Lehrstuhl für Bildverarbeitung
Institute of Imaging & Computer Vision

Texture Classification by Modeling Joint Distributions of Local Patterns with Gaussian Mixtures

Henning Lategahn and Sebastian Groß and Thomas Stehle and Til Aach
Institute of Imaging and Computer Vision
RWTH Aachen University, 52056 Aachen, Germany
tel: +49 241 80 27860, fax: +49 241 80 22200
web: www.lfb.rwth-aachen.de

in: IEEE Transactions on Image Processing. See also BibTeX entry below.

BibTeX:

```
@article{LAT10a,
author = {Henning Lategahn and Sebastian Gro{\ss} and Thomas Stehle and Til Aach},
title = {Texture Classification by Modeling Joint Distributions of Local Patterns
                        with Gaussian Mixtures},
journal = {IEEE Transactions on Image Processing},
volume = {19},
number = {6},
year = {2010},
pages = {1548--1557}}
```


document created on: September 12, 2011
created from file: paper.tex
cover page automatically created with CoverPage.sty
(available at your favourite CTAN mirror)

RWTH AACHEN
RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN

Author Addresses:
Dipl.-Inform. Henning Lategahn
lategahn@mrt.uni-karlsruhe.de
Institute of Measurement and Control
Karlsruhe Institute of Technology (KIT)
Engler-Bunte-Ring 21
D-76131 Karlsruhe, Germany
phone: +49 721 6082748

Dipl.-Ing. Sebastian Gross
sebastian.gross@lfb.rwth-aachen.de
Dipl.-Inform. Thomas Stehle
thomas.stehle@lfb.rwth-aachen.de
Prof. Dr.-Ing. Til Aach
til.aach@lfb.rwth-aachen.de
Institute of Imaging & Computer Vision
RWTH Aachen University
D-52056 Aachen, Germany
phone: +49 241 8027860, fax: +49 241 8022200

EDICS:
ARS-RBS: Region, Boundary and Shape Analysis
SMR-SMD: Statistical-Model Based Methods

Texture Classification by Modeling Joint Distributions of Local Patterns with Gaussian Mixtures

Henning Lategahn, Sebastian Gross, *Student Member, IEEE*, Thomas Stehle, *Student Member, IEEE*, Til Aach, *Senior Member, IEEE*

***Abstract*— Texture classification generally requires the analysis of patterns in local pixel neighborhoods. Statistically, the underlying processes are comprehensively described by their joint probability density functions (jPDFs). Even for small neighborhoods, however, stable estimation of jPDFs by joint histograms (jHSTs) is often infeasible, since the number of entries in the jHST exceeds by far the number of pixels in a typical texture region. Moreover, evaluation of distance functions between jHSTs is often computationally prohibitive. Practically, the number of entries in a jHST is therefore reduced by considering only two-pixel patterns, leading to 2D-jHSTs known as co-occurrence matrices, or by quantization of the gray levels in local patterns to only two gray levels, yielding local binary patterns (LBPs). Both approaches result in a loss of information. We introduce here a framework for supervised texture classification which reduces or avoids this information loss. Local texture neighborhoods are first filtered by a filter bank. Without further quantization, the jPDF of the filter responses is then described parametrically by Gaussian mixture models (GMMs). We show that the parameters of the GMMs can be reliably estimated from small image regions. Moreover, distances between the thus modelled jPDFs of different texture patterns can be computed efficiently in closed form from their model parameters. We furthermore extend this texture descriptor to achieve full invariance to rotation. We evaluate the framework for different filter banks on the Brodatz texture set. We first show that combining the LBP difference filters with the GMM-based density estimator outperforms the classical LBP approach and its codebook extensions. When replacing these – rather elementary – difference filters by the wavelet frame transform (WFT), the performance of the framework on all 111 Brodatz textures exceeds the one obtained more recently by spin image and RIFT descriptors in [Lazebnik et al., 2005].**

***Index Terms*— Texture, Classification, Filter Banks, Wavelet Frame Transform, Parametric Models, Gaussian Mixture Models, Support Vector Machines.**

I. Introduction

Texture in images has often rather concisely been described as a "neighborhood property" [1], [2], [3], [4]. Texture classification consequently requires the analysis of patterns in local pixel neigborhoods. From a statistical point of view, the random process generating a certain texture pattern is completely described by its joint probability density function (jPDF). Even for small neighborhoods, though, stable estimation of jPDFs by joint histograms (jHSTs) is practically often infeasible, since the number of entries in the jHST exceeds by far the number of pixels in a texture region of typical size - a phenomenon which is known as the curse of dimensionality. A such estimated jHST is then almost empty. An attempt to overcome this problem using Parzen window estimation would require huge efforts in terms of both time and storage. Moreover, computational evaluation of distance measures between jHSTs in order to assign the observed patterns to classes is also often prohibitive. Different approaches to extract partial information about the underlying jPDFs from the observed texture samples were therefore developed. A classical method is the so-called texture energy transform by Laws [5], where a bank of local filter masks is applied to the texture, followed by computing the local variance in each channel. The set of variances can be shown to be a linear transform of the covariances of the texture process [6], [7] (cf. for the DCT also [8], [9, p.76]), i.e., of second order joint moments of its jPDF. These describe the texture process completely only if its jPDF is Gaussian (and zero-mean), which, however, is generally too rigid a model. For supervised segmentation, Ade replaced Laws' ad-hoc filters by so-called eigenfilters [10], while the use of quadrature filters allowed a more elegant computation of local energy [7], [11], [12], for instance in Gabor filter banks [13], [14], [15], [16], [17], [18]. Another alternative are wavelet filter banks, for instance the shift-invariant wavelet frame transform [19]. Unser [3] has shown that convolution operators yield projections of histograms that approximate a jPDF by means of local linear transforms of neighborhoods, and derived conditions for optimal separability (cf. also [20]). Spectral marginal histograms computed from filter bank outputs are used in [21]. A comparative review of filter bank approaches to texture classification can be found in [22].

An alternative to extracting information about a jPDF via moments is to reduce the number of entries in a jHST by, e.g., considering only pixel pairs. This leads to 2D-jHSTs known as gray level co-occurrence matrices (GLCMs) [1], which often serve as an intermediate step from which further features are computed [1], [23]. A related idea combining the consideration of pixel pairs with elementary filters is to estimate 1D histograms of differences of neighboring pixel intensities (gray level difference histograms) [24]. A scheme

H. Lategahn was with the Institute of Imaging and Computer Vision, RWTH Aachen University, Germany. He is now with the Institute of Measurement and Control, Karlsruhe Institute of Technology (KIT), Germany. S. Gross, T. Stehle and T. Aach are with the Institute of Imaging & Computer Vision, RWTH Aachen University, Germany. This work was supported by the excellence initiative of the German federal and state governments.

combining Gabor filter banks and gray level co-occurrence statistics of pixel pairs can be found in [25].

A pioneering approach to texture feature extraction, developed by Ojala et al., decreases the number of jHST entries by quantization of the pixel values in local patterns to only two values while keeping the full dimensionality of the jHST, leading to features called local binary patterns (LBPs) [26], [27]. To achieve invariance with respect to gray level offsets, the gray level of the central pixel is subtracted from all other gray levels in the local pattern prior to the binary quantization. This operation can be regarded as applying a bank of oriented difference filters [28]. For a second-order neighborhood, i.e., a neighborhood consisting of the eight nearest pixels to its center pixel, the binarized gray level differences can then be coded as an eight-bit word, thus elegantly allowing to represent the 8D-jHST as a 1D histogram with 256 bins. If, for larger neighborhoods, the number of LBP values grows too large, they may be rebinned such that only a lower number of prominent LBP values are distinguished, while only using a few bins for the others [27]. A certain degree of rotation invariance is achieved by cyclic shifts of the LBPs. Genuine rotational invariance, however, then holds only for rotations by some integer multiples of $2\pi/Q$, where Q is the number of pixels in the neighborhood. Modifications of LBPs include thresholding to increase robustness against noise in flat regions [29], making LBPs center-symmetric [30] for region description, and combining LBPs with co-occurrence matrices [31]. An alternative approach [32], [33] uses vector quantization [34] of the QD-jHSTs to reduce their size, which may be regarded as a special instance of signatures [35]. Compared to the complete jPDF of non-trivial neighborhoods, both the GLCM approach and the LBP approach result in a loss of information.

We describe here a framework for supervised texture classification which avoids the dilemma of either considering only very low-dimensional jHSTs or applying very coarse quantization by modelling the jPDFs of local patterns by Gaussian mixture models (GMMs) [36], [37]. We show that the parameters of the GMMs can be reliably estimated, even from small image regions, by the expectation-maximization (EM) algorithm [38]. The texture features are then formed by the jPDF estimates, or, more precisely, by their GMM parameters. For classification, distances between the thus modelled jPDFs can be computed in closed form from their model parameters. Using the – rather elementary – oriented LBP difference filters leads to a descriptor related to the LBP which we will refer to as the local continuous patterns (LCP) descriptor.

While results show that the LCP descriptor outperforms the LBP method, it does not reach the performance of a more recent approach by Lazebnik et al. [39]. In their approach, Lazebnik et al. model the jPDF of the filter outputs by signatures, which consist of cluster centers in the filter output space and associated weights. Furthermore, they use histograms as non-parametric density estimators. Signatures are compared for similarity by the well-known Earth Mover's Distance (EMD) [40], while the histograms are compared by the χ^2-distance.

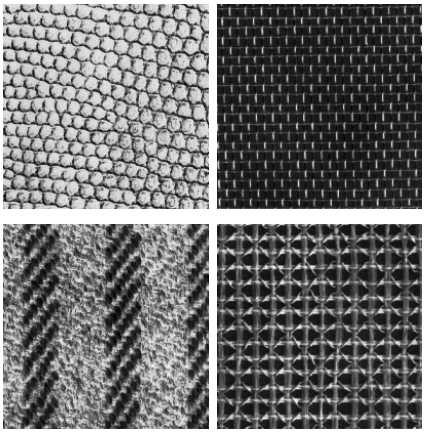


Fig. 1. Examples of Brodatz textures (200×200 pixels). From left to right, top to bottom: D3, D6, D11, D52.

Our above approach is comparable to the one of Lazebnik et al. as both subject the region of interest to a filter bank and estimate the jPDF of the filter outputs. Later on in this paper, after having replaced quantization and histogram computation of the LBP method by the GMM-based jPDF estimator, we will therefore also replace the simple LBP difference filters by more sophisticated filters, namely, the wavelet frame transform (WFT). The thus constructed texture descriptor, consisting of a filtering step by a WFT filter bank and subsequent GMM density estimation, is evaluated by classifying the entire Brodatz set (111 textures) in a setting as used in [39].

The paper is structured as follows. In section II, we review the LBP approach in more detail. Section III is devoted to rotational invariance. We then develop the models for the gray level differences observed in local patterns, and describe how the parameters are estimated (section IV). Section V shortly reviews the WFT. In section VI, we discuss distance measures. In section VII, we turn to classifiers, viz., k-nearest neighbor (k-NN) classifiers and support vector machines (SVM) that use a kernel derived from the aforementioned distance function. In section VIII, we provide results and comparisons between the described classification approach and the methods discussed above, viz., LBPs, the method by Lazebnik et al., and to the recent Gabor-Filter-LBPs [41]. The evaluations and comparisons are based on the Brodatz texture set [42] in setups as employed in [33], [27] and [41] for the LBPs, and as described in [39] for signatures. We conclude with a discussion in section IX.

Throughout the remainder, and unless otherwise mentioned, we shall assume the following setup. There exists a set of textured images (Fig. 1), each of which is subdivided into smaller regions of, for instance, 64×64 pixels. Each such region is to be uniquely assigned to one of K texture classes.

II. Local Binary Patterns

LBPs as devised by Ojala et al. [26], [27] consider the gray levels y_1 to y_Q of the Q pixels on a circle of radius R around the central pixel to be processed, as illustrated for $R = 1$

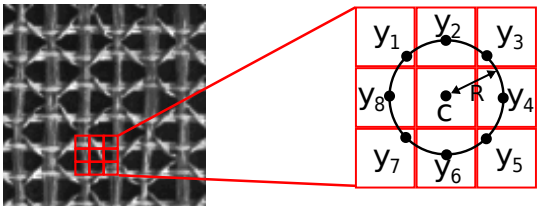


Fig. 2. Illustration of LBP formation for $R = 1$ and $Q = 8$.

and $Q = 8$ in Fig. 2. From each y_i, $i = 1, ..., Q$, the gray level c of the central pixel is subtracted, yielding the feature vector $z = (z_1, ..., z_Q) = (y_1 - c, ..., y_Q - c)$. Computing the differences can be regarded as convolution with the oriented difference filters

$$h_1^{LBP} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \ldots, h_8^{LBP} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} . \quad (1)$$

The differences z_i, $i = 1, \ldots, Q$ are then quantized to two levels, and converted into a scalar value according to

$$LBP_{Q,R} = \sum_{i=1}^{Q} \mathbb{I}_{\{z_i > 0\}} \cdot 2^{i-1} \quad (2)$$

where $\mathbb{I}_b$ is the indicator function which evaluates to one if the Boolean expression b is true and to zero otherwise. While the binary quantization is rather coarse, it makes the LBPs invariant to gray level scaling. A certain degree of rotation invariance is achieved by cyclically shifting the binary representation $(\mathbb{I}_{\{z_1>0\}}, \ldots, \mathbb{I}_{\{z_Q>0\}})$ so that $LBP_{Q,R}$ becomes minimal, which corresponds to cyclically interchanging the axes of the QD-jHST. The QD-jHST can now be conveniently represented by the 1D histogram of the $LBP_{Q,R}$ as computed by (2) over one texture region. These histograms form the features upon which classification of the texture regions is based [27]. Distances between the histograms of LBPs can, for instance, be evaluated by the Kullback-Leibler divergence

$$d_{\mathrm{KL}}(H, G) = \sum_{n=1}^{N} h_n \log\left(\frac{h_n}{g_n}\right) \quad (3)$$

where h_n and g_n are the bins of the histograms H and G to be compared. While LBPs constitute thus a significant step forward towards consideration of the jPDF with all its dimensions, the coarse quantization omits gray scale information which may well be beneficial for classification (although it may provide an additional degree of invariance to gray level scaling); indeed, Ojala et al. propose to combine the LBP histograms with variance histograms of the same region in [27] to improve recognition performance. Secondly, shifting of the binary representation is limited to shifts that correspond to rotations of the texture by full pixel positions. For $Q = 8$, rotation invariance is reliably achieved only for angles of $2\pi i/8$ $(i = 0, \ldots, 7)$. In the next section, we will treat a refinement of rotation invariance. We then show how to model the jPDFs of observed local patterns without additional quantization, and how the model parameters can be estimated from the texture regions.

III. Rotation Invariance

To achieve rotation invariance, interpolation between the Q elements of $z = (z_1, \ldots, z_Q)$ — which can be interpreted as a sampled version of the continuous circle around the center pixel — is required, as well as the determination of a defined starting point on the arc of the circle . Towards the end of the latter, and because of the circular LBP support, the elements $z_1, \ldots, z_Q$ are periodically repeated. By piecewise cubic interpolation (cf. [43], [44]), a continuous function $x_{\mathrm{cont}}(\beta)$ (with $\beta \in \mathbb{R}$ continuous) extending z around its sampling points $x_{\mathrm{cont}}(n) = z_{(n \bmod Q)}$ for $n \in \mathbb{N}$ is then computed. If the texture region is rotated arbitrarily, each pixel on the circle will produce a very similar (possibly equal) x'_{cont} that is a circularly shifted version of x_{cont}. All that remains now is to define one period of x_{cont} that is unique regardless of the shift. This period is defined by its starting point $s_{\mathrm{start}} \in [0, Q)$, and the period length Q. We define the starting point s_{start} such that the integral over the first half of the period is maximized, i.e., such that

$$s_{\mathrm{start}} = \underset{s \in [0,Q)}{\arg\max} \left\{ \int_s^{s+\frac{Q}{2}} x_{\mathrm{cont}}(\beta) d\beta \right\} \quad (4)$$

where the evaluation of (4) for each candidate s is implemented as convolution with a moving sum filter. To finally extract the rotation invariant feature vector, the extracted period is sampled, yielding $x = (x_1, \ldots, x_T)$ with $x_t = x_{\mathrm{cont}}(s_{\mathrm{start}} + \frac{tQ}{T})$, $t = 1, \ldots, T$. The number of extracted values T does not necessarily need to be equal to Q, though this may seem like a good value. Instead of infinitesimal sampling, an even higher degree of rotation invariance can be achieved by taking the average over a small neighborhood ϵ around $x_{\mathrm{cont}}(s_{\mathrm{start}} + \frac{tQ}{T})$ as the value for x_t, yielding $x_t = \frac{1}{2\epsilon} \int_{-\epsilon}^{\epsilon} x_{\mathrm{cont}}(s_{\mathrm{start}} + \frac{tQ}{T} + \gamma) d\gamma$. The process is illustrated in Fig. 3. Experimental results for rotation invariance are provided in section VIII-B.

IV. Local Texture Descriptors: Density Estimation

We first seek to overcome the coarse binarization of the LBP approach. Towards this end, we show that the computation of the one-dimensional histograms of LBP-values can be interpreted as computing a Q-dimensional joint histogram of the filter responses quantized merely to their signs. The binary representation $(\mathbb{I}_{\{z_1>0\}}, \ldots, \mathbb{I}_{\{z_Q>0\}})_2$, which is calculated in an intermediate step of the LBP value computation, is the bin index of the Q-dimensional jHST. In fact, the summation in equation (2) simply unfolds the Q-dimensional jHST to a one-dimensional histogram [28]. Hence, the LBP descriptor can be regarded as a filtering step by oriented difference filters, and a subsequent density estimation step by Q-dimensional jHST computation. The goal of this section is to develop a parametric density estimator using GMMs to avoid the binary quantization. To motivate GMMs, Fig. 4 (a) shows a 2D-jHST (or GLCM) which, for illustrative purposes, is not populated densely enough to form a reliable estimate. In

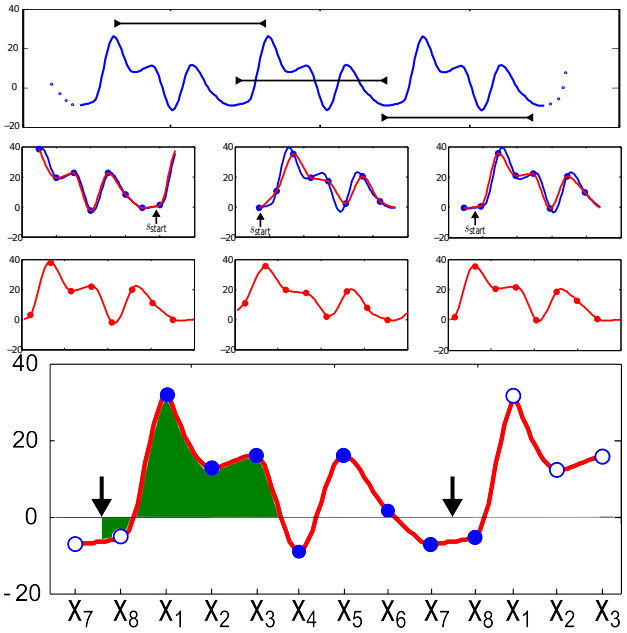


Fig. 3. Top: Illustration of gray levels on a continuous circle around a center pixel after periodic repetition. Middle row: Three randomly selected periods in blue (dark gray), each indicated by a horizontal bar in the top row, which correspond to rotations by arbitrary angles. Blue (dark) spots correspond to the sampling points $(z_1, \ldots, z_8)$ used to interpolate $x_{\mathrm{cont}}(\beta)$ shown in red (light gray). For each rotation, the automatically determined starting point s_{start} is also shown. Third row: Interpolated gray level curves after shifting each curve by s_{start}, and the resampled feature vector $(x_1, \ldots, x_8)$ as red (light gray) spots. The differences between the three curves are negligible, thus illustrating rotation invariance. Bottom: Area as computed in (4) shown shaded.

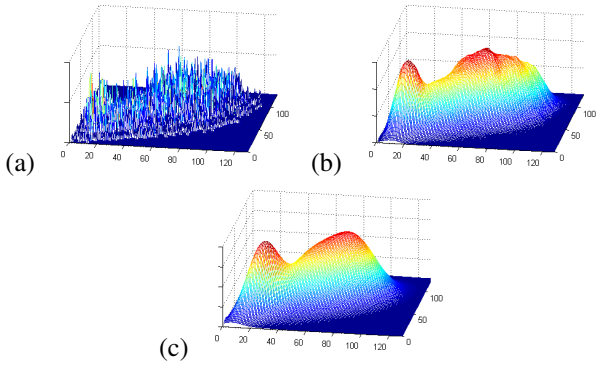


Fig. 4. Sparsely populated 2D-jHST (a), the Parzen estimate derived from it (b), and the GMM estimated from it (c).

Fig. 4 (b), a Parzen estimate obtained by a Gaussian window from the 2D-jHST is shown. While the estimate is smoothed, this approach requires huge computational efforts especially for more dimensions, as does the computation of distance functions between different estimates. Fig. 4 (c) shows an estimate of the jPDF obtained from the histogram in (a) by a sum of weighted normal distributions, or GMM.

Given a texture region P, we observe all local patterns z as defined above present in it. The observed patterns may be optionally made rotation invariant, yielding a realization x of a random vector X as described in section III. All observations are then assumed to obey the jPDF $p(X|P)$. We model $p(X|P)$ by a sum of weighted normal distributions according to

$$p(X|P) = \sum_{i=1}^{I} \alpha_{i|P} \, \mathcal{N}\left(X|\theta_{i|P}\right) \tag{5}$$

where $\mathcal{N}(X|\theta_{i|P})$ is a normal distribution with parameters $\theta_{i|P} = \left\{\mu_{i|P}, \Sigma_{i|P}\right\}$. The variable I denotes the number of Gaussian components in the mixture (sometimes referred to as modes). The coefficients $\alpha_{1|P}, \ldots, \alpha_{I|P}$ sum to unity. From the observed patterns x, the unknown parameters $\Theta_P = \{\theta_{i|P}, \alpha_{i|P} | i = 1, \ldots, I\}$ can now be estimated such that their likelihood given the data x is maximized. To ensure stable estimation even in the presence of relatively few samples, each covariance matrix is split into a diagonal matrix $\Sigma_{i,\mathrm{diag}|P}$ describing the variances for each Gaussian component $\mathcal{N}(X|\theta_{i|P})$ in (5), and a global full matrix $\Sigma_{\mathrm{glob}|P}$, which is identical for all summands of (5), and captures also covariances [45]. This yields

$$\Sigma_{i|P} = (1 - \lambda)\Sigma_{\mathrm{glob}|P} + \lambda\Sigma_{i,\mathrm{diag}|P}, \tag{6}$$

where the parameter λ is set to $\lambda = 0.5$. This smoothing technique is common practice to counteract overfitting when using Bayesian Normal classifiers (see [45]). A standard algorithm for solving the maximum likelihood estimation problem for GMMs, which we also use here, is the EM-algorithm [38], [46], [47].

V. Local Texture Descriptors: Filter Banks

Results in section VIII-A will confirm that the GMM-based density estimator provides indeed a tangible performance increase over LBPs. When applied to more than 100 texture classes as done in section VIII-C, the performance of the GMM-based density estimator together with the difference filters of (1) does, however, not quite reach the performance of more recent texture classification methods, specifically of [39]. In the following, we therefore show that replacing the simple yet computationally efficient difference filters of the LBPs by more sophisticated filter banks provides another increase of classification accuracy. We exemplarily chose WFT filter banks [48]. The wavelet frame transform corresponds to the regular Wavelet Transform with the exception that no dyadic downsampling is carried out in the subbands. Rather, the analysis filters are dilated accordingly, leaving the filter outputs at equal size at each decomposition level. The WFT thus avoids the shift-variant phenomena tied to downsampling in the dyadic wavelet transform [49], [50], and fits well into the concept of texture description by filtering and subsequent density estimation. In section VIII-C, we will employ two versions of the WFT, where one is based on coiflets, and the other one on the Haar wavelet. The Haar decomposition filters are given by $h_{low}^{haar} = [0.7071 \quad 0.7071]$ for the lowpass channel and $h_{hi}^{haar} = [-0.7071 \quad 0.7071]$ for the highpass channel, and the coiflets by $h_{low}^{coif} = [-0.0157 \quad -0.0727 \quad 0.3849 \quad 0.8526 \quad 0.3379 \quad -0.0727]$ and $h_{hi}^{coif} = [0.0727 \quad 0.3379 \quad -0.8526 \quad 0.3849 \quad 0.0727 \quad -0.0157]$ for lowpass and highpass channel, respectively.

VI. Distance Functions on GMMs

Once the GMMs are estimated for every texture region in the entire set, classification requires the comparison of GMMs for similarity. A well-established information theoretic distance measure is the Kullback-Leibler (KL) divergence [51], [52]. The KL divergence of two jPDFs $p(X|P_1)$ and $p(X|P_2)$ (associated with two regions P_1 and P_2) is defined as (cf. for discrete histograms (3)):

$$\mathrm{D}_{\mathrm{KL}}\left(p(\cdot|P_1), p(\cdot|P_2)\right) = \int_{-\infty}^{\infty} p(x|P_1) \log\left(\frac{p(x|P_1)}{p(x|P_2)}\right) dx. \tag{7}$$

The KL divergence is not a true distance metric, since it does not obey the triangle inequality which is the basis of fast search algorithms [53], such as successive elimination in motion estimation by block matching [54]. Moreover, it lacks symmetry, but a symmetrized version can be computed by the sum $\mathrm{D}_{s\,\mathrm{KL}}\left(p(\cdot|P_1), p(\cdot|P_2)\right) = \mathrm{D}_{\mathrm{KL}}\left(p(\cdot|P_1), p(\cdot|P_2)\right) + \mathrm{D}_{\mathrm{KL}}\left(p(\cdot|P_2), p(\cdot|P_1)\right)$, or by the minimum $\mathrm{D}_{s\,\mathrm{KL}}\left(p(\cdot|P_1), p(\cdot|P_2)\right) = \min\{\mathrm{D}_{\mathrm{KL}}\left(p(\cdot|P_1), p(\cdot|P_2)\right), \mathrm{D}_{\mathrm{KL}}\left(p(\cdot|P_2), p(\cdot|P_1)\right)\}$. The KL divergence has been shown to perform well on, e.g., histograms [55]. Its biggest drawback with respect to GMMs is that it cannot be computed in closed form from a GMM. Using Monte Carlo sampling [56], [57], which is known to converge to the KL divergence for sufficiently large sets of samples, is computationally prohibitive in this setting. A well-suited alternative is to use the normalized L2 distance, as done for genre classification in [53]. First, all GMMs have to be scaled to have unit L2 norm by

$$p'(X|P) = \frac{p(X|P)}{\|p(\cdot|P)\|_2}\,, \quad \|p(\cdot|P)\|_2 = \sqrt{\int_{-\infty}^{\infty} p^2(x|P)dx}. \tag{8}$$

The normalized L2 distance between the normalized GMMs of regions P_1 and P_2 is then defined as

$$\begin{aligned} & \mathrm{D}_{\mathrm{L2}}(p(\cdot|P_1), p(\cdot|P_2)) && (9)\\ = & \int_{-\infty}^{\infty} \left(p'(x|P_1) - p'(x|P_2)\right)^2 dx \\ = & \; 2\left(1 - \int_{-\infty}^{\infty} p'(x|P_1)\cdot p'(x|P_2)dx\right)\,. && (10) \end{aligned}$$

Note the equivalence between equation (10) and the correlation coefficient. As can also be seen from equation (10), $\mathrm{D}_{\mathrm{L2}}(\cdot,\cdot) \in [0, 2]$ holds. A distance of 2 is reached if the distributions do not overlap, i.e., if they are orthogonal. For Gaussian distributions, this is only reached in the limiting cases when variances approach zero or the Euclidean distance between each pair of mean vectors grows large. A distance of zero is reached if and only if the distributions are equal, i.e., for $p(X|P_1) = p(X|P_2)$. To solve equation (8) and (10) analytically the following integral has to be solved:

$$\begin{aligned} & \int_{-\infty}^{\infty} \left(\sum_{i=1}^{I_1} \alpha_{1i}\, \mathcal{N}\left(x|\theta_{1i}\right)\right)\left(\sum_{j=1}^{I_2} \alpha_{2j}\, \mathcal{N}\left(x|\theta_{2j}\right)\right) dx \\ = & \sum_{i=1}^{I_1}\sum_{j=1}^{I_2} \alpha_{1i}\alpha_{2j} \int_{-\infty}^{\infty} \mathcal{N}\left(x|\theta_{1i}\right)\mathcal{N}\left(x|\theta_{2j}\right) dx && (11) \end{aligned}$$

where, to simplify notation, α_{1i} stands for $\alpha_{i|P}$ with $P = P_1$, and so on. With $\mathcal{N}\left(X|\theta\right)$ being a normal distribution, the closed form solution for the last integral in (11) is

$$\begin{aligned} c_{ij} &= \int_{-\infty}^{\infty} \mathcal{N}\left(x|\theta_{1i}\right)\mathcal{N}\left(x|\theta_{2j}\right) dx && (12)\\ &= \frac{1}{\sqrt{\det\left(2\pi(\Sigma_{1i}+\Sigma_{2j})\right)}}\cdot \\ &\quad \exp\left(-\frac{1}{2}(\mu_{1i}-\mu_{2j})^T(\Sigma_{1i}+\Sigma_{2j})^{-1}(\mu_{1i}-\mu_{2j})\right). \end{aligned}$$

The complexity of the distance evaluation (10) is quadratic in the number of modes because of the double summation and cubic in the number of dimensions because of the matrix inversion $(\Sigma_{1i}+\Sigma_{2j})^{-1}$.

The final distance function is obtained by substituting (12) into (11) and then into (10), yielding

$$\begin{aligned} & \mathrm{D}_{\mathrm{L2}}\left(p(\cdot|P_1), p(\cdot|P_2)\right) && (13)\\ = & \; 2\left(1 - \frac{1}{\|p(\cdot|P_1)\|_2\|p(\cdot|P_2)\|_2}\sum_{i=1}^{I_1}\sum_{j=1}^{I_2}\alpha_{1i}\alpha_{2j}c_{ij}\right)\,. \end{aligned}$$

VII. Classification

This section deals with assigning class labels to the texture regions based on the estimated GMMs. We review the k-nearest-neighbor (k-NN) classifier [58], [59] and support vector machines (SVMs) briefly. To evaluate the performance of the texture classifiers, the set of all regions is subdivided into a training set and a test set, which are kept strictly disjoint. The former is used to train a classifier, such as the SVM, which is then evaluated on the test set. The class labels of the test patterns are only used for the calculation of the classification error.

A. k-Nearest-Neighbor Classifier

Given all (normalized) GMMs of the training set $\{p'(\cdot|P_1), \ldots, p'(\cdot|P_N)\}$ and a single GMM $p'(\cdot|P^*)$ extracted from a region P^* of the test set, the k-NN classifier compares $p'(\cdot|P^*)$ to every $p'(\cdot|P_n)$ $(n = 1, \ldots, N)$ by means of the distance measure (10), which, for GMMs, simplifies to (13). The class membership of the k closest mixture distributions of the training set are used to "vote" for a class label that is then assigned to $p'(\cdot|P^*)$, or more precisely, to the texture region P^*. For instance, if $k = 1$, the class label of the most similar training pattern is assigned to the test pattern under investigation. Note that the computational complexity of the k-NN classifier rises linearly with the size of the training set. As the training set grows large, the k-NN classifier may become infeasible, though methods to speed up the search process exist, for instance by exploiting the triangle inequality [53].

B. Support Vector Machines (SVMs)

SVMs are typical two class classifiers, that have gained large popularity in the last decade [60]. After a training phase a new sample x is assigned to the class

$$r(x) = \operatorname{sgn}\left(\sum_{n\in S} \alpha_n t_n K(x, x_n) + b\right) \tag{14}$$

with $\mathcal{S}$ being the set of support vector indicies. x_n are the training samples. Extending distance functions to kernel functions is a well-known method to create problem specific SVMs [61], [62]. In the case of texture classification by means of GMMs, we have used the following kernel function which exhibits some similarity to the Gaussian kernel:

$$K(p'(\cdot|P_n), p'(\cdot|P_m)) = \exp\left(\frac{-\,\mathrm{D_{L2}}(p'(\cdot|P_n), p'(\cdot|P_m))}{\gamma} + B\right) \tag{15}$$

where B and γ are parameters that can be tuned to optimize recognition performance. The parameter values were empirically determined and set to $B = 10$ and $\gamma = 12$ for all experiments below.

VIII. Experimental Results

We evaluated our method using the Brodatz test set [42], where we first followed the experimental setup of [33] and [27]. The Brodatz test set consists of textured images of size 640×640 pixels representing one texture class each, such as wood, water, rocks, fur etc. Examples are illustrated in Fig. 1. We then evaluate the rotational invariance within the same experimental setup as in [41] (section VIII-B). We finally evaluate the GMM-based density estimator combined with the WFT for both the coiflet and the Haar wavelet in the same experimental setup as used in [39].

A. Experiment 1 - Large Number of Classes

This set of experiments shall show the performance of our texture classification method when large numbers of different classes are present. Training and test set are constructed from the Brodatz source images as follows: First, the images were downsampled by factor of 2.5 and then subdivided into 16 regions of size 64×64 pixels. Half of these regions were taken as training samples while the other half contributed to the test set, leaving eight regions per class in both training and test set. We did not distribute the regions randomly as in [33], instead, we deterministically used the left half of each source image as training material, thus leaving the right half to the test set. This way of dividing into training and test set means that both training and test set are formed from spatially adjacent regions. The training set does thus not reflect small variations over space which commonly occur in real life data. We believe this to be a more realistic scenario making the classification task more difficult. No further preprocessing was performed.

To assess the robustness of the classifier for different numbers of classes, we determined the relative error for classifying the samples in the test set for 16, 32, 48, 64 and 80 classes respectively. The texture classes used in each case are given in Appendix A. The classification results are summarized in table I.

Num. of Classes	basis system	k-NN k=3	k-NN leave-1-out	SVM (1-vs-all)	LBP codebook
16	0.78	1.56	0.0	0.78	n.a.
24	0.52	1.56	0.0	0.52	n.a.
32	0.39	1.56	0.0	0.39	3.2 [33]
48	1.56	2.6	0.52	1.3	n.a.
64	3.3	3.97	0.39	2.5	n.a.
80	6.56	8.13	1.64	5.3	n.a.

TABLE I
RELATIVE CLASSIFICATION ERROR IN %

, We selected one setup of the classifier as a basis system that parameter changes can be compared to. The basis system uses $I = 8$ Gaussians in (5) and a neighborhood of $Q = 8$ pixels (see Fig. 2). The classifier of the basis system is a k-nearest-neighbor classifier with $k = 1$. The exact number I of Gaussians used in (5) is not critical, as changing I to six or twelve Gaussians did neither improve nor degrade the recognition performance (though it had an impact on the computational load). Addressing how to identify the number of Gaussians in general is beyond the scope of this paper, a top-down approach can be found in [63]. Increasing the size of the neighborhood to sixteen pixels, i.e., $R = 2, Q = 16$, made results significantly worse. This can be explained by the fact that too few samples were used to reliably estimate all parameters. To evaluate another classifier on the basis setup a Support Vector Machine was trained and tested within a one-against-all setup which turned out to be slightly superior to the one-against-one setup. We have used the SVM implementation of [64]. Finally, the k-NN classifier was tested by the leave-one-out method because of the small training sets. The relative classification error was as low as 0.39 % for 64 classes. Since there is no rotation invariance needed in this setup, the difference vectors are not shifted.

As expected the error increases once a certain number of classes is exceeded. Note, though, that even in the 80 class setup there are still only eight training regions per class. Also, some of the classes, e.g., D3, D22 and D35, are visually very similar. These are the ones which cause the observed confusions. Note moreover that the complete error rate in the 16, 24 and 32 class setup is caused by a single confusion between the classes shown in Fig. 5. Comparing classification accuracy in the 32 class setup to the results for LBPs of [33] shows that the new method achieves lower error rates even though regions are not distributed randomly into learning and test set. On 32 classes, the standard LBP approach reaches 8.8 % error, while the LBP-codebook approach — also reported in [33] — achieves a recognition error of 3.2 %.

B. Experiment 2 - Rotation Invariance

This experiment evaluates the rotation invariance of the described method. It is based on the exact same data used in

method	training angle									
	0°	20°	30°	45°	60°	70°	90°	120°	135°	150°
$LBP_{16,2}$	3.8	1.0	1.4	1.1	1.5	0.9	2.4	1.4	1.3	2.5
$LBP_{8,1+16,2+24,3}/VAR_{8,1+16,2+24,3}$	0.0	0.3	0.5	0.2	0.4	0.3	0.2	0.4	0.2	0.1
basis system ($I = 8, Q = 8$, 1-NN)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

TABLE II

CLASSIFICATION ERROR IN %

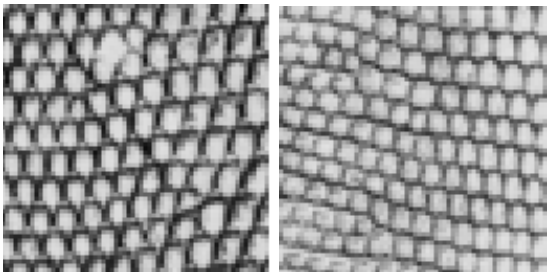


Fig. 5. The only confusion occurring in the 16, 24 and 32 class setup. Left: D22, Right: D3.

[27] in order to allow comparing the results on a one-to-one basis. The same benchmark set has also been used recently for comparison in [41] who use dominant local binary patterns in conjunction with circularly symmetric Gabor filter responses, achieving an average error rate of 0.2 %. The experiments consist of ten runs. Each run uses a given training set and classifies a test set of sixteen classes. Test and training sets are kept disjoint. Each class consists of eight textured images of size 180×180 pixels at rotation angle 0°. Furthermore, versions rotated by angles of 20°, 30°, 45°, 60°, 70°, 90°, 120°, 135°, 150° were generated. Only one image per class is used as a training sample. This training image is subdivided into a 11×11 grid of 16×16 pixel sized training regions totalling in 121 training regions per class. The remaining seven texture images rotated by the remaining nine angles are used as test samples, thus totalling in 63 test images per class. Note that the test samples are still of size 180×180 pixels while the training samples are of size 16×16. During the training phase the training samples are combined into one GMM for each class in order to numerically reliably estimate the parameters, in a manner analog to [27] where all training samples for each class where collected into one histogram. To extract the GMMs, a sixteen pixel neighborhood of radius $R = 2$ was used. Regions are classified by the k-NN classifier with $k = 1$. Table II shows results obtained by LBPs for the same neighborhood setting as well as the best results obtained by LBPs on that particular set in which three different neighborhoods are combined with additional variance measures. Our approach classified all testing samples correctly in all cases, yielding an average classification error of 0 %.

C. Experiment 3 - WFT Filter Banks

The experiments of sections VIII-A and VIII-B confirm that GMM-based density estimation indeed results in a tangible performance increase compared to the LBP approach. To evaluate our framework on an even larger number of classes, and to compare to a recent state-of-the-art classification method, we apply our descriptor within the same experimental setup as in [39]. The entire Brodatz test set (consisting of 111 texture classes) is separated into a training and test set as follows. Each source image (640×640 pixels) is divided into 3×3 non-overlapping image regions of equal size all belonging to one class. N samples of each class contribute to the training set while the remaining $9 - N$ samples form the test set with $N = 1, \ldots, 8$. The partitions are furthermore computed randomly and classification results are averaged over 1000 trial runs. Note that, in [39], all results are always for $N = 3$.

The reference result for using $N = 3$ training samples per class taken from [39] is an error rate of 11.4 %. Testing the descriptor using the LBP difference filter bank of equation (1) achieves an error rate of 18.1 % on the very same task. It is not surprising that a simple filter bank with such small support area is inferior to the results of Lazebnik et al. Therefore we have replaced the difference filters by a WFT filter bank as detailed in section V. The number of decomposition levels of the Wavelet Frame Transform is four. Classification results for various numbers of Gaussian components (I in equation (5)) and choices of wavelets in the WFT are summarized in Table III. Best classification results are obtained by using 16 modes and a Haar wavelet, achieving an error rate of 10.1 % for $N = 3$ training samples per class. The repetitive sampling of training and test sets coupled with only few training samples per class (as low as $N = 1$, which is often referred to as retrieval rather than classification) mitigates effects of potential overfitting. The comparable performance of coiflets and Haar wavelets indicates that the decision to use a hierarchical decomposition architecture such as the WFT has a stronger influence on the classification results than the choice of a specific wavelet filter. Accordingly, the classification results depend also on the number of decomposition levels of the WFT. Using too low a number of decomposition levels may result in the loss of critical texture characteristics. On the other hand, if increased beyond a certain number, the lowpass channels will contain no more structural information but rather only local averages. This increases the dimensionality of the feature vector without providing additional relevant information for the classification, thus making it in turn more likely for the EM algorithm to converge to a poor local optimum.

N	12 Modes coiflet	16 Modes coiflet	16 Modes haar wavelet	20 Modes haar wavelet	Lazebnik [39]	LCP
1	19.3	19.5	19.4	19.4	n.a.	30.7
2	13.2	13.1	13.2	12.2	n.a.	22.6
3	10.5	10.4	**10.1**	10.3	11.4	18.1
4	8.9	8.9	8.3	8.5	n.a.	15.1
5	7.7	7.7	7.5	7.6	n.a.	13.4
6	6.9	7.1	6.4	6.6	n.a.	12.0
7	6.3	6.5	5.7	5.9	n.a.	11.2
8	5.7	6.2	5.3	5.6	n.a.	9.6

TABLE III

RELATIVE CLASSIFICATION ERROR IN % ON 111 BRODATZ CLASSES FOR VARYING NUMBER OF TRAINING SAMPLES ($N = 1, \ldots, 8$)

IX. DISCUSSION AND CONCLUSIONS

We have developed a framework for texture classification which filters a given texture region by a set of filters and subsequently estimates the jPDF by GMMs. Using the oriented difference filters of the LBP method, we could show that this framework avoids the quantization errors of the LBP method, and that its classification performance is better than the one of LBPs. Another performance gain of the GMM-based density estimator was obtained when the elementary LBP difference filters were replaced by WFT filter banks. Once the GMM parameters are calculated, distances between the estimated jPDF models of different textures can be compared by the normalized L2 distance, which can be computed in closed form from the parameters. Computation of the local difference patterns by filtering according to (1) followed by parametric modeling of the jPDF is thus consistent with what is stated in [65], viz., that summarizing texture appearance consists of filtering the image data and approximating the distribution of the filter outputs [65, p.53]. Ample experimental results on the Brodatz test set indicate a high recognition rate for both the k-NN classifier and the SVM. For 64 classes with separate training and test sets, the error rate was 2.5 % for the SVM. With larger training sets in a leave-one-out scenario, the error even fell to only 1.64 % for the k-NN and 80 classes. Extending the test scenario to all 111 Brodatz classes, our descriptor achieved an error rate of 10.1 % for $N = 3$ and 5.3 % for $N = 8$ training samples per class. The framework works still reliably when the textures are rotated. These results also show that, while one might argue that parametric modelling of the jPDFs basically may involve potentially erroneous assumptions [27, p.972], modelling by Gaussian mixtures is sufficiently flexible to outweigh such reservations.

Let us also briefly comment on computational costs. Consisting mainly of simple difference filters and quantization, the standard LBP descriptor can be computed efficiently. Classification, however, may be time-consuming as it involves distance computations using the Kullback-Leibler distance. In our approach, the most costly part is the estimation of the GMM parameters by the EM algorithm. Classification then requires analytical distance computations using the normalized L_2-distance. In an example run using Matlab-implementations of both the LBPs and the LCPs, the breakdown of computation times was as follows (64 classes, $Q = 8$, $I = 8$, k-nn classification with $k = 1$): Filtering 50s for both LBP and LCP, model computation 45s (LBP) vs. 133s (LCP), and classification 2419s (LBP) vs. 661s (LCP).

The starting point of the algorithms developed in this paper were LBPs. As these do not account for viewpoint invariances, we have not considered such invariances here, either. Achieving such "representational-level invariance" ([39, p. 1271]) often involves nonlinear steps in the filtering stage, such as feature detection [39, p. 1271], or determining the maximum response over several filters [66, Section 2.1.3]. As the primary goal of this contribution is to circumvent the curse of dimensionality using GMM-based density estimation, extensions towards such nonlinear steps are beyond the scope of this paper. We consequently restricted the evaluations to the Brodatz texture base which exhibits considerable interclass variability, but no geometrically transformed members of the same class (cf. [39, p. 1266, p. 1273]). Although consequently not as versatile as, for instance, the method of [66], our framework is applicable to a wide variety of classification problems, such as industrial inspection, where illumination and image acquisition can be controlled, making the possible number of classes the more challenging problem [67]. Achieving such representational-level invariance in our framework is, though, an interesting future topic. Other plans include to replace the k-NN by its fuzzy counterpart [68] (cf. also the codebook with ambiguity in [69]), which has shown to perform well in industrial inspection tasks such as [70]. Moreover, the investigation of yet other density estimators within this framework seems worthwhile. Finally, we plan to extend our framework towards unsupervised classification.

X. APPENDIX

A. Brodatz Classes Used in the Experiments

The Brodatz album consists of 111 texture classes. We have chosen test sets of varying size which are true subsets of the subsequent larger sets.

1) 16 Classes: In the 16 class setup the following Brodatz textures have been used: D1, D11, D18, D20, D21, D22, D3, D34, D4, D56, D6, D64, D66, D68, D79, D87.

2) 24 Classes: Additionally to the 16 classes the following textures have been used: D17, D28, D33, D37, D46, D49, D8, D9.

3) 32 Classes: Additionally to the 24 classes the following textures have been used: D101, D102, D103, D52, D75, D82, D94, D95.

4) 48 Classes: Additionally to the 32 classes the following textures have been used: D10, D104, D106, D109, D110, D111, D15, D25, D26, D32, D35, D47, D57, D65, D74, D93.

5) 64 Classes: Additionally to the 48 classes the following textures have been used: D112, D16, D19, D29, D48, D5, D53, D55, D67, D76, D77, D80, D81, D83, D84, D85.

6) 80 Classes: Additionally to the 64 classes the following textures have been used: D100, D107, D108, D12, D13, D2, D24, D36, D38, D60, D7, D73, D78, D86, D92, D98.

REFERENCES

[1] R. M. Haralick, "Statistical and structural approaches to texture," Proceedings of the IEEE, vol. 67, pp. 786–804, 1979.

[2] M. Pietikaïnen, A. Rosenfeld, and L. S. Davis, "Experiments with texture classification using averages of local pattern matches," IEEE Transactions on Systems, Man, and Cybernetics, vol. 13, no. 3, pp. 421–426, 1983.

[3] M. Unser, "Local linear transforms for texture measurements," Signal Processing, vol. 11, pp. 61–79, 1986.

[4] S. Grigorescu and N. Petkov, "Texture analysis using Renyi's generalized entropies," in Proc. IEEE Int. Conf. on Image Processing (ICIP) 2003, vol. I. IEEE, 2003, pp. 241–244, CD-ROM, ISBN 0-7803-7751-6.

[5] K. I. Laws, "Textured image segmentation," Ph.D. dissertation, Image Processing Institute, Univ. of Southern California, 1980, rept. 940.

[6] T. Aach, R. Mester, and U. Franke, "From texture energy measures to quadrature filter pairs – A system theoretical view of texture feature extraction," in Signal Processing IV: Theories and Applications, J. L. Lacoume, A. Chehikian, N. Martin, and J. Malbos, Eds., EUSIPCO 88. Grenoble, France: Elsevier Science Publishers, September 1988, pp. 827–830.

[7] T. Aach, A. Kaup, and R. Mester, "On texture analysis: Local energy transforms versus quadrature filters," Signal Processing, vol. 45, no. 2, pp. 173–181, 1995.

[8] A. N. Akansu and R. A. Haddad, "On asymmetrical performance of discrete cosine transform," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 38, no. 1, pp. 154–156, 1990.

[9] ——, Multiresolution Signal Decomposition. Boston: Academic Press, 2001.

[10] F. Ade, "Characterization of textures by 'eigenfilters'," Signal Processing, vol. 5, pp. 451–457, 1983.

[11] H. Knutsson and G. H. Granlund, "Texture analysis using two-dimensional quadrature filters," in IEEE Workshop on Computer Architecture for Pattern Analysis and Image Data Base Management. Pasadena, California: IEEE, October 1983.

[12] G. H. Granlund and H. Knutsson, Signal Processing for Computer Vision. Dordrecht: Kluwer, 1995.

[13] D. Gabor, "Theory of communication," Journal of the IEE, vol. 93, pp. 429–457, 1946.

[14] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," Journal Optical Society of America, vol. 2, no. 7, pp. 1160–1169, 1985.

[15] A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel texture analysis using localized spatial filters," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 1, pp. 55–73, 1990.

[16] A. C. Bovik, "Analysis of multichannel narrow-band filters for image texture segmentation," IEEE Transactions on Signal Processing, vol. 39, no. 9, pp. 2025–2043, 1991.

[17] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," Pattern Recognition, vol. 24, no. 12, pp. 1167–1186, 1991.

[18] S. E. Grigorescu, N. Petkov, and P. Kruizinga, "Comparison of texture features based on gabor filters," IEEE Transactions on Image Processing, vol. 10, pp. 1160–1167, 2002.

[19] M. Unser, "Texture classification and segmentation using wavelet frames," IEEE Transactions on Image Processing, vol. 4, no. 11, pp. 1549–1560, 1995.

[20] T. Randen and J. H. Husoy, "Texture segmentation using filters with optimized energy separation," IEEE Transactions on Image Processing, vol. 8, no. 4, pp. 571–582, 1999.

[21] X. Liu and D. Wang, "Texture classification using spectral histograms," IEEE Transactions on Image Processing, vol. 12, no. 6, pp. 661–670, 2003.

[22] T. Randen and J. H. Husoy, "Filtering for texture classification: A comparative study," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 4, pp. 291–309, 1999.

[23] A. Baraldi and F. Parmiggiani, "An investigation of the textural characteristics associated with gray level cooccurrence matrix statistical parameters," IEEE Transactions on Geoscience and Remote Sensing, vol. 33, no. 2, pp. 293–304, 1995.

[24] M. Unser, "Sum and difference histograms for texture classification," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 1, pp. 118–125, 1986.

[25] D. A. Clausi and H. Deng, "Design-based texture feature extraction fusion using Gabor filters and co-occurrence probabilities," IEEE Transactions on Image Processing, vol. 14, no. 7, pp. 925–936, 2005.

[26] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," Pattern Recognition, vol. 29, no. 1, pp. 51–59, 1996.

[27] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 971–987, 2002.

[28] T. Ahonen and M. Pietikaïnen, "Image description using joint distribution of filter bank responses," Pattern Recognition Letters, vol. 30, pp. 368–376, 2009.

[29] M. Heikkilä and M. Pietikainen, "A texture-based method for modeling the background and detecting moving objects," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 4, pp. 657–662, 2006.

[30] M. Heikkilä, M. Pietikainen, and C. Schmid, "Description of interest regions with local binary patterns," Pattern Recognition, vol. 42, pp. 425–436, 2009.

[31] L. Setia, A. Teynor, A. Halawani, and H. Burkhardt, "Image classification using cluster cooccurrence matrices of local relational features," Proceedings of the 8th ACM international workshop on Multimedia information retrieval, pp. 173–182, 2006.

[32] K. Valkealahti and E. Oja, "Reduced multidimensional co-occurrence histograms in texture classification," IEEE Transaction on Pattern Analysis and Machine Intellegence, 1998.

[33] T. Ojala, K. Valkealahti, E. Oja, and M. Pietikäinen, "Texture discrimination with multidimensional distributions of signed gray-level differences," Pattern Recognition, vol. 34, no. 3, pp. 727–739, 2001.

[34] T. Leung and J. Malik, "Representing and recognizing the visual appearence of materials using three-dimensional textons," International Journal of Computer Vision, vol. 43, no. 1, pp. 29–44, 2001.

[35] F. Serratosa and A. Sanfeliu, "Signatures versus histograms: Definitions, distances and algorithms," Pattern Recognition, vol. 39, no. 5, pp. 921–934, 2006.

[36] X. Xie and M. Mirmehdi, "Texems: Texture exemplars for defect detection on random textured surfaces," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, pp. 1454–1464, 2007.

[37] T. D. Rikert, M. J. Jones, and V. P., "A cluster-based statistical model for object detection," in Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, vol. 2, 1999, pp. 1046–1053.

[38] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," Journal Royal Statistical Society B, vol. 39, pp. 1–38, 1977.

[39] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine regions," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, pp. 1265–1278, 2005.

[40] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," International Journal of Computer Vision, vol. 40, pp. 99–121, 2000.

[41] S. Liao, M. W. Law, and A. C. Chung, "Dominant local binary patterns for texture classification," IEEE transactions on image processing, vol. 18, pp. 1107–1118, 2009.

[42] P. Brodatz, Textures: A Photographic Album for Artists and Designers. Dover, 1966.

[43] M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing: Part I — Theory," IEEE Transactions on Signal Processing, vol. 41, no. 2, pp. 821–833, 1993.

[44] ——, "B-spline signal processing: Part II — Efficient design and applications," IEEE Transactions on Signal Processing, vol. 41, no. 2, pp. 834–848, 1993.

[45] R. Duda, P. Hart, and D. Stork, Pattern Classification. Wiley-Interscience, 2000.

[46] R. Neal and G. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," Learning in Graphical Models, vol. 89, pp. 355–368, 1998.

[47] C. Biernacki, G. Celeux, and G. Govaert, "Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models," Computational Statistics and Data Analysis, vol. 41, no. 3-4, pp. 561–575, 2003.

[48] S. Li, J. T. Kwok, H. Zhu, and Y. Wang, "Texture classification using the support vector machines," Pattern Recognition, vol. 36, pp. 2883–2893, 2003.

[49] T. Aach, "Comparative analysis of shift variance and cyclostationarity in multirate filterbanks," IEEE Transactions on Circuits and Systems–I: Regular Papers, vol. 54, no. 5, pp. 1077–1087, 2007.

[50] T. Aach and H. Führ, "On bounds of shift variance in two-channel multirate filter banks," IEEE Transactions on Signal Processing, vol. 57, no. 11, pp. 4292–4303, 2009.

[51] S. Kullback and R. A. Leibler, "On information and sufficiency," Annals of Mathematical Statistics, vol. 22, no. 1, pp. 79–86, 1951.

[52] J. R. Hershey and P. A. Olsen, "Approximating the Kullback Leibler divergence between Gaussian mixture models," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 4, 15–20 April 2007, pp. IV–317–IV–320.

[53] J. H. Jensen, D. P. W. Ellis, M. G. Christensen, and S. H. Jensen, "Evaluation of distance measures between Gaussian mixture models of MFCCS," 2007.

[54] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," IEEE Transactions on Image Processing, vol. 4, no. 1, pp. 105–107, 1995.

[55] J. Puzicha, Y. Rubner, C. Tomasi, and J. Buhmann, "Empirical evaluation of dissimilarity measures for color and texture," in Proceedings of the International Conference on Computer Vision, vol. 2, 1999, pp. 1165–1173.

[56] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan, "An introduction to MCMC for machine learning," Machine Learning, vol. 50, no. 1, pp. 5–43, 2003.

[57] J. Goldberger, H. K. Greenspan, and J. Dreyfuss, "Simplifying mixture models using the unscented transform," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 8, pp. 1496–1502, 2008.

[58] C. W. Therrien, Decision, Estimation, and Classification. New York: John Wiley, 1989.

[59] K. Fukunaga, Introduction to Statistical Pattern Recognition. New York: Academic Press, 1972.

[60] N. Cristianini and J. Shawe-Taylor, An introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press New York, NY, USA, 1999.

[61] P. Moreno, P. Ho, and N. Vasconcelos, "A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications," Advances in Neural Information Processing Systems, vol. 16, 2004.

[62] O. Chapelle, P. Haffner, and V. Vapnik, "Support vector machines for histogram-based image classification," Neural Networks, IEEE Transactions on, vol. 10, no. 5, pp. 1055–1064, 1999.

[63] D. Ververidis and C. Kotropoulos, "Gaussian mixture modeling by exploiting the Mahanalobis distance," IEEE Transactions on Signal Processing, vol. 56, no. 7, pp. 2797–2811, 2008.

[64] S. Canu, Y. Grandvalet, V. Guigue, and A. Rakotomamonjy, "SVM and kernel methods Matlab toolbox," 2005.

[65] M. Mellor, B. W. Hong, and M. Brady, "Locally rotation, contrast and scale invariant descriptors for texture analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 1, pp. 52–61, 2008.

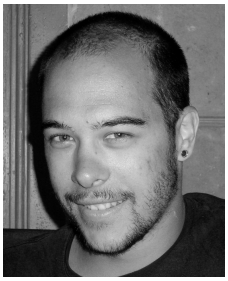
[66] M. Varma and A. Zisserman, "A statistical approach to texture classification from single images," International Journal of Computer Vision: Special Issue on Texture Analysis and Synthesis, vol. 62, no. 1–2, pp. 61–81, April 2005.

[67] A. Kumar, "Computer-vision-based fabric defect detection: A survey," IEEE Transactions on Industrial Electronics, vol. 55, pp. 348–363, 2008.

[68] J. M. Keller, M. R. Gray, and J. A. Givens, "A fuzzy K-nearest neighbor algorithm," IEEE Transactions on System, Man, and Cybernetics, vol. 15, no. 4, pp. 580–585, 1985.

[69] J. C. van Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. Smeulders, "Kernel codebooks for scene categorization," in Computer Vision – ECCV 2008, ser. LNCS, D. Forsyth, P. Torr, and A. Zisserman, Eds., vol. 5304. Springer, 2008, pp. 696–709.

[70] D. Toth, A. Condurache, and T. Aach, "A two-stage-classifier for defect classification in optical media inspection," in International Conference on Pattern Recognition, R. Kasturi, D. Laurendeau, and C. Suen, Eds. Quebec City, Canada: IEEE, August 11–15 2002, pp. 373–376.



Henning Lategahn studied Computer Science at RWTH Aachen University, Germany, where he received his Diploma degree (with honors) in 2008. Since 2009, he has been with the Department of Measurement and Control, Karlsruhe Institute of Technology, Germany. His research interests are in image analysis, specifically feature extraction, pattern recognition and visual SLAM.

Sebastian Gross studied electrical engineering with a focus on information and communication technology at RWTH Aachen University. Since receiving his diploma degree in 2007, he has been working as a research assistant at the Institute of Imaging and Computer Vision, RWTH Aachen University, towards a Doctoral Degree. His research interests are in medical image processing, pattern recognition, and computer vision. He is a student member of the IEEE.

Thomas Stehle studied Computer Science at Karlsruhe Institute of Technology, Germany, and received his Diploma degree in 2005. Since then he has been working as research scientist towards his PhD degree at the Institute of Imaging and Computer Vision, RWTH Aachen University, Germany. In 2007, he joined the IEEE as student member and in 2008 he joined the IEEE Computer Society. His research interests are medical image analysis and processing, motion estimation, camera calibration and computer vision.

Til Aach (M 1994 – SM 2002) received his Diploma and Doctoral degrees, both with honors in EE, from RWTH Aachen University in 1987 and 1993, respectively. While working towards his Doctoral Degree, he was a research scientist with the Institute for Communications Engineering, RWTH Aachen University, being in charge of several projects in image analysis, 3D-television and medical image processing. In 1984, he was an intern with Okuma Machinery Works Ltd., Nagoya, Japan. From 1993 to 1998, he was with Philips Research Labs, Aachen, Germany, where he was responsible for several projects in medical imaging, image processing and analysis. In 1996, he was also an independent lecturer with the University of Magdeburg, Germany. In 1998, he was appointed a Full Professor and Director of the Institute for Signal Processing, University of Luebeck. In 2004, he became Chairman of the Institute of Imaging and Computer Vision, RWTH Aachen University. His research interests are in medical and industrial image processing, signal processing, pattern recognition, and computer vision. He has authored or co-authored over 250 papers, and received several awards, among these the award of the German ”Informationstechnische Gesellschaft” (ITG/VDE), for a paper published in the IEEE Transactions on Image Processing in 1998. Dr. Aach is a co-inventor for about 20 patents. From 2002 to 2008, he was an Associate Editor of the IEEE Transactions on Image Processing. He was a Technical Program Co-Chair for the IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI) in 2000, 2002, 2004, and 2006. He is a member of the Bio-Imaging and Signal Processing Committee (BISP-TC) of the IEEE Signal Processing Society.