d_h and d_v [99, 97] by:

$$\vec{d} = \mathbf{\Gamma}^{\text{-}1}\, \mathbf{\Phi}\,, \tag{B.1–5}$$

where:

$$\mathbf{\Gamma} = \begin{pmatrix} \mathbf{\Gamma}_{h,h} & \mathbf{\Gamma}_{h,v} \\ \mathbf{\Gamma}_{v,h} & \mathbf{\Gamma}_{v,v} \end{pmatrix}, \qquad \mathbf{\Gamma}_{h,v} = \sum_{j=0}^{J-1} k_{j;h}\, k_{j;v}\, a_j\, a'_j\,,$$
$$\mathbf{\Phi} = \begin{pmatrix} \mathbf{\Phi}_h \\ \mathbf{\Phi}_v \end{pmatrix}, \qquad \mathbf{\Phi}_h = \sum_{j=0}^{J-1} a_j\, a'_j\, k_{j;h}\, (\phi_j - \phi'_j)\,. \tag{B.1–6}$$

B.2 Maximum Likelihood Disparity Estimation

The estimation of the disparity between a Gabor jet $\mathcal{J}$ and a maximum likelihood Gabor jet $\mathcal{J}^{\mathrm{ML}}$ can be estimated similar to Appendix B.1. In this case, there is no need for making the detour over the cosine, but:

$$\begin{aligned} S_{[D]}^{\mathrm{ML}}\left(\mathcal{J}^{\mathrm{ML}},\mathcal{J}\right) = &-\frac{1}{2J}\sum_{j=0}^{J-1}\frac{\left(a_j-\mu_j^{[a]}\right)^2}{\kappa_j^{[a]}} \\ &-\frac{1}{2J}\sum_{j=0}^{J-1}\frac{d_\phi\left(\phi_j+\vec{k}_j^{\mathrm{T}}\vec{d}-\mu_j^{[\phi]}\right)^2}{\kappa_j^{[\phi]}} \end{aligned} \tag{B.2-1}$$

can be used directly[2] to derive the estimate of the disparity vector $\vec{d}$. Similar to Equation (B.1-2), the similarity function is partially differentiated with respect to d_h and d_v:

$$\begin{aligned} \frac{\partial S_{[D]}^{\mathrm{ML}}}{\partial d_h} &= -\frac{1}{J}\sum_{j=0}^{J-1}\frac{\left(\phi_j+d_h\,k_{j;h}+d_v\,k_{j;v}-\mu_j^{[\phi]}\right)k_{j;h}}{\kappa_j^{[a]}} \stackrel{!}{=} 0\,, \\ \frac{\partial S_{[D]}^{\mathrm{ML}}}{\partial d_v} &= -\frac{1}{J}\sum_{j=0}^{J-1}\frac{\left(\phi_j+d_h\,k_{j;h}+d_v\,k_{j;v}-\mu_j^{[\phi]}\right)k_{j;v}}{\kappa_j^{[a]}} \stackrel{!}{=} 0\,, \end{aligned} \tag{B.2-2}$$

with the first part of Equation (B.2-1) vanishing. Factoring out d_h and d_v leads to:

$$\begin{aligned} \left(\sum_{j=0}^{J-1}\frac{k_{j;h}^2}{\kappa_j^{[\phi]}}\right)d_h+\left(\sum_{j=0}^{J-1}\frac{k_{j;h}\,k_{j;v}}{\kappa_j^{[\phi]}}\right)d_v &= \sum_{j=0}^{J-1}\frac{\left(\mu_j^{[\phi]}-\phi_j\right)k_{j;h}}{\kappa_j^{[\phi]}}\,, \\ \left(\sum_{j=0}^{J-1}\frac{k_{j;h}\,k_{j;v}}{\kappa_j^{[\phi]}}\right)d_h+\left(\sum_{j=0}^{J-1}\frac{k_{j;v}^2}{\kappa_j^{[\phi]}}\right)d_v &= \sum_{j=0}^{J-1}\frac{\left(\mu_j^{[\phi]}-\phi_j\right)k_{j;v}}{\kappa_j^{[\phi]}}\,. \end{aligned} \tag{B.2-3}$$

[2] Temporarily, the d_ϕ function is ignored by setting: $d_\phi(x)=x$.

In turn, solving for d_h and d_v yields:

$$\mathbf{\Gamma}_{h,v} = \sum_{j=0}^{J-1} \frac{k_{j;h}\, k_{j;v}}{\kappa_j^{[\phi]}}\,, \qquad \mathbf{\Phi}_h = \sum_{j=0}^{J-1} \frac{\left(\mu_j^{[\phi]} - \phi_j\right) k_{j;h}}{\kappa_j^{[\phi]}}\,. \tag{B.2–4}$$

The resulting disparity vectors usually point into the correct direction, but the length of these vectors tend to be too small (examples are not shown here). This issue arises due to unstable phase differences caused by low absolute values. It can be solved by modifying the base similarity function:

$$S_{[D]}^{\mathrm{ML}}\big(\mathcal{J}^{\mathrm{ML}}, \mathcal{J}\big) = -\frac{1}{2J} \sum_{j=0}^{J-1} \frac{a_j\, \mu_j^{[a]} \left(d_\phi\left(\phi_j + \vec{k}_j^{\mathrm{T}}\vec{d} - \mu_j^{[\phi]}\right)\right)^2}{\kappa_j^{[\phi]}}\,, \tag{B.2–5}$$

similarly to Equation (B.1–1) by introducing the absolute values into the equation[3]. Solving for d_h and d_v changes the calculation of $\mathbf{\Gamma}$ and $\mathbf{\Phi}$ to:

$$\begin{aligned} \mathbf{\Gamma}_{h,v} &= \sum_{j=0}^{J-1} \frac{a_j\, \mu_j^{[a]}\, k_{j;h}\, k_{j;v}}{\kappa_j^{[\phi]}}\,, \\ \mathbf{\Phi}_h &= \sum_{j=0}^{J-1} \frac{a_j\, \mu_j^{[a]} \left(\mu_j^{[\phi]} - \phi_j\right) k_{j;h}}{\kappa_j^{[\phi]}}\,. \end{aligned} \tag{B.2–6}$$

B.3 Auto Focus

The estimation given in Equation (B.1–6) has one major problem. It ignores the important fact that phases and, thus, phase differences are circular. Hence, the phase correction $\vec{k}_j^{\mathrm{T}}\vec{d}$ that is applied to $\phi - \phi'$ is in fact $\vec{k}_j^{\mathrm{T}}\vec{d} + m_j\, 2\pi$:

$$S_{[D]}\big(\mathcal{J}, \mathcal{J}'\big) = \frac{\sum_{j=0}^{J-1} a_j\, a'_j\, \cos\!\big(\phi_j - \phi'_j - \vec{k}_j^{\mathrm{T}}\vec{d} - m_j\, 2\pi\big)}{\sqrt{\left(\sum_{j=0}^{J-1} a_j^2\right)\left(\sum_{j=0}^{J-1} a_j'^2\right)}}\,, \tag{B.3–1}$$

with each m_j being an unknown integral number. Unfortunately, when m_j is actually not vanishing, i. e., the disparity is larger than one half of the

[3] For legibility, the first term that vanishes during derivation is left out.

wavelength of the wavelet, the phase difference does no longer point to the correct location, but into the opposite direction. Wiskott *et al.* [96, 98, 97] proposed to leave out the highest frequency wavelets in the estimation of the disparity by starting at level ζ (which they called the *focus*):

$$\vec{d}_\zeta = \mathbf{\Gamma}_\zeta^{-1}\, \mathbf{\Phi}_\zeta\,, \tag{B.3-2}$$

with:

$$\begin{aligned} \mathbf{\Gamma}_{\zeta;h,v} &= \sum_{j=\zeta\,\nu_{\max}}^{\zeta_{\max}\,\nu_{\max}\text{-}1} k_{j;h}\, k_{j;v}\, a_j\, a'_j\,, \\ \mathbf{\Phi}_{\zeta;h} &= \sum_{j=\zeta\,\nu_{\max}}^{\zeta_{\max}\,\nu_{\max}\text{-}1} a_j\, a'_j\, k_{j;h}\, (\phi_j - \phi'_j)\,. \end{aligned} \tag{B.3-3}$$

Although Wiskott *et al.* [96, 98, 97] did not specify, which focus to use, there is an easy way to automatically choose a focus based on the currently examined Gabor jets. Starting with $\vec{d}_{\zeta_{\max}\text{-}1} = \vec{0}$, it can be tested whether half the wavelength of the current scale $\frac{2\pi}{k_\zeta}$ is less than the length of the disparity vector $\|\vec{d}_{\zeta+1}\|$. When this is the case, the estimation stops with the focus being the last scale used, i. e., $\zeta + 1$, and the returned disparity vector $\vec{d} = \vec{d}_{\zeta+1}$ is the last estimated vector. The focus for the $S^{\mathrm{ML}}_{[D]}$ disparity function can be estimated identically, the equations are not given here.

B.4 Phase Difference Correction

Leaving out some important, i. e., the highest frequency information cannot be the last word on the subject. Although the m_j values in Equation (B.3-1) are indeed unknown, they can be estimated easily by exploiting the estimates of the disparity vector of larger scale Gabor wavelet responses, i. e., $\vec{d}_\zeta$ from Equation (B.3-3). Starting with $m_j = 0$ for all Gabor wavelets of the largest scale $\zeta_{\max}$-1, the estimations of $\vec{d}_\zeta$ are refined in each level by incorporating Equation (B.3-2) and:

$$\begin{aligned} \mathbf{\Gamma}_{\zeta;h,v} &= \sum_{j=\zeta\,\nu_{\max}}^{\zeta_{\max}\,\nu_{\max}\text{-}1} k_{j;h}\, k_{j;v}\, a_j\, a'_j\,, \\ \mathbf{\Phi}_{\zeta;h} &= \sum_{j=\zeta\,\nu_{\max}}^{\zeta_{\max}\,\nu_{\max}\text{-}1} a_j\, a'_j\, k_{j;h}\, (\phi_j - \phi'_j - m_j\, 2\pi)\,. \end{aligned} \tag{B.4-1}$$

The integral m_j values can be easily estimated by:

$$m_j = \left\lfloor \frac{\phi_j - \phi'_j - \vec{k}_j^{\mathrm{T}} \vec{d}_{\zeta+1}}{2\pi} \right\rceil , \tag{B.4–2}$$

with $\lfloor \cdot \rceil$ being the rounding operator. The final disparity $\vec{d} = \vec{d}_0$ vector is simply the estimated disparity using all Gabor wavelet levels.

The same estimation can be done based on the $S^{\mathrm{ML}}_{[D]}$ similarity function. Instead of using the d_ϕ function that reduces the phase difference $\phi_j - \phi'_j$ into $]-\pi, \pi]$, the m_j values are introduced[3]:

$$S^{\mathrm{ML}}_{[D]}\left(\mathcal{J}^{\mathrm{ML}}, \mathcal{J}\right) = -\frac{1}{2J} \sum_{j=0}^{J\text{-}1} \frac{a_j \, \mu_j^{[a]} \left(\phi_j + \vec{k}_j^{\mathrm{T}} \vec{d} + m_j \, 2\pi - \mu_j^{[\phi]}\right)^2}{\kappa_j^{[\phi]}} . \tag{B.4–3}$$

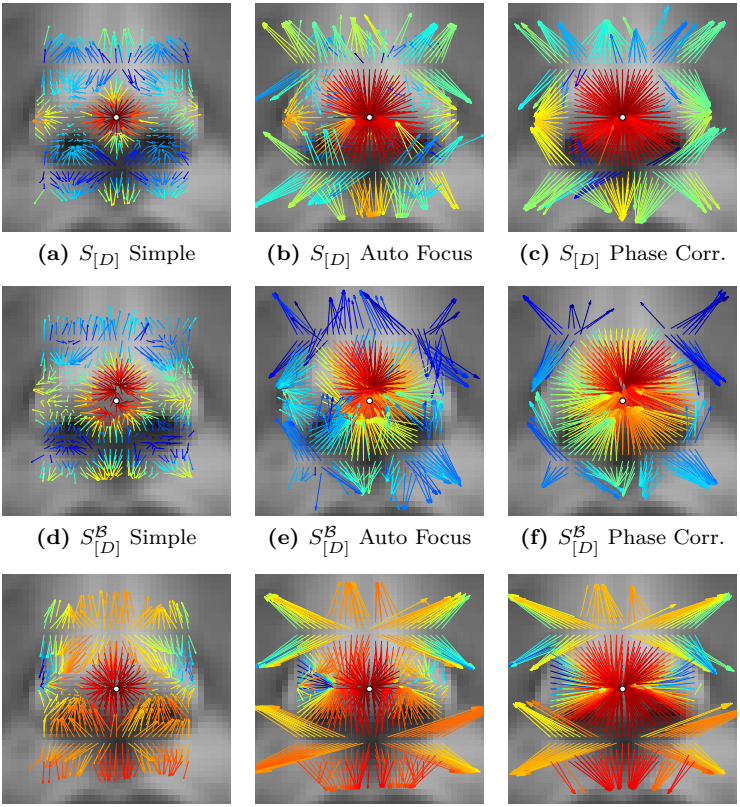
Computing the intermediate $\mathbf{\Gamma}_\zeta$ matrix and the $\mathbf{\Phi}_\zeta$ vector is straightforward:

$$\begin{aligned} \mathbf{\Gamma}_{\zeta;h,v} &= \sum_{j=\zeta \, \nu_{\max}}^{\zeta_{\max} \, \nu_{\max}\text{-}1} \frac{a_j \, \mu_j^{[a]} \, k_{j;h} \, k_{j;v}}{\kappa_j^{[\phi]}} , \\ \mathbf{\Phi}_{\zeta;h} &= \sum_{j=\zeta \, \nu_{\max}}^{\zeta_{\max} \, \nu_{\max}\text{-}1} \frac{a_j \, \mu_j^{[a]} \, (\mu_j^{[\phi]} - \phi_j - m_j \, 2\pi) \, k_{j;h}}{\kappa_j^{[\phi]}} . \end{aligned} \tag{B.4–4}$$

B.5 Comparison of Disparity Estimations

How well the simple disparity estimation, the disparity estimation with automatically limited focus, and disparity estimation with phase difference correction work, shall be illustrated with a simple example. From a probe image, which is the leftmost image shown in Figure 3.10(b), a 25×25 pixel region around the nose tip is considered. In this region, the Gabor jets are extracted and the disparities of these Gabor jets relative to a reference Gabor jet are computed. In the first test, the reference Gabor jet is taken from the same image at the desired nose tip location, while further tests compute disparities according to a $\mathcal{J}^{\mathcal{B}}$ or a $\mathcal{J}^{\mathrm{ML}}$ trained on 30 hand-labeled nose tip Gabor jets, where the probe image is not included in the training set.

Figure B.1 shows the disparities estimated with several disparity calculation setups. The arrows display the disparity vectors $\vec{d}$ estimated at the source of the arrow, the colors of the arrows code the similarity values, with



(a) $S_{[D]}$ Simple **(b)** $S_{[D]}$ Auto Focus **(c)** $S_{[D]}$ Phase Corr.

(d) $S^{\mathcal{B}}_{[D]}$ Simple **(e)** $S^{\mathcal{B}}_{[D]}$ Auto Focus **(f)** $S^{\mathcal{B}}_{[D]}$ Phase Corr.

(g) $S^{\mathrm{ML}}_{[D]}$ Simple **(h)** $S^{\mathrm{ML}}_{[D]}$ Auto Focus **(i)** $S^{\mathrm{ML}}_{[D]}$ Phase Corr.

Figure B.1: Disparity Estimations: *This figure exemplary displays different types of disparity estimations for the nose tip landmark. Each arrow shows the disparity vector* $\vec{d}$ *estimated from the Gabor jet at the that location and:*

- *(a), (b), and (c): Gabor jet* $\mathcal{J}$ *at the nose-tip,*
- *(d), (e), and (f):* $\mathcal{J}^{\mathcal{B}}$ *including nose-tip Gabor jets,*
- *(g), (h), and (i):* $\mathcal{J}^{\mathrm{ML}}$ *built from nose-tip Gabor jets.*

In combination, three different ways to compute the disparity are executed:

- *(a), (d), and (g): Simple (cf. Appendix B.1 and Appendix B.2),*
- *(b), (e), and (h): Auto focus (cf. Appendix B.3),*
- *(c), (f), and (i): Phase correction (cf. Appendix B.4).*

red being the highest value and blue the lowest. In the upper row, the reference Gabor jet is taken from the same image and, hence, the resulting disparity vectors are as precise as possible for the given setup. In the central row, disparities are estimated with the $S^{\mathcal{B}}_{[D]}$ bunch similarity function. Clearly, the estimated disparities point to different locations in the image, caused by the fact that different Gabor jets of the bunch won the maximum similarity calculation in the bunch similarity function (cf. Equation (2.4 – 11)), but none of them point to the correct position that is labeled with a white circle in the images. In opposition, the $S^{\mathrm{ML}}_{[D]}$ similarity functions that are shown in the lower row point to a common target position.

In the columns of Figure B.1, three different ways to deal with high frequency Gabor wavelet responses are employed. In the left-most column, the disparities are estimated with Equations (B.1 – 6) and (B.2 – 6), i. e., by simply using all scales without any phase correction. Clearly, the estimated disparities are quite good when the offset point is near to the nose tip, but offset positions further away only point to the right direction, while not being long enough. This changes when the focus is estimated automatically as proposed in Appendix B.3, the resulting disparity estimates are shown in the middle column of Figure B.1. This time the length of the disparity vectors are more appropriate, but the direction of the disparity vectors are imprecise when the offset point is further away. This can be seen most explicitly in Figure B.1(h), where the disparities estimated for the positions above and below the target point are not pointing directly downwards or upwards, respectively. When the phase difference correction described in Appendix B.4 is applied instead, both the length and the direction of the estimated disparity vectors are much more precise. Exemplary results for the nose landmark are shown in the last column of Figure B.1. Still, the estimated disparities are completely wrong when the nodes are too far away. There, the estimation $m_j = 0$ for the lowest frequency Gabor wavelet scales is not fulfilled. This could be solved by additionally using Gabor wavelets with lower frequencies, but then the estimations of the highest frequency Gabor wavelets might be incorrect and, thus, a mixture of auto focus and phase correction need to be settled.

Unfortunately, $S^{\mathrm{ML}}_{[D]}$ similarity values are high for some of the wrong estimations since $\vec{\kappa}^{[\phi]}$ used in Equation (B.4 – 4) is calculated without estimating the disparity. Including the disparity estimation in the calculation of $\vec{\kappa}^{[\phi]}$ decreases these similarity values slightly (the graphics are not shown here). Nonetheless, most of the disparities close to the correct offset point are estimated near-to-perfect. Hence, the disparity estimations of any $S_{[D]}$ disparity similarity function reported throughout this thesis is computed by using the phase correction algorithm introduced in Appendix B.4.

Appendix C

Gabor Wavelet Prefactor

Looking at the Gabor wavelet family creation:

$$\psi_{k,\vartheta}(\vec{x}) = k^2\,\psi\big(k\,Q(\vartheta)^{\mathrm{T}}\,\vec{x}\big) \qquad \text{(C.1–1)}$$

from Equation (2.2–5), the question arises, what motivates the prefactor k^2 instead of the prefactor k that would be required for fulfilling the unitary second moment as given in Equation (2.1–9) (cf. Sections 2.1 and 2.2). Würtz [105] opined that the scale factor k^2 is justified by the fact that for natural images, the Gabor wavelets in frequency domain should have norms proportional to k and, thus, the Gabor wavelet responses for different Gabor wavelet scales $\zeta = 0, \ldots, \zeta_{\max}$-1 (cf. Section 2.2.3) should be in the same order of magnitude.

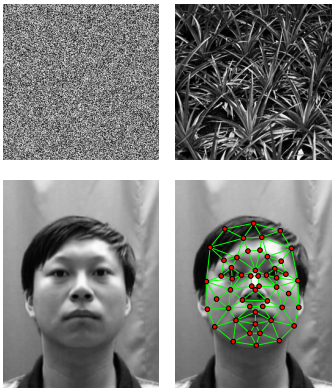
It is interesting to see whether this assumption also holds for facial images. Therefore, the average response of the Gabor wavelets are computed by:

$$q_j = \sum_{\vec{t}} \left| \mathcal{T}_{\vec{k}_j}\big(\vec{t}\big) \right| , \qquad \text{(C.1–2)}$$

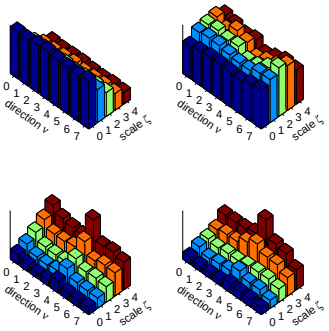
with $\vec{t}$ iterating over all pixels of the image, or over node positions of a graph. To test the assumption, the q_j values are calculated for some test images shown in Figure C.1(a), namely: random noise, a natural image[1], a facial image in resolution 168×224 pixel, and a face graph.

The resulting q_j values employing default prefactor k_ζ^2 are given in Figure C.1(b). It can be seen that the average response of the natural image is relatively stable in all five scales of Gabor wavelets and, thus, the expectation of Würtz [105] is fulfilled. Apparently, for the facial image and the face graph, the q_j values are not stable. With the default k_ζ^2 prefactor, the responses of the Gabor wavelets with larger spatial extends, i. e., with the higher ζ values usually dominate the high frequency ones. Therefore, in the $S_{[A]}$ Gabor jet similarity function (cf. Equation (2.3–10)), the high frequency

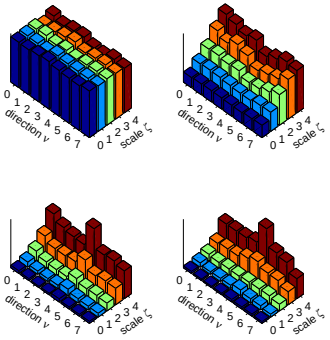
[1] The leaves image shown in Figure C.1(a) is part of the Vision Texture database[67]. It was down-scaled to resolution 256×256 pixel and converted to gray-scale.



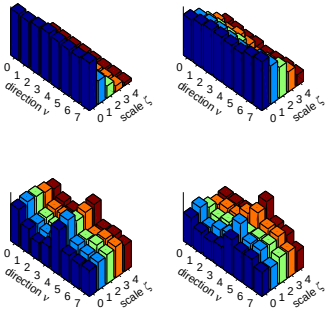
(a) Test objects



(b) Prefactor k_ζ^2 as proposed by [105]



(c) Prefactor k_ζ as required for Equation (2.1 – 9)



(d) Prefactor k_ζ^3 as suggested by me

Figure C.1: Gabor Wavelet Prefactors: *This figure displays averaged absolute values of Gabor wavelet responses of the test images: random noise, a natural image, a facial image, and a face graph shown in (a) with Gabor wavelet prefactors: (b) k_ζ^2, (c) k_ζ, and (d) k_ζ^3.*

responses are nearly completely suppressed. Interestingly, the q_j values for the noise image show the opposite shape, here the high frequency responses dominate the low frequency ones. For the scale factor k_ζ that is suggested by the two-dimensional wavelets (cf. Section 2.1.2), the results are even worse, see Figure C.1(c). Now, the noise image generates q_j values that are stable over scale, but obviously prefactor k_ζ does not work for facial images or face graphs either. Quite the contrary, when using prefactor k_ζ^3 as shown in Figure C.1(d), the difference of q_j values are approximately equaled out between different Gabor wavelet scales ζ, both for the face image and the face graph. Notably, the horizontal direction, i. e., with index $\nu = 4$ seems to be the predominant direction in facial images, as one would expect since most of the facial structures like mouth and eyes are horizontally expanded.

In fact, it is unclear whether the k_ζ^3 prefactor performs better for face detection, recognition, or classification. However, one indication in favor of k_ζ^3 is that for face recognition the Gabor jet similarity functions that normalize out the prefactor, i. e., $S_{[C]}$ and $S_{[c]}$ are superior to the ones that do not. Still, it is not known how the Gabor jet normalization impacts the recognition accuracies and, hence, the new prefactor would need to be tested exhaustively.

Unfortunately, there is one issue with the prefactor k_ζ^3 in the image reconstruction from Gabor wavelet responses. Prefactor k_ζ^3 in spatial domain would translate to prefactor k_ζ in frequency domain:

$$\check{\psi}_{\vec{k}}(\vec{\omega}) = k \left[\mathrm{e}^{-\frac{\sigma^2\left(\vec{\omega}-\vec{k}\right)^2}{2\vec{k}^2}} - \mathrm{e}^{-\frac{\sigma^2\left(\vec{\omega}^2+\vec{k}^2\right)}{2\vec{k}^2}} \right] . \qquad \text{(C.1–3)}$$

From this it follows that $\max_{\vec{\omega}} \ \check{\psi}_{\vec{k}}(\vec{\omega}) \approx k$ and, hence, $\check{C}_\psi(\vec{\omega})$ would have its maximum at $\vec{\omega}^2$. Finally, the calculation of the minimum value $C_{\min}$ (cf. Equations (6.1 – 16) and (6.1 – 17)):

$$C_{\min}(\vec{\omega}) = \frac{1}{4}\vec{\omega}^2 \qquad \text{(C.1–4)}$$

would be dependent on $\vec{\omega}$. Thus, $C_{\min}(\vec{\omega}_0)$ would vanish and, therefore, the dual Gabor wavelet (cf. Equation (6.1 – 18)):

$$\check{\psi}^d_{\vec{k}_j}(\vec{\omega}) = \frac{\check{\psi}_{\vec{k}_j}(\vec{\omega})}{\max\left\{\check{C}_\psi(\vec{\omega}), C_{\min}(\vec{\omega})\right\}} \qquad \text{(C.1–5)}$$

would be divided by zero at $\vec{\omega} = \vec{\omega}_0$ since both $\check{C}_\psi(\vec{\omega}_0)$ and $C_{\min}(\vec{\omega}_0)$ vanish. One way out of this dilemma is to neglect the k-value in the computation of

$\check{C}_\psi$ and $C_{\min}$ and compute the dual Gabor wavelets:

$$\check{\psi}^d_{\vec{k}_j}(\vec{\omega}) = \frac{\check{\psi}_{\vec{k}_j}(\vec{\omega})}{\vec{k}_j^2 \max\left\{\check{C}_\psi(\vec{\omega}), C_{\min}(\vec{\omega})\right\}} \tag{C.1–6}$$

neutralizing the k value. Apparently, the resulting dual Gabor wavelet would be correct, but this procedure is inapplicable for the Gaussian ψ_0 of $\Gamma^{\{g\}}$ or $\Gamma^{\{c\}}$ kernels (cf. Section 6.2.1). Thus, for the common set Γ, the dual kernels are still computable, but not for the extended sets $\Gamma^{\{g\}}$ and $\Gamma^{\{c\}}$.

Appendix D

Face Detection Schedules

GWT with Γ	
Global Move with $S_{[A]}$	
Scan Whole Image	true
Step Width	8
Global Move with $S_{[A]}$	
Local Area Size	17×17
Step Width	1
Scale Move with $S_{[A]}$	
Scale Horizontally	true
Scale Vertically	true
First Scale	0.80
Scale Step	0.10
Last Scale	1.20
Scale Move with $S_{[A]}$	
Scale Horizontally	true
Scale Vertically	false
First Scale	0.90
Scale Step	0.05
Last Scale	1.10
Scale Move with $S_{[A]}$	
Scale Horizontally	false
Scale Vertically	true
First Scale	0.90
Scale Step	0.05
Last Scale	1.10
Scan Local Move with $S_{[P]}$	
Local Area Size	17×17
Step Width	1
Topology Weight	0.01
Scan Local Move with $S_{[P]}$	
Local Area Size	17×17
Step Width	1
Topology Weight	0.10
Scan Local Move with $S_{[P]}$	
Local Area Size	17×17
Step Width	1
Topology Weight	1.00

(a) Scan Local Move

GWT with Γ	
Global Move with $S_{[A]}$	
Scan Whole Image	true
Step Width	8
Global Move with $S_{[A]}$	
Local Area Size	17×17
Step Width	1
Scale Move with $S_{[A]}$	
Scale Horizontally	true
Scale Vertically	true
First Scale	0.80
Scale Step	0.10
Last Scale	1.20
Scale Move with $S_{[A]}$	
Scale Horizontally	true
Scale Vertically	false
First Scale	0.90
Scale Step	0.05
Last Scale	1.10
Scale Move with $S_{[A]}$	
Scale Horizontally	false
Scale Vertically	true
First Scale	0.90
Scale Step	0.05
Last Scale	1.10
Scan Disparity Local Move	
Local Area Size	17×17
Step Width	4
Topology Weight	0.01
Scan Disparity Local Move	
Local Area Size	17×17
Step Width	4
Topology Weight	0.10
Scan Disparity Local Move	
Local Area Size	17×17
Step Width	4
Topology Weight	1.00

(b) Disparity Scan Local Move

Table D.1: Simple FaceGen Detection Schedules: *This table shows FaceGen schedules used for face detection and landmark localization, employing (a) the scan local move with $S_{[P]}$ and (b) the disparity scan local move using $S_{[D]}$ (cf. Section 5.2.1).*

GWT with $\Gamma^{\{e\}}$	
SSR Global Move with $S_{[A]}$	
Scan Whole Image	true
Step Width	6
Scale Base	$2^{\frac{1}{4}}$
First Scale	0.50
Last Scale	2.00
Angle Step	5
First/Last Angle	10
SSR Global Move with $S_{[A]}$	
Local Area Size	13×13
Step Width	3
Scale Base	$2^{\frac{1}{8}}$
First Scale	0.84
Last Scale	1.19
Scale Differences	1
Angle Step	2
First/Last Angle	4
SSR Global Move with $S_{[A]}$	
Local Area Size	7×7
Step Width	1
Scale Base	$2^{\frac{1}{16}}$
First Scale	0.92
Last Scale	1.09
Scale Differences	1
Angle Step	1
First/Last Angle	2
Standard. based on Detected Scale	
GWT with Γ	
Scan Local Move with $S_{[P]}$	
Local Area Size	17×17
Step Width	1
Topology Weight	0.01
Scan Local Move with $S_{[P]}$	
Local Area Size	17×17
Step Width	1
Topology Weight	0.10
Scan Local Move with $S_{[P]}$	
Local Area Size	17×17
Step Width	1
Topology Weight	1.00
Standard. based on Eyes	
GWT with Γ	

(a) Distance

GWT with $\Gamma^{\{e\}}$	
SSR Global Move with $S_{[A]}$	
Scan Whole Image	true
Step Width	6
Scale Base	$2^{\frac{1}{4}}$
First Scale	0.84
Last Scale	1.19
Angle Step	5
First/Last Angle	10
SSR Global Move with $S_{[A]}$	
Local Area Size	13×13
Step Width	3
Scale Base	$2^{\frac{1}{8}}$
First Scale	0.92
Last Scale	1.09
Scale Differences	1
Angle Step	2
First/Last Angle	4
SSR Global Move with $S_{[A]}$	
Local Area Size	7×7
Step Width	1
Scale Base	$2^{\frac{1}{16}}$
First Scale	0.92
Last Scale	1.09
Scale Differences	1
Angle Step	1
First/Last Angle	2
Standard. based on Detected Scale	
GWT with Γ	
Scan Local Move with $S_{[P]}$	
Local Area Size	17×17
Step Width	1
Topology Weight	0.01
Scan Local Move with $S_{[P]}$	
Local Area Size	17×17
Step Width	1
Topology Weight	0.10
Scan Local Move with $S_{[P]}$	
Local Area Size	17×17
Step Width	1
Topology Weight	1.00
Standard. based on Eyes	
GWT with Γ	

(b) Normal, Expression, and Lighting

Table D.2: CAS-PEAL Detection Schedules: *This table shows the schedules used for the detection of the CAS-PEAL face graphs and the scale and angle error estimations (cf. Section 5.2). The schedule from (a) is used in the Distance subset, while all other subsets employ the schedule in (b).*

GWT with $\Gamma^{\{e\}}$	
SSR Global Move with $S_{[A]}$	
Local Area Size	3×3
Step Width	5
Scale Base	$2\frac{1}{8}$
Scale Differences	2
GWT with Γ	
Scan Local Move with $S_{[P]}$	
Local Area Size	17×17
Step Width	1
Topology Weight	0.01
Scan Local Move with $S_{[P]}$	
Local Area Size	17×17
Step Width	1
Topology Weight	0.10
Scan Local Move with $S_{[P]}$	
Local Area Size	17×17
Step Width	1
Topology Weight	1.00

(a) CAS-PEAL Schedule

GWT with $\Gamma^{\{e\}}$	
SSR Global Move with $S_{[A]}$	
Local Area Size	3×3
Step Width	5
Scale Base	$2\frac{1}{16}$
Scale Differences	2
GWT with Γ	
Scan Local Move with $S_{[P]}$	
Local Area Size	17×17
Step Width	1
Topology Weight	0.01
Scan Local Move with $S_{[P]}$	
Local Area Size	17×17
Step Width	1
Topology Weight	0.10
Scan Local Move with $S_{[P]}$	
Local Area Size	17×17
Step Width	1
Topology Weight	1.00

(b) FRGC Schedule

Table D.3: Local Schedules: *This table shows the half-automatic landmark localization schedules of the Local setups of (a) the CAS-PEAL (cf. Section 5.3.2) and (b) the FRGC database (see Section 5.2.3). Both schedules expect that the image is standardized and the bunch or ML graph is already placed at the eye positions.*

GWT with $\Gamma^{\{e\}}$	
SSR Global Move with $S_{[A]}$	
Scan Whole Image	true
Step Width	6
Scale Base	$2^{\frac{1}{4}}$
First Scale	0.71
Last Scale	1.41
Angle Step	5
First/Last Angle	10
SSR Global Move with $S_{[A]}$	
Local Area Size	13 × 13
Step Width	3
Scale Base	$2^{\frac{1}{8}}$
First Scale	0.92
Last Scale	1.09
Scale Differences	1
Angle Step	2
First/Last Angle	4
SSR Global Move with $S_{[A]}$	
Local Area Size	7 × 7
Step Width	1
Scale Base	$2^{\frac{1}{16}}$
First Scale	0.92
Last Scale	1.09
Scale Differences	1
Angle Step	1
First/Last Angle	2
Standard. based on Detected Scale	
GWT with Γ	
Scan Local Move with $S_{[P]}$	
Local Area Size	17 × 17
Step Width	1
Topology Weight	0.01
Scan Local Move with $S_{[P]}$	
Local Area Size	17 × 17
Step Width	1
Topology Weight	0.10
Scan Local Move with $S_{[P]}$	
Local Area Size	17 × 17
Step Width	1
Topology Weight	1.00

(a) Global FRGC Schedule

GWT with $\Gamma^{\{e\}}$	
SSR Global Move with $S_{[A]}$	
Scan Whole Image	true
Step Width	6
Scale Base	$2^{\frac{1}{4}}$
First Scale	0.84
Last Scale	1.19
SSR Global Move with $S_{[A]}$	
Local Area Size	13 × 13
Step Width	3
Scale Base	$2^{\frac{1}{8}}$
First Scale	0.92
Last Scale	1.09
Scale Differences	1
Angle Step	2
First/Last Angle	2
SSR Global Move with $S_{[A]}$	
Local Area Size	7 × 7
Step Width	1
Scale Base	$2^{\frac{1}{16}}$
First Scale	0.96
Last Scale	1.04
Scale Differences	1
Angle Step	1
First/Last Angle	1
GWT with Γ	
Scan Local Move with $S_{[P]}$	
Local Area Size	13 × 13
Step Width	1
Topology Weight	0.01
Scan Local Move with $S_{[P]}$	
Local Area Size	13 × 13
Step Width	1
Topology Weight	0.10
Scan Local Move with $S_{[P]}$	
Local Area Size	13 × 13
Step Width	1
Topology Weight	1.00

(b) Human Genetics Schedule

Table D.4: Other Global Schedules: *This table shows two fully-automatic schedules employed for face detection and landmark localization in (a) the FRGC (cf. Section 5.2.3) and (b) the Human Genetic database (see Section 5.2.4). Since the images of the Human Genetic database are aligned at the eye positions, no image standardization is applied in (b).*

Bibliography

[1] T. Ahonen, A. Hadid, and M. Pietikainen. *Face recognition with local binary patterns.* In *European Conference on Computer Vision*, pages 469–481. Proc. Workshop Dynamical Vision, 2004.

[2] Mohamed Aly. *Face recognition using SIFT features.* Technical report, California Institute of Technology, 2006.

[3] M.S. Bartlett, J.R. Movellan, and T.J. Sejnowski. *Face recognition by independent component analysis.* Neural Networks, IEEE Transactions on, volume 13 (6), pages 1450–1464, November 2002.

[4] R. Beveridge, D.S. Bolme, M. Teixeira, and B. Draper. *The CSU face identification evaluation system user's guide version 5.0.*, 2003. `http://www.cs.colostate.edu/evalfacerec`.

[5] D. Blackburn, M. Bone, and P. Phillips. *Face recognition vendor test 2000: Evaluation report*, February 2001.

[6] Christian Blatter. *Wavelets - eine Einführung.* Vieweg, Wiesbaden, 2. edition, 2003.

[7] S. Böhringer, M. Günther, S. Sinigerova, R.P. Würtz, B. Horsthemke, and D. Wieczorek. *Automated syndrome detection in a set of clinical facial photographs.* American Journal of Medical Genetics, 2011.

[8] S. Böhringer, T. Vollmar, C. Tasse, R.P. Würtz, G. Gillessen-Käsbach, B. Horsthemke, and D. Wieczorek. *Syndrome identification based on 2D analysis software.* European Journal of Human Genetics, volume 14 (10), pages 1082–1089, 2006.

[9] David S. Bolme. *Elastic bunch graph matching.* Master's thesis, Colorado State University, May 2003. `http://www.cs.colostate.edu/evalfacerec/papers/EBGMThesis_Final.pdf`.

[10] J. Buhmann, J. Lange, and C. v.d. Malsburg. *Distortion invariant object recognition by matching hierarchically labeled graphs.* In *IJCNN International Conference on Neural Networks, Washington*, pages 155–159. IEEE, 1989.

[11] Bundeskriminalamt. *Face recognition as a search tool "Foto-Fahndung": Final report.* Technical report, Bundeskriminalamt, 2007.

[12] Matteo Carandini. *What simple and complex cells compute.* The Journal of Physiology, volume 577 (2), pages 463–466, December 2006.

[13] Ingrid Daubechies. *The wavelet transform, time-frequency localization and signal analysis.* IEEE Transactions on Information Theory, volume 36 (5), pages 961–1005, September 1990.

[14] John G. Daugman. *Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters.* Journal of the Optical Society of America A, volume 2 (7), pages 1160–1169, July 1985.

[15] K. Delac, M. Grgic, and S. Grgic. *Statistics in face recognition: analyzing probability distributions of PCA, ICA and LDA performance results.* In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pages 289–294, September 2005.

[16] G. Doddington, W. Liggett, A. Martin, M. Przybocki, and D. Reynolds. *Sheep, goats, lambs and wolves: A statistical analysis of speaker performance in the NIST 1998 speaker recognition evaluation.* In *Proceeding ICSLP*, 1998.

[17] EPIC and Privacy International. *Privacy and human rights 2001: An international survey of privacy laws and developments.* Electronic Privacy Information Center (EPIC), 2001.

[18] Alberto Escalante and Laurenz Wiskott. *Gender and age estimation from synthetic face images.* In Eyke Hüllermeier and Rudolf Kruse, editors, *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2010*, pages 240–249, 2010.

[19] Kamran Etemad and Rama Chellappa. *Discriminant analysis for recognition of human face images.* Journal of the Optical Society of America A, volume 14, pages 1724–1733, 1997.

[20] J.-M. Fellous, L. Wiskott, N. Krüger, and C. v.d. Malsburg. *Face recognition by elastic bunch graph matching.* In *Proceedings of the International Conference on Vision, Recognition, Action: Neural Models of Mind and Machine.* Boston University, USA, May 1997.

[21] Robert Fischer. *Automatic facial expression analysis and emotional classification.* Master's thesis, University of Applied Science Darmstadt, 2004. `http://cbcl.mit.edu/publications/theses/thesis-masters-fischer-robert.pdf`.

[22] 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, volume 38, pages 149–161, January 2008.

[23] 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.

[24] Cong Geng and Xudong Jiang. *Face recognition using SIFT features.* In *International Conference on Image Processing*, pages 3313–3316, 2009.

[25] A.B.A. Graf and F.A. Wichmann. *Gender classification of human faces.* In *Biologically Motivated Computer Vision 2002, LNCS 2525*, pages 491–501, 2002.

[26] H.-M. Gross, H.-J. Böhme, Ch. Schröter, St. Müller, A. König, E. Einhorn, Ch. Martin, M. Merten, and A. Bley. *TOOMAS: Interactive shopping guide robots in everyday use – final implementation and experiences from long-term field trials.* In *International Conference on Intelligent Robots and Systems*, pages 2005–2012. IEEE, 2009.

[27] H.-M. Gross, H.-J. Böhme, Ch. Schröter, St. Müller, A. König, Ch. Martin, M. Merten, and A. Bley. *ShopBot: Progress in developing an interactive mobile shopping assistant for everyday use.* In *IEEE International Conference on Systems, Man, and Cybernetics*, 2008.

[28] Manuel Günther. *Klassifikation von Gesichtern mit optimierten lokalen Graphen auf 2D und 3D Bilddaten.* Diploma thesis, Technische Universität Ilmenau, March 2005.

[29] Manuel Günther and Rolf P. Würtz. *Face detection and recognition using maximum likelihood classifiers on Gabor graphs.* International Journal of Pattern Recognition and Artificial Intelligence, volume 23 (3), pages 433–461, May 2009.

[30] Dennis Haufe. *Einfluss der Bildauflösung auf Gesichtserkennung durch Graphenvergleich.* BSc thesis, Institut für Neuroinformatik, Ruhr-Universität Bochum, July 2008.

[31] Dennis Haufe. *Gabor phases for face classification.* Master's thesis, Institut für Neuroinformatik, Ruhr-Universität Bochum, May 2011.

[32] A. Heinrichs, M.K. Müller, A.H.J. Tewes, and R.P. Würtz. *Graphs with principal components of Gabor wavelet features for improved face recognition.* In Gabriel Cristóbal, Bahram Javidi, and Santiago Vallmitjana, editors, *Information Optics: 5th International Workshop on Information Optics; WIO'06*, pages 243–252. American Institute of Physics, 2006.

[33] R. Heintz, E. Monari, and G. Schäfer. *Local invariant object localization based on Gabor feature space.* In *Proceedings 5th International Conference on Visualization, Imaging and Image Processing*, pages 575–580, 2005.

[34] D.H. Hubel and T.N. Wiesel. *Receptive fields, binocular interaction and functional architecture in the cat's visual cortex.* The Journal of Physiology, volume 160, pages 106–154, January 1962.

[35] Zheng Ji and Bao-Liang Lu. *Gender classification based on support vector machine with automatic confidence.* In *Proceedings of the 16th International Conference on Neural Information Processing: Part I*, ICONIP '09, pages 685–692, Berlin, Heidelberg, 2009. Springer-Verlag.

[36] D. González Jiménez, M. Bicego, J.W.H. Tangelder, B.A.M. Schouten, Onkar O. Ambekar, J. Alba-Castro, E. Grosso, and M. Tistarelli. *Distance measures for gabor jets-based face authentication: A comparative evaluation.* In *Proceedings of the international conference on Advances in Biometrics*, pages 474–483, Berlin, Heidelberg, 2007. Springer-Verlag.

[37] J.P. Jones and L.A. Palmer. *An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex.* Journal of Neurophysiology, volume 58 (6), pages 1233–1258, December 1987.

[38] J.P. Jones and L.A. Palmer. *The two-dimensional spatial structure of simple receptive fields in cat striate cortex.* Journal of Neurophysiology, volume 58 (6), pages 1187–1211, December 1987.

[39] Karl Dirk Kammeyer and Kristian Kroschel. *Digitale Signalverarbeitung : Filterung und Spektralanalyse.* Teubner, Stuttgart, 1989.

[40] Simon Kriegel. *The application of active appearance models to comprehensive face analysis.* Technical report, TU München, April 2007.

[41] J. Križaj, V. Štruc, and N. Pavešič. *Adaptation of SIFT features for robust face recognition.* In *ICIAR10*, pages 394–404, 2010.

[42] N. Krüger, M. Pötzsch, and C. v.d. Malsburg. *Determination of face position and pose with a learned representation based on labelled graphs.* Image and Vision Computing, volume 15, pages 665–673, 1997.

[43] M. Lades, J.C. Vorbrüggen, J. Buhmann, J. Lange, C. v.d. Malsburg, R.P. Würtz, and W. Konen. *Distortion invariant object recognition in the dynamic link architecture.* IEEE Transactions on Computers, volume 42, pages 300–311, 1993.

[44] Martin Lades. *Invariant object recognition with dynamical links, robust to variations in illumination.* PhD thesis, Ruhr University Bochum, 1995.

[45] H.S. Loos, D. Wieczorek, R.P. Würtz., C. v.d. Malsburg, and B. Horsthemke. *Computer-based recognition of dysmorphic faces.* European Journal of Human Genetics, volume 11, pages 555–560, 2003.

[46] David G. Lowe. *Object recognition from local scale-invariant features.* In *Proceedings of the International Conference on Computer Vision ICCV, Corfu*, volume 2, pages 1150–1157, Los Alamitos, CA, USA, August 1999. IEEE Computer Society.

[47] David G. Lowe. *Distinctive image features from scale-invariant keypoints.* International Journal of Computer Vision, volume 60, pages 91–110, November 2004.

[48] Ch. Martin, U. Werner, and H.-M. Gross. *A real-time facial expression recognition system based on active appearance models using gray images and edge images.* In *IEEE International Conference on Face and Gesture Recognition*, pages 1–6, 2008.

[49] Yves Meyer. *Ondelettes et opérateurs. I.* Actualités Mathématiques. [Current Mathematical Topics]. Hermann, Paris, 1990. Ondelettes. [Wavelets].

[50] B. Moghaddam, C. Nastar, and A. Pentland. *Bayesian face recognition with deformable image models.* In *CIAP01*, pages 26–35, 2001.

[51] B. Moghaddam, W. Wahid, and A. Pentland. *Beyond eigenfaces: Probabilistic matching for face recognition.* IEEE International Conference on Automatic Face and Gesture Recognition, pages 30–35, 1998.

[52] Baback Moghaddam and Alex Pentland. *Probabilistic visual learning for object detection.* In *International Conference on Computer Vision*, pages 786–793, Cambridge, USA, June 1995.

[53] Baback Moghaddam and Alex Pentland. *Probabilistic visual learning for object representation.* IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 19 (7), pages 696–710, 1997.

[54] M.N. Moustafa and H. Mahdi. *A simple evaluation of face detection algorithms using unpublished static images.* In *10th International Conference on Intelligent Systems Design and Applications (ISDA)*, pages 1–5, December 2010.

[55] Marco K. Müller. *Finden von Punktkorrespondenzen zwischen Gesichtsbildern.* Internal report, Institut für Neuroinformatik, Ruhr-Universität, D-44780 Bochum, Germany, March 2004. `ftp://ftp.ini.rub.de/pub/manuscripts/IRINI/irini2004-01/irini2004-01.pdf`.

[56] Marco K. Müller. *Lernen von Identitäteserkennung unter Bildvariation.* PhD thesis, Ruhr-Universität Bochum, Germany, 2010.

[57] Marco K. Müller and Rolf P. Würtz. *Learning from examples to generalize over pose and illumination.* In *Proceedings of the 19th International Conference on Artificial Neural Networks: Part II*, ICANN '09, pages 643–652, Berlin, Heidelberg, 2009. Springer-Verlag.

[58] M.K. Müller, A. Heinrichs, A.H.J. Tewes, A. Schäfer, and R.P. Würtz. *Similarity rank correlation for face recognition under unenrolled pose.* In Seong-Whan Lee and Stan Z. Li, editors, *Advances in Biometrics, ICB 2007*, LNCS 4642, pages 67–76. Springer-Verlag Berlin Heidelberg, August 2007.

[59] National Science and Technology Council. *Biometrics testing and statistics*, August 2006. `http://www.biometrics.gov/ReferenceRoom/Introduction.aspx`.

[60] Norbert Neuser. *Die Auswirkung von Positionierungsfehlern auf graphenbasierte Gesichtserkennung.* BSc thesis, Institut für Neuroinformatik, Ruhr-Universität Bochum, October 2008.

[61] H. Nyquist. *Certain topics in telegraph transmission theory.* In *American Institute of Electrical Engineers Transactions*, volume 47, pages 617–644, April 1928.

[62] A.J. O'Toole, P.J. Phillips, F. Jiang, J. Ayyad, N. Penard, and H. Abdi. *Face recognition algorithms surpass humans matching faces over changes in illumination.* IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 29, pages 1642–1646, 2007.

[63] P. Phillips, P. Grother, R. Micheals, D. Blackburn, E. Tabassi, and M. Bone. *Face recognition vendor test 2002: Evaluation report*, March 2003.

[64] P. Phillips, P. Rauss, and S.Z. Der. *FERET (face recognition technology) recognition algorithm development and test results.* Army Research Lab technical report, October 1996.

[65] P.J. Phillips, P.J. Flynn, T. Scruggs, K.W. Bowyer, and W. Worek. *Preliminary face recognition grand challenge results.* In *Proceedings 7th International Conference on Automatic Face and Gesture Recognition*, pages 15–24, 2006.

[66] P.J. 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*, 2007.

[67] R. Picard, C. Graczyk, S. Mann, J. Wachman, L. Picard, and L. Campbell. *Vision texture.* `http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html`, 2002.

[68] D.A. Pollen and S.F. Ronner. *Spatial computation performed by simple and complex cells in the visual cortex of the cat.* Vision Res, volume 22 (1), pages 101–118, 1982.

[69] D.A. Pollen and S.F. Ronner. *Visual cortical neurons as localized spatial frequency filters.* Transactions on Systems, Man and Cybernetics, volume 13, pages 907–916, 1983.

[70] M. Pötzsch, T. Maurer, L. Wiskott, and C. v.d. Malsburg. *Reconstruction from graphs labeled with responses of Gabor filters.* In C. v.d. Malsburg, W. v. Seelen, J.C. Vorbrüggen, and B. Sendhoff, editors, *Proceedings of the ICANN 1996*, Springer Verlag, Berlin, Heidelberg, New York, pages 845–850, Bochum, July 1996.

[71] Michael Pötzsch. *Die Behandlung der Wavelet-Transformation von Bildern in der Nähe von Objektkanten.* Internal report, Institut für Neuroinformatik, Ruhr-Universität, D-44780 Bochum, Germany, May 1994. `ftp://ftp.ini.rub.de/pub/manuscripts/IRINI/body94-04.ps.gz`.

[72] Jiang Qiang-Rrong and Gao Yuan. *Face recognition based on graph kernels.* In *2010 The 2nd IEEE International Conference on Information Management and Engineering (ICIME)*, pages 226–230, April 2010.

[73] Yuan Ren. *Facial expression recognition system.* PhD thesis, University of Waterloo (Canada), June 2008.

[74] H.A. Rowley, S. Baluja, and T. Kanade. *Neural network-based face detection.* IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 20 (1), pages 23–38, 1998.

[75] H.A. Rowley, S. Baluja, and T. Kanade. *Rotation invariant neural network-based face detection.* In *CVPR'98*, pages 38–44, 1998.

[76] A. Ryan, J.F. Cohn, S. Lucey, J. Saragih, P. Lucey, F.d.l. Torre, and A. Ross. *Automated facial expression recognition system.* In *IEEE International Carnahan Conference on Security Technology*, October 2009.

[77] Robert E. Schapire. *The boosting approach to machine learning: An overview.* In *MSRI Workshop on Nonlinear Estimation and Classification*, 2002.

[78] Nele De Schepper. *Multi-dimensional continuous wavelet transforms and generalized fourier transforms in clifford analysis.* PhD thesis, Universiteit Gent, Faculteit Ingenieurswetenschappen, Vakgroep Wiskundige Analyse, 2007.

[79] H.J. Schneider, R.P. Kosilek, M. Günther, J. Römmler., G.K. Stalla, C. Sievers, M. Reincke, J. Schopohl, and R.P. Würtz. *A novel approach to the detection of acromegaly: Accuracy of diagnosis by automatic face classification.* Journal of clinical endocrinology and metabolism, July 2011.

[80] C.E. Shannon. *Communication in the presence of noise.* Proceedings of the IRE, volume 37 (1), pages 10–21, 1949.

[81] Yunlong Sheng. *Wavelet transform*, In Alexander D. Poularikas, editor, *The Transforms and Applications Handbook, Second Edition*, chapter 10. CRC press, 2000.

[82] L. Sirovich and M. Kirby. *Low-dimensional procedure for the characterization of human faces.* Journal of the Optical Society of America A, volume 4 (3), pages 519–524, March 1987.

[83] Benedikt Stratmann. *Einfluss der Graphenstruktur auf die Leistung eines Gesichtserkennungssystems.* BSc thesis, Institut für Neuroinformatik, Ruhr-Universität Bochum, January 2010.

[84] Kah-Kay Sung and Tomaso Poggio. *Example-based learning for view-based human face detection.* IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 20, pages 39–51, 1998.

[85] Marcio Luis Teixeira. *The bayesian intrapersonal/extrapersonal classifier.* Master's thesis, Colorado State University, July 2003. `http://www.cs.colostate.edu/evalfacerec/papers/teixeiraThesis.pdf`.

[86] A. Tewes, R.P. Würtz, and C. v.d. Malsburg. *A flexible object model for recognising and synthesising facial expressions.* In *AVBPA*, pages 81–90, 2005.

[87] Andreas Tewes. *A flexible object model for encoding and matching human faces.* PhD thesis, Ruhr-University Bochum, November 2005.

[88] Wolgang M. Theimer and Hanspeter A. Mallot. *Phase-based binocular vergence control and depth reconstruction using active vision.* CVGIP: Image Understanding, volume 60 (3), pages 343–358, 1994.

[89] Matthew Turk and Alex Pentland. *Eigenfaces for recognition.* Journal of Cognitive Neuroscience, volume 3 (1), pages 71–86, 1991.

[90] Paul Viola and Michael Jones. *Robust real-time object detection.* In *International Journal of Computer Vision*, volume 57, pages 137–154, 2002.

[91] T. Vollmar, B. Maus, R.P. Würtz, G. Gillessen-Käsbach, 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, volume 51, pages 44–53, 2008.

[92] Frank Wallhoff. *Facial expressions and emotion database.* Technical report, Technische Universität München, 2006. `http://www.mmk.ei.tum.de/~waf/fgnet/feedtum.html`.

[93] Xiaogang Wang and Xiaoou Tang. *Bayesian face recognition using Gabor features.* In *Proceedings of the 2003 ACM SIGMM workshop on Biometrics methods and applications*, WBMA '03, pages 70–73, 2003.

[94] Günter Westphal. *Feature-driven emergence of model graphs for object recognition and categorization.* PhD thesis, University of Lübeck, Germany, 2007.

[95] M. Wimmer, U. Zucker, and B. Radig. *Human capabilities on video-based facial expression recognition.* In Dirk Reichardt and Paul Levi, editors, *Proceedings of the 2nd Workshop on Emotion and Computing – Current Research and Future Impact*, pages 7–10, Osnabrück, Germany, September 2007.

[96] L. Wiskott, J.-M. Fellous, N. Krüger, and C. v.d. Malsburg. *Face recognition by elastic bunch graph matching.* IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 19, pages 775–779, July 1997.

[97] L. Wiskott, J.-M. Fellous, N. Krüger, and C. v.d. Malsburg. *Face recognition by elastic bunch graph matching.* In G. Sommer, K. Daniilidis, and J. Pauli, editors, *Proc. 7th Intern. Conf. on Computer Analysis of Images and Patterns, CAIP'97, Kiel*, number 1296 in Lecture Notes in Computer Science, pages 456–463, Heidelberg, September 1997. University Kiel, Springer-Verlag.

[98] L. Wiskott, J.-M. Fellous, N. Krüger, and C. v.d. Malsburg. *Face recognition by elastic bunch graph matching.* In L.C. Jain, U. Halici, I. Hayashi, and S.B. Lee, editors, *Intelligent Biometric Techniques in Fingerprint and Face Recognition*, chapter 11, pages pages 355–396. CRC Press, 1999.

[99] Laurenz Wiskott. *Labeled graphs and dynamic link matching for face recognition and scene analysis*, volume 53 of *Reihe Physik*. Verlag Harri Deutsch, Thun, Frankfurt am Main, Germany, 1995. PhD thesis.

[100] Laurenz Wiskott and Terrence Sejnowski. *Slow feature analysis: Unsupervised learning of invariances.* Neural Computation, volume 14 (4), pages 715–770, 2002.

[101] I.J. Wundrich, C. v.d. Malsburg, and R.P. Würtz. *Image reconstruction from Gabor magnitudes.* In *BMCV '02: Proceedings of the Second International Workshop on Biologically Motivated Computer Vision*, pages 117–126, London, UK, 2002. Springer-Verlag.

[102] I.J. Wundrich, C. v.d. Malsburg, and R.P. Würtz. *Image representation by complex cell responses.* Neural Computation, volume 16 (12), pages 2563–2575, 2004.

[103] Ingo J. Wundrich. *Untersuchungen zur Rekonstruierbarkeit lokaler Bildmerkmale aus der Gaborwavelettransformierten.* Diploma thesis, Institut für Neuroinformatik, Ruhr-Universität Bochum, Germany, October 1998. `ftp://ftp.ini.rub.de/pub/manuscripts/IRINI/irini99-03/irini99-03.ps.gz`.

[104] Ingo J. Wundrich. *Parametrisierte zweidimensionale Modelle für dreidimensionale Gesichtserkennung.* PhD thesis, Electrical Engineering Dept., Univ. of Bochum, Germany, July 2004.

[105] Rolf P. Würtz. *Multilayer dynamic link networks for establishing image point correspondences and visual object recognition.* PhD thesis, Fakultät für Physik und Astronomie, Ruhr-Universität, D-44780 Bochum, Germany, December 1994.

[106] Wendy S. Yambor. *Analysis of PCA-based and Fisher discriminant-based image recognition algorithms.* Master's thesis, Colorado State University, July 2000.

[107] W.S. Yambor, B. Draper, and R. Beveridge. *Analyzing PCA-based face recognition algorithms: Eigenvector selection and distance measures.* World Scientific Press, Singapore, 2002.

[108] M.-H. Yang, D.J. Kriegman, and N. Ahuja. *Detecting faces in images: A survey.* IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 24 (1), pages 34–58, 2002.

[109] Cha Zhang and Zhengyou Zhang. *A survey of recent advances in face detection.* Technical report, Microsoft Research, June 2010. `http://research.microsoft.com/pubs/132077/facedetsurvey.pdf`.

[110] 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.* Computer Vision, IEEE International Conference on, volume 1, pages 786–791, 2005.

[111] W. Zhang, S. Shan, L. Qing, X. Chen, and W. Gao. *Are Gabor phases really useless for face recognition?* Pattern Analysis & Applications, volume 12, pages 301–307, 2009.

[112] W. Zhao, A. Krishnaswamy, R. Chellappa, D.L. Swets, and J. Weng. *Discriminant analysis of principal components for face recognition*, In H. Wechsler, P.J. Phillips, V. Bruce, F.F. Soulie, and T.S. Huang, editors, *Face Recognition: From Theory to Applications*, pages 73–85.

Springer Verlag Berlin, 1998. http://www.face-rec.org/algorithms/LDA/zhao98discriminant.pdf.

[113] Ursula Zucker. *Facial expression recognition - a comparison between humans and algorithms.* Systementwicklungsprojekt, TU München, February 2007. http://www9-old.in.tum.de/people/wimmerm/lehre/sep_zucker/sep_zucker.pdf.

View publication stats