Chapter 2

Texture feature extraction

Experiments on cats and apes, which have visual systems similar to the human visual system [67], show that these animals decompose the visual images into several filtered images. The filtered images are tuned to different spatial frequency ranges and orientations [15]. Spatial filtering forms the basis for most of the texture segmentation schemes used in this dissertation. Figure 2.1[1] shows the general experimental setup. The focus of this dissertation will be on the filter, and a description of the principles of filtering for texture feature extraction/classification and segmentation is discussed in Section 2.1. A survey of past filtering approaches will be presented in Sections 2.2 and 2.3, while the major classes of non-filtering approaches are discussed briefly in Section 2.4. Some new approaches to texture filtering will be presented in Chapters 3, 4, and 5.

2.1 Principles of texture analysis by filtering

Before proceeding with the description of different filtering approaches, a brief description of the elements of the system in Figure 2.1 will be given. Consider the simple synthetic textured image in Figure 2.2(a). This image consists of two textures generated by sinusoids. The left half of the image has one low-frequency sinusoid and the right half has a high-frequency sinusoid superimposed on the low frequency one.

For illustrational purposes, consider a horizontal line through this image, Figure 2.2(b). The first operation of the system in Figure 2.1 is filtering. Assume that the image is filtered with a filter stopping the low frequency sinusoid and passing the high frequency sinusoid. A line in the resulting image is illustrated in Figure 2.2(c). In this case, we see that the filter response for the left texture has low energy and the right texture high. However, we can still not classify the image by its pixel values alone without significant classification errors.

Next, a local energy function is applied, consisting of a nonlinearity (Figure 2.2(d)), basically rectifying[2] the filter response, and smoothing (Figure 2.2(e)). The resulting feature image is given in Figure 2.2(f) and this feature image can be classified with success.

[1] The normalizing nonlinearity is rarely used, and is optional.

[2] Rectification is here understood as the operation of transforming both negative and positive amplitudes to positive amplitudes.

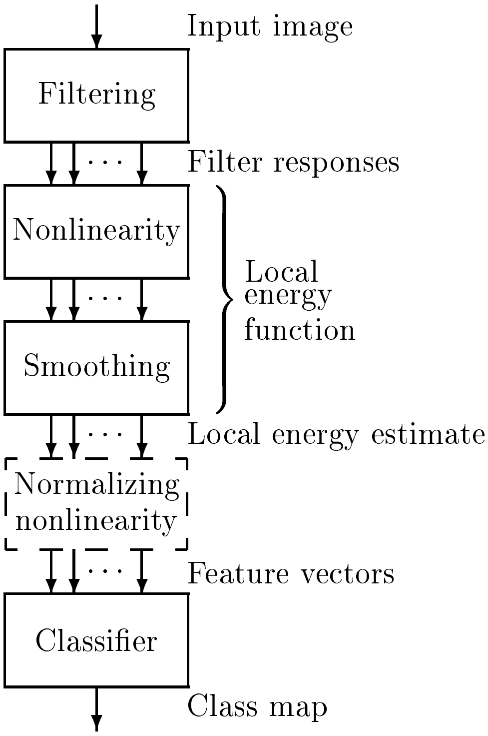


Figure 2.1: Experimental setup for most experiments in this dissertation. The dashed box is optional, see the text.

The optional second nonlinearity box in Figure 2.1 is not illustrated in this experiment. This nonlinearity is typically monotonic and it do only have effect when there is more than one feature image.

2.1.1 Local energy function and post processing

The objective of the local energy function, see Figures 2.1 and 2.2, is to estimate the energy in the filter output in a local region. Each filter is in essence a band-pass filter with frequency and orientation selective properties. The local energy function is utilized for the purpose of transmitting areas in each channel where the local pass band energy is strong into a high gray-level and vice versa. However, accurate edge preservation and accurate energy estimation are conflicting goals. High spatial resolution is required for accurate edge localization, while high spatial frequency resolution is required for accurate energy estimation.

Several smoothing filters are possible, and the Gaussian low-pass filter is one candidate. The Gaussian low-pass filter, which is separable and given by

$$h_G(n) = \frac{1}{\sqrt{2\pi}\sigma_s} e^{-\frac{1}{2}\frac{n^2}{\sigma_s^2}} \tag{2.1}$$

in each dimension, has joint optimum resolution in the spatial and spatial frequency domains [36, 26]. The Gaussian low-pass filter has infinite unit pulse response, thus it is

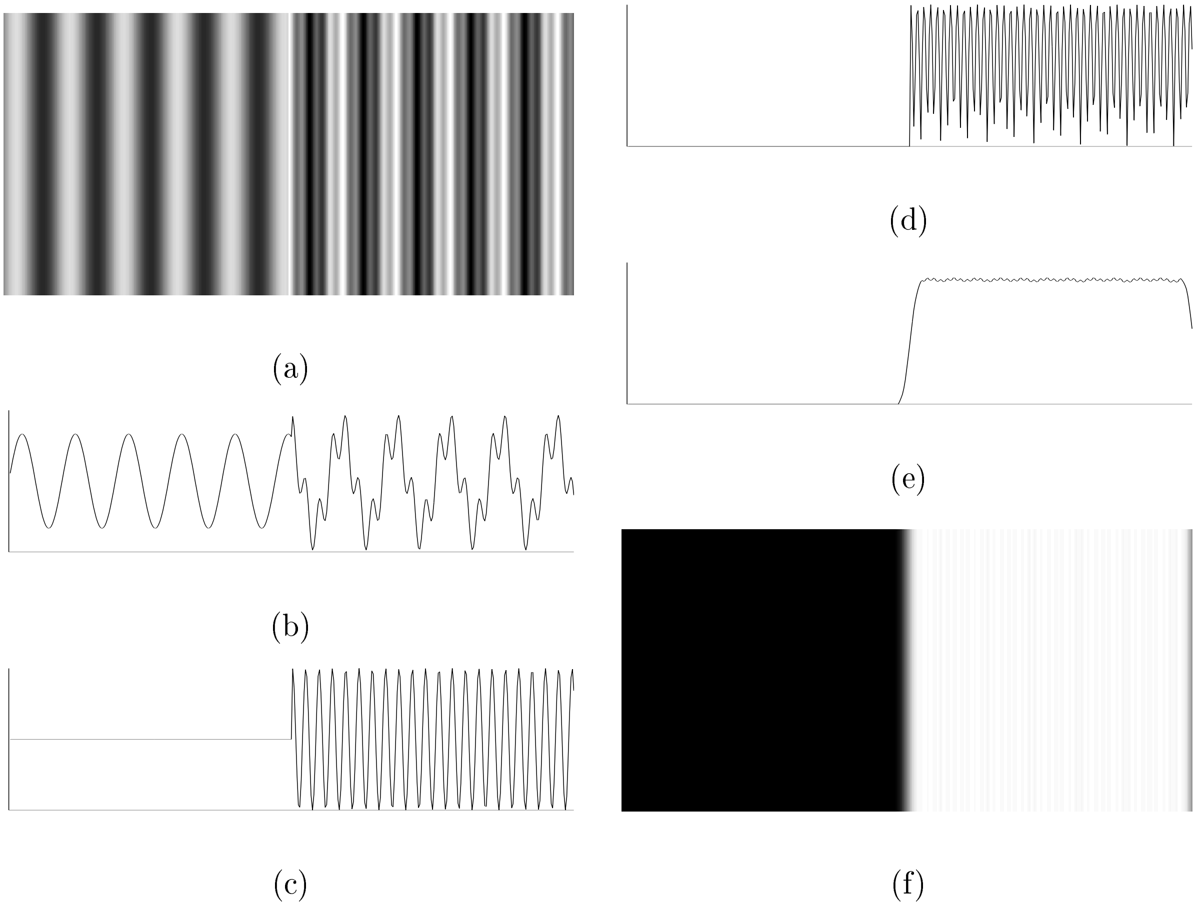


Figure 2.2: Illustration of a typical texture feature extraction process with filters. (a) Two synthetic textures, (b) a horizontal scan line through the image, (c) filtered, (d) nonlinearly transformed, (e) smoothed, and (f) the resulting 2D feature image.

not realizable. However, in all experiments it is realized by a finite approximation. Where not otherwise noted, this smoothing filter is used for the smoothing operations throughout this dissertation.

How do we determine the size of the smoothing filter? If we want to estimate the local energy of a signal with low spatial frequency, the smoothing filter must have a wide unit pulse response. On the other hand, we can allow narrower smoothing filters for higher frequencies. Hence, for a band-limited filter output, we may set the smoothing filter size as a function of, for instance, the band center frequency. With f_0 being the normalized center frequency (in the range $-\frac{1}{2} \leq f_0 \leq \frac{1}{2}$), Jain and Farrokhnia [50] suggested

$$\sigma_s = \frac{1}{2\sqrt{2}f_0}. \tag{2.2}$$

Strictly speaking, energy is defined with a squaring nonlinearity. However, in a generalized energy function, other alternatives may be used. Numerous nonlinearities have been applied in the literature. Some of the most popular are the magnitude (e.g. [31, 30])

$$|\cdot|, \tag{2.3}$$

the squaring (e.g. [132])

$$(\cdot)^2, \tag{2.4}$$

and the rectified sigmoid (e.g. [50]),

$$|\tanh(\alpha\cdot)|. \tag{2.5}$$

It is worth noting that the magnitude and squaring nonlinearities are parameter free, whereas the rectified sigmoid nonlinearity requires tuning of the saturation parameter, α. This may be an advantage if it is easy to tune this parameter or a disadvantage otherwise. The saturation parameter is dependent on the dynamic range of the input image and on the filter amplification.

The second nonlinearity illustrated in Figure 2.1 is less common. Unser [132] proposed and tested several combinations of the first and second nonlinearity for unsupervised texture segmentation. He concluded that squaring, in combination with a logarithmic normalizing nonlinearity, was the best combination. This combination will consequently be used in most experiments in this dissertation. Unser did not test the rectified sigmoid nonlinearity. However, due to the issue of appropriate saturation parameter determination, this nonlinearity will primarily not be used.

2.1.2 Classification

The output from the local energy function is a set of images, one image per filter. These images are the feature images that will form the basis for the classification. They are used to form feature vectors, where each feature image corresponds to one element in the feature vectors. Classification is the task of assigning class labels to these feature vectors.

Numerous classification approaches are possible and are described in several text books, e.g. [29, 35, 116]. The choice of classifier will not be examined in detail, but classifiers will be used for illustrating the usefulness of the proposed feature extraction schemes. In the experiments, the most common classifier will be the "Type One Learning Vector Quantizing (LVQ)" supervised classifier of Kohonen [58]. However, in some experiments, class maps will be made by thresholding or other nearest neighbor estimations, when the features are designed appropriately.

In order to have a reliable test of the classification, it is necessary to have separate test and training feature vectors. Despite this, it is quite common in texture segmentation to pick the training feature vectors as a subset of the test vectors, see e.g. [32, 51, 104, 123]. This practice has implications on the performance of the classification system and yields results not attainable in a realistic environment. Farrokhnia [32] suggests using 6% of the feature vectors for classifier training in the supervised experiments. To illustrate the effect of this choice, experiments with the texture image of Figure 1.1(a) were conducted. In these experiments, the classifiers were trained using two different approaches:

1. Using 6% of the test feature vectors for training the LVQ.
2. Picking training and test data from non-intersecting regions of the source texture images, without using any edge specific information. The details of the experiments are listed in Tables 6.3, 6.4, and 6.6.

Otherwise, the experiments were identical. A total of 45 feature extraction techniques were applied (see Tables 6.3, 6.4, and 6.6 for details). The average results were 6.3% classification error when using a subset of the test features as training data and 18.6% error when using separate training data. The standard deviations for the error percentages were 4.4% when using a subset and 12.4% when using separate data. In 87% of the cases, the error with separate training data was more than 50% higher. As expected, we see a significant increase in classification error, primarily due to increased edge inaccuracy, when using separate training data. It is also important to note that the number of training vectors was increased when a separate training set was used, typically yielding better classifier training [35].

A number of the illustrative experiments in this dissertation will be with design data from the test images, but in the comparative experiments in Chapter 6, separate test and design data sets will be applied. The experiments with dependent training and test data are given to make the results comparable to other published results.

2.2 Previous works on fixed filter banks

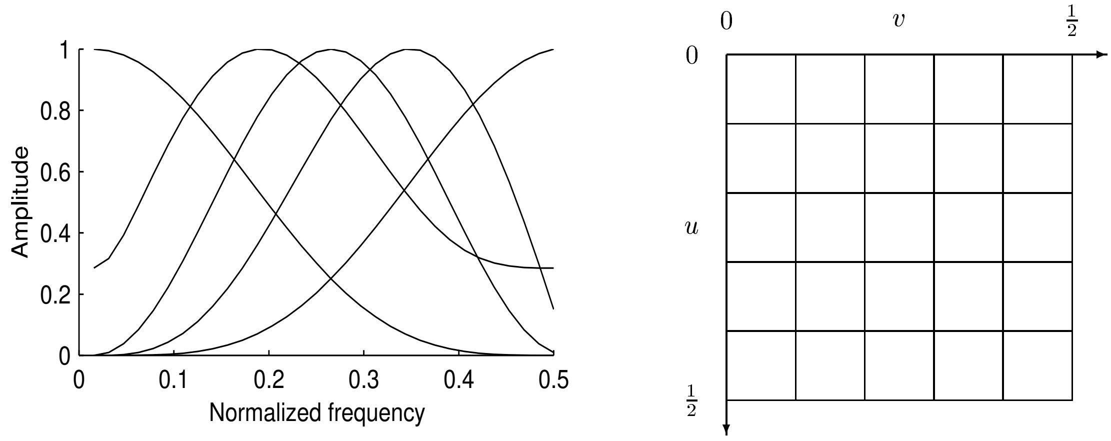
A basic assumption for most filtering approaches is that the energy distribution in the frequency domain identifies a texture. Hence, if the frequency spectrum is decomposed into a sufficient number of sub-bands, the spectral energy signatures of different textures are different. Utilizing this, several filter bank approaches and related schemes have been proposed. The approaches presented here are tested against the new approaches presented in this dissertation in Chapter 6 and, where required, parameter choices for these experiments are described in the following subsections.

2.2.1 Laws filter masks

One of the first approaches to filtering for texture identification was presented in the work by Laws [62]. Laws suggested using a bank of separable filters, five in each dimension, i.e., a total of 25 filters. The filter masks suggested were $h_1 = [1, 4, 6, 4, 1]$, $h_2 = [-1, -2, 0, 4, 1]$, $h_3 = [-1, 0, 2, 0, -1]$, $h_4 = [-1, 2, 0, -2, 1]$, and $h_5 = [1, -4, 6, -4, 1]$. The resulting 25 sub-bands are illustrated in Figure 2.3(b), with the one-dimensional equivalent in Figure 2.3(a).

2.2.2 Ring and wedge filters

Assuming that texture is discriminated by spatial frequency and orientation, Coggins and Jain [21] suggest using seven dyadically spaced ring filters and four wedge-shaped orientation filters for feature extraction. The filters are designed in the two-dimensional



(a) (b)

Figure 2.3: (a) Sub-band split of the one-dimensional equivalent separable Laws [62] filter masks (normalized). (b) The resulting two-dimensional frequency band split. The axis labels are vertical and horizontal normalized spatial frequencies.

spatial frequency domain, giving the amplitude responses of Figure 2.4. The rings and wedges have Gaussian cross sections.

2.2.3 Dyadic Gabor filter bank

Jain and Farrokhnia [50] suggest a bank of Gabor filters, i.e., Gaussian shaped band-pass filters, with dyadic coverage of the radial spatial frequency range and multiple orientations. This choice is justified by the relation to models for the early vision of mammals as well as the filters' joint optimum resolution in time and frequency [36, 26],

The basic even-symmetric Gabor filter oriented at 0° is a band-pass filter with unit pulse response

$$h(k,l) = e^{-\frac{1}{2}\left(\frac{k^2}{\sigma_x^2}+\frac{l^2}{\sigma_y^2}\right)} \cos\left(2\pi f_0 k\right), \tag{2.6}$$

where f_0 is the radial center frequency. Other orientations are obtained by rotating the reference coordinate system, (k,l). This filter has infinite unit pulse response, but in practical experiments it is approximated by a finite filter.

Five radial frequencies are suggested [50] (for images of size 256×256) and four orientations. In the tests presented here, the same number of filters are used, irrespective of the size of the input image. The discrete radial center frequencies

$$\frac{\sqrt{2}}{2^6}, \frac{\sqrt{2}}{2^5}, \frac{\sqrt{2}}{2^4}, \frac{\sqrt{2}}{2^3}, \text{ and, } \frac{\sqrt{2}}{2^2} \tag{2.7}$$

and orientations 0°, 45°, 90°, and 135° are used, see Figure 2.5.

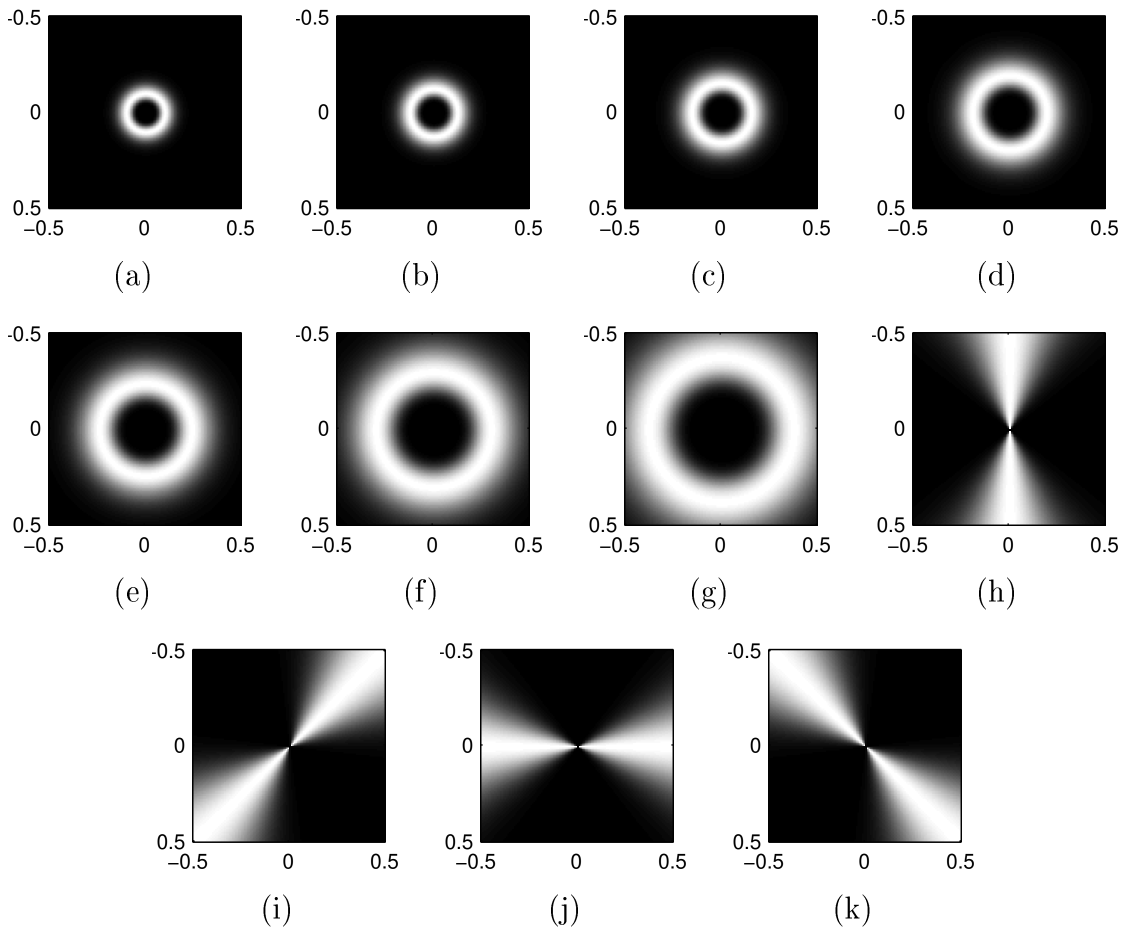


Figure 2.4: Amplitude responses of the ring and wedge filters [21]. The axis labels are normalized spatial frequencies.

2.2.4 Wavelet transforms, packets, and frames

A transform like the discrete wavelet transform corresponds to a critically sampled filter bank with particular filter parameters and sub-band decompositions [2, 124]. Wavelet transform approaches are consequently filter bank approaches. The application of the discrete wavelet transform, and variants thereof, for texture identification has received considerable attention in the literature. The use for texture analysis was pioneered by Mallat [68], who applied a "standard" wavelet transform for feature extraction, i.e., critically decimated with dyadic sub-band structure. The work by Chang and Kuo [16], however, indicate that *texture* features are most prevalent in intermediate frequency bands, thus that the octave band decomposition is not optimal. The trend therefore seems to be a concentration on the wavelet *packet* transform [16, 60, 115], which basically is the wavelet transform with sub-band decompositions not restricted to be dyadic.

The discrete wavelet transform and the discrete wavelet packet transform are critically sampled multi rate filter banks. Particular restrictions apply for the filter coefficients and

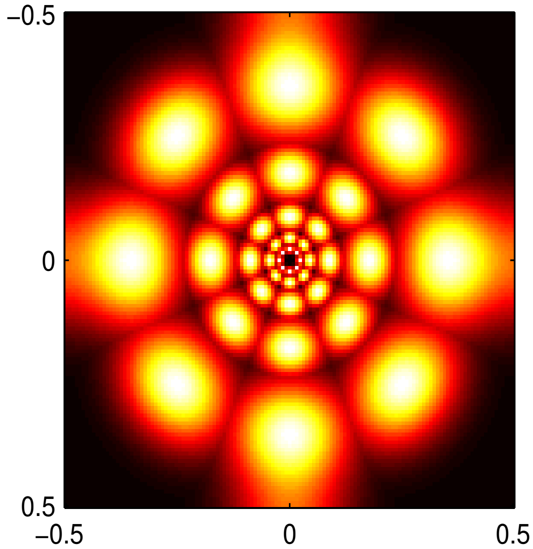


Figure 2.5: The frequency response of the dyadic bank of Gabor filters. The maximum amplitude response over all filters is plotted. Each filter is represented by one center-symmetric pair of lobes in the illustration. The axes are in normalized spatial frequencies.

the sub-band decomposition. However, critically sampled filter banks typically imply inaccurate texture edge localization, see Section 3.2. The use of overcomplete wavelet representations, i.e., wavelet frames [131], is a remedy for alleviating this problem. Improved results with overcomplete representations have been reported [131], see also Section 3.2 and Chapter 6.

Evaluation of absolutely all approaches to texture analysis using wavelet representations is beyond the scope of the experiments. The attention will be restricted to the Daubechies class of wavelets [25]. The sub-band decompositions illustrated in Figure 2.6 will be evaluated.

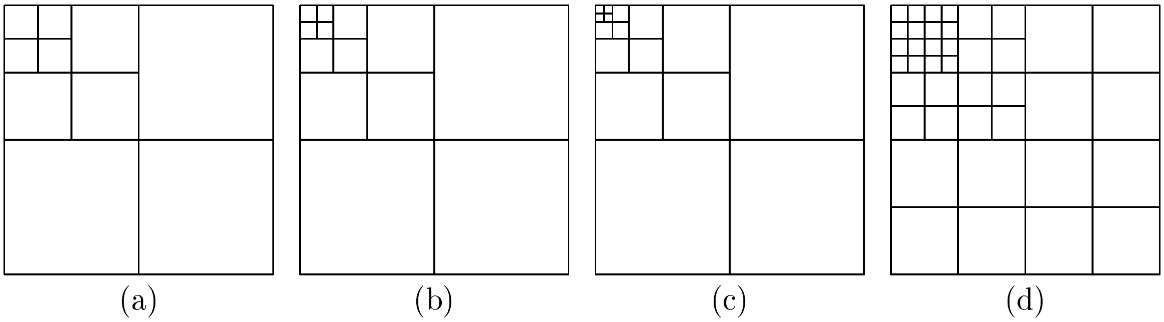


Figure 2.6: Sub-band decompositions evaluated in this dissertation. Decompositions (a-c) are dyadic (octave band), while (d) is not. The axes are the same as in Figure 2.3(b).

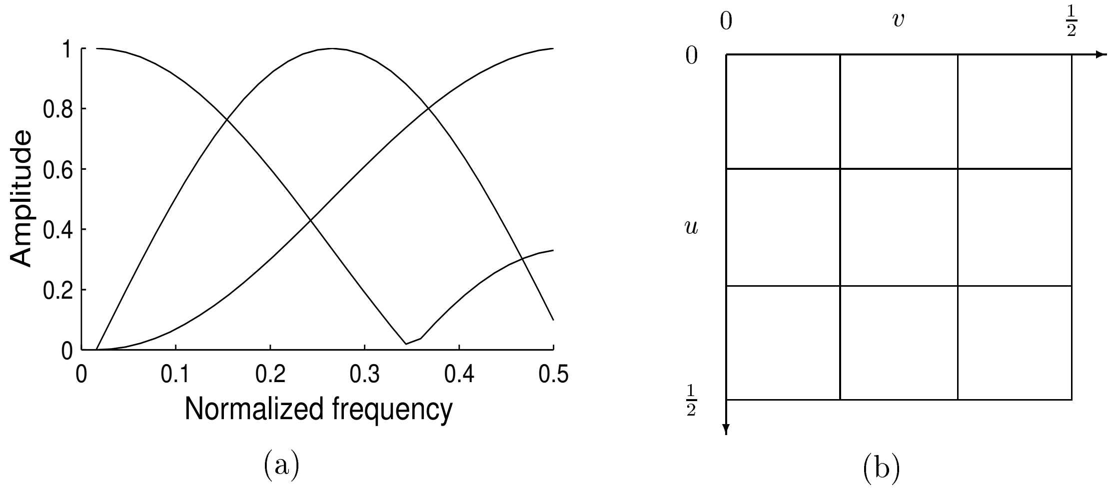


Figure 2.7: (a) One-dimensional equivalent of the amplitude response of the filter bank corresponding to a separable 3×3 Discrete Cosine Transform (normalized). (b) The resulting two-dimensional frequency band split. The axis labels are vertical and horizontal normalized spatial frequencies.

2.2.5 Discrete Cosine Transform (DCT)

The discrete cosine transform is popular in image coding due to good performance and fast implementation [96]. It is, for instance, the backbone in the JPEG compression standard. Ng et al. [79] suggest using a 3×3 DCT for texture feature extraction. They furthermore suggest excluding the low-frequency component of the DCT, thus yielding 8 features.

Image transforms are equivalent to critically sampled filter banks. The above approach is tested in a filter bank implementation, without critical sampling. The filter bank is separable, determined by the one-dimensional filter masks $h_1 = [1, 1, 1]$, $h_2 = [1, 0, -1]$, $h_3 = [1, -2, 1]$. The amplitude responses are illustrated in Figure 2.7.

2.3 Previous work on optimized filters and filter banks

The heuristically[3] designed filter banks have been reported to yield successful results in numerous cases. There are, however, cases where many of them show poor performances, see Chapter 6. Furthermore, most of the heuristically designed filter banks imply large numbers of features. Consequently the computational complexities are large in feature extraction and classification. Hence, it may be desirable to attempt to optimize the filtering operation. Optimizing filters or filter banks may yield low feature vector dimensionality, maximized feature separation, and in some cases more simple classifiers. As in Section 2.2, the approaches presented here are tested against the new approaches of Chapters 3, 4, and 5 in Chapter 6.

As is evident from the presentation, some of the optimization approaches are inher-

[3] By a heuristically designed filter bank is meant a filter bank that is not optimized, i.e., designed more or less irrespective of the textures.

ently restricted to two-texture problems, while others are applicable also to multi-texture problems. Furthermore, some of the approaches yield one filter, while others yield banks of filters. However, common for all approaches is that the filters are optimized with respect to some criterion.

2.3.1 Eigenfilter

Maybe the first approach to optimizing the filter function for texture feature extraction was presented in 1983 by Ade [1]. Ade suggests using eigenfilters derived from the autocorrelation functions of the textures. From each texture, a 9×9 matrix is constructed from the autocorrelation function and the eigenvectors and eigenvalues are computed. Each 9×1 eigenvector corresponds to a 3×3 filter mask. Then, the image is filtered with the 3×3 filters corresponding to the principal eigenvectors. One set of filters is designed for each texture and all filters are applied to the composite image.

In the original approach, the filters corresponding to the eigenvalues summing up to at least 99% of the total eigenvalue sum were selected. This leads to a significant number of filters, typically in the range 5 to 9 per texture. Two of the test images in Chapter 6 have 10 and 16 textures, thus a considerable number of filters are constructed. This implies heavy computational complexities for the feature extraction and classification systems. Therefore, the total number of filters was restricted to maximally $40/N_T$ per texture, where N_T is the number of textures. Hence, the maximum number of filters for any image is 40.

The filters designed by this technique are not band-pass filters. Consequently, using a smoothing operator dependent on the center frequency, as suggested in Section 2.1.1, is not adequate. Hence, a fixed Gaussian smoothing filter with $\sigma_s = 8$ is used. This selection of filter size was based on numerous experiments.

It is worth noting that this approach is optimized with respect to image *representation* – it is closely related to the Karhunen-Loeve transform [96]. However, optimal representation does not imply optimal *discrimination*. This can be illustrated by a simple example. Assume that the two vectors $x_1 = [10\ 0.1]^T$ and $x_2 = [10\ 0.2]^T$ were to be *represented* by a one-dimensional sub-space. Then the Karhunen-Loeve transform yields the transformation vector $[0.9999\ 0.0101]^T$, and the vectors are represented by the one-dimensional representation vectors 10.0005 and 10.0015, respectively. The relative distance between these representations is only 0.01%. On the other hand, if x_1 and x_2 were transformed by the vector $[0\ 1]^T$, the one-dimensional representation vectors would be 0.1 and 0.2. The relative distance between these is 67%. The last representation would be much more robust than the first from a discrimination point of view.

2.3.2 Optimal representation Gabor filter bank

In order to tune the filters to the characteristics of the underlying textures, Bovik et al. [11, 12] suggested using narrow-band Gabor filters. The filters' central frequencies are tuned to the spectral peaks of the textures. That is, for each texture, the central frequencies of the corresponding Gabor filters are selected as the frequencies corresponding to the principal spectral peaks of the texture. Bovik et al. suggest a semi-automatic procedure for determining the spectral peaks. However, this procedure should easily be extendable to be fully automatic [11, 12]. Such a procedure was implemented, selecting two filters per texture. The smoothing filter was tuned with respect to the center frequency of the selected filter.

Like the eigenfilter approach, this approach is only optimal with respect to image *representation*. There is no guarantee that the selected center frequencies will yield good feature *separation*.

2.3.3 Optimal two-class Gabor filter

A Gabor filter design scheme yielding filters optimized with respect to feature *separation* has been suggested by Dunn and Higgins [30]. The Gabor filter center frequency giving features yielding minimum modeled classification error for a pair of textures is designed. The optimal center frequency is determined by evaluating a large range of center frequencies (using the Fourier transform) and selecting the best candidate. The user is required to select the bandwidth, σ, of the filter, see Equation (2.6). Gabor filters with σ-values 2, 4, 8, and 16 were evaluated. These σ-values are suggested by Weldon et al. [138]. Furthermore, as suggested by Weldon et al., smoothing Gaussian filters with spatial widths given by $\sigma_s = 2\sigma$ (see Section 2.1.1) were applied.

Inherently, this filter design approach is based on classification of the feature image by a threshold classifier. Such a simple classifier is less complex to design than a more complex classifier as the LVQ. Furthermore, a large parameter size typically requires more training data vectors. Hence, since a threshold classifier has only one parameter, the needed amount of training data is relatively low.

2.3.4 Optimal multi-class Gabor filter bank

The previous approach to optimal filter design was limited to two-texture problems with one filter. Furthermore, the filter bandwidth had to be determined heuristically. These issues are addressed by Weldon et al. [138, 139]. In their approach, a bank of Gabor filters is designed for problems involving an arbitrary number of textures. The user is only required to select the number of filters to be used. The approach is based on the feature extraction model developed by Dunn and Higgins [30]. However, a modified criterion function, incorporating the effect of the filter size on the edge accuracy, is used. This approach furthermore allows more than two textures and more than one filter.

The filter size is determined by evaluating a number of filter sizes and selecting the best candidate. Different filter sizes are allowed for the different filters in the filter bank. The σ and σ_s values, Equations (2.6) and (2.1), suggested by Weldon et al. [138] were also used in these experiments, that is $\sigma = 2, 4, 8, 16$ and $\sigma_s = 2\sigma$.

2.3.5 Optimal two-texture FIR filters

The Gabor filter is a band-pass filter with a Gaussian shaped pass band. Consequently, the only free parameters of such a filter are the radial center frequency, the orientation, and the bandwidths. By allowing more free parameters, it should be possible to adapt the filter better to the underlying textures. Mahalanobis and Singh [120, 66] suggest a two-texture design approach yielding more general FIR filters with maximum ratio between the extracted mean feature values. That is, filters maximizing the criterion

$$J_{MS} = \frac{\mu_{v_1}}{\mu_{v_2}} \tag{2.8}$$

are designed, where μ_{v_i} is the mean feature value for texture number i.

Inherently, this filter design scheme is based on the assumption of classification by a threshold classifier, just like with the optimal Gabor filters. Like the eigenfilters, filters designed by these approaches are not band-pass filters, thus a Gaussian smoothing filter with $\sigma_s = 8$ is used in the experiments.

The size of the filter mask is also of importance for the results. A symmetric region of support is intuitively preferential for accurate edge localization. In order to have a symmetric region of support, an odd sized mask is required. Experiments indicate that for most textures, mask sizes 5×5 and up are adequate. A few texture pairs have required larger than 5×5 mask sizes. In the experimental chapter, filters with mask sizes 7×7 are optimized.

In Chapter 5, several new related approaches are presented, yielding considerable improvements in feature quality. Furthermore, an extension to problems with multiple textures is proposed.

2.3.6 Random search optimized multi-texture FIR filters

Yet another approach to FIR filter design has been presented by Cohen and You [22]. In their approach, filters are optimized using unguided random search. The search is accomplished by random draw of a filter, feature extraction with that filter, computing an objective function from the extracted features, random draw of a new filter, etc. Eventually, the filter yielding best criterion function is selected. For the design of an $M_h \times N_h$ filter, the parameter space dimensionality is $M_h N_h$. Hence, even moderately sized filters yield very large parameter dimensionality, thus the computation time required to find a good filter will typically be tremendous.[4]

This approach has also been improved with the incorporation of a numerically estimated gradient of the classification error [23, 143], but still the number of computationally complex iterations is high. Furthermore, the search may easily get stuck in a local optimum. None of these approaches will be tested in the comparative experiments in Chapter 6, due to the considerable computational complexity.

2.3.7 Back propagation designed mask

Neural networks have proved to be of practical utility in numerous applications and have also been applied to texture feature extraction. Jain and Karu [51] suggest extracting features and classifying using a feed forward neural network trained by the back-propagation rule. In this approach, the input nodes cover a neighborhood of image pixels. The inputs

[4] This was verified in a discussion with Dr. Cohen in 1995.

are weighted and summed in the nodes in the network, thus parts of the network may be formulated as a filter or filter bank. Furthermore, the application of the nonlinearities in the network resembles the nonlinearity in the local energy function. Hence, this approach is very similar to filtering approaches. The scheme may be modified to utilize the effect of the smoothing operation in the local energy function [51]. However, this generalization is not used in the experiments due to implementational and complexity issues.[5]

A three-layer neural network is used and the neural network parameters (numbers of nodes in the layers, etc.) used are the same at the ones used in [51]. Since this approach has filtering, nonlinearity, smoothing, and classification all in one, it is impossible to use the standard setup (Figure 2.1) for these operations. The only extra element is a 5×5 median filter that is applied as proposed [51] to the resulting class map to reduce speckle-like classification errors.

2.4 Major non-filtering approaches

The focus of this dissertation is on filtering and local energy estimation for texture recognition. An extensive coverage of non-filtering approaches will not be given. However, due to the fundamental importance and wide range of applications of some non-filtering approaches, a brief review is necessary. Furthermore, to provide a reference framework with techniques familiar to readers who are not familiar with filtering, comparisons with two classical non-filtering approaches are provided. Results with statistical co-occurrence and model-based autoregressive features are presented in the comparison of Chapter 6.

2.4.1 Statistical features

In statistical approaches, the textures are described by statistical measures. The probably most commonly applied and referenced method for texture description is the co-occurrence method, introduced by Haralick [42]. In the co-occurrence method, the relative frequencies of gray-level pairs of pixels at certain relative displacements are computed and stored in a matrix, the *co-occurrence matrix* $\mathbf{P}$. For G gray-levels in the image, $\mathbf{P}$ will be of size $G \times G$. If G is large, the number of pixel pairs contributing to each element, p_{ij}, in $\mathbf{P}$ will be low, and the statistical significance poor. On the other hand, if the number of gray-levels is low, much of the texture information will be lost in the image quantization. Ohanian and Dubes [80] reported that $G = 8$ was an appropriate choice for subimages of size 32×32. These parameters have been chosen for the experiments in this dissertation. Since the task is segmentation, overlapping subimages will be used. As suggested by other researchers [80, 123], the combination of the nearest neighbor pairs at orientations $0°$, $45°$, $90°$, and $135°$ will be used in the experiments.

Haralick [42] suggests 14 features describing the two-dimensional probability density function p_{ij}. Four features that are widely used [42, 80, 123] have been selected. These are the Angular Second Moment (ASM), Contrast (Con), Correlation (Cor), and Entropy

[5] In discussions in 1996 and 1997, Kalle Karu claimed that the complexity requirements were too large for practical purposes.

(Ent). They are given by

$$\begin{aligned}
\text{ASM} &= \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} p_{ij}^2 && (2.9)\\
\text{Con} &= \sum_{n=0}^{G-1} N^2 \left\{ \sum_{|i-j|=n} p_{ij} \right\} && (2.10)\\
\text{Cor} &= \frac{1}{\sigma_x\sigma_y} \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} ij p_{ij} - \mu_x\mu_y && (2.11)\\
\text{Ent} &= \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} p_{ij}\log p_{ij}, && (2.12)
\end{aligned}$$

where μ_x, μ_y, σ_x, and σ_y are the means and the standard deviations corresponding to the distributions

$$\begin{aligned}
p_i^{(x)} &= \sum_{j=0}^{G-1} p_{ij} && (2.13)\\
p_j^{(y)} &= \sum_{i=0}^{G-1} p_{ij}. && (2.14)
\end{aligned}$$

2.4.2 Model based features

Another major class of texture features is the model-based features. With model based features, some image model is assumed, its parameters estimated for a subimage, and the model parameters, or attributes derived from them, are used as features.

As an example of this class of features, the multi resolution autoregressive (AR) features introduced by Mao and Jain [71] are used. The autoregressive model for an image $x(m,n)$ can be expressed as

$$x(m,n) = \sum_{(k,l)\in\mathcal{N}} \theta(k,l)x(m-k,n-l) + \sigma_\varepsilon\varepsilon(m,n), \qquad (2.15)$$

where $\mathcal{N}$ is the model neighborhood, $\theta(k,l)$ are the model parameters, and $\sigma_\varepsilon\varepsilon(m,n)$ the model error term. The feature vectors are composed by all the model parameters from the three neighborhoods in Figure 2.8. Furthermore, the error energy, σ_ε, is used as a feature. The parameters are determined in 25×25 pixel overlapping windows. In order to overcome problems with too high feature variances, the features are smoothed. The smoothing filter applied in the experiments here was a Gaussian low-pass filter with $\sigma_s = 3$, see Equation (2.1).

2.5 Summary

In this chapter, an intuitive analysis of texture segmentation with filter, local energy function, and classifier has been given. Different choices for the elements of the local energy

$$
\begin{array}{ccccccc}
\mathcal{N}_3 & & & \mathcal{N}_3 & & & \mathcal{N}_3 \\
& \mathcal{N}_2 & & \mathcal{N}_2 & & \mathcal{N}_2 & \\
& & \mathcal{N}_1 & \mathcal{N}_1 & \mathcal{N}_1 & & \\
\mathcal{N}_3 & \mathcal{N}_2 & \mathcal{N}_1 & & \mathcal{N}_1 & \mathcal{N}_2 & \mathcal{N}_3 \\
& & \mathcal{N}_1 & \mathcal{N}_1 & \mathcal{N}_1 & & \\
& \mathcal{N}_2 & & \mathcal{N}_2 & & \mathcal{N}_2 & \\
\mathcal{N}_3 & & & \mathcal{N}_3 & & & \mathcal{N}_3
\end{array}
$$

Figure 2.8: Neighborhood sets $\mathcal{N}_1$, $\mathcal{N}_2$, and $\mathcal{N}_3$ for the AR features. Each $\mathcal{N}_i$ corresponds to one relative pixel position.

function have been discussed. Furthermore, several fixed filter banks along with filter and filter bank optimization approaches presented in the literature have been reviewed briefly. The review sections are primarily reviews of prior knowledge. However, in order to compare the performances of the filters, the local energy function and classifier should not be very different in the various experiments. Hence, some new considerations are made regarding the local energy function and the classifier. However, inherent filter design requirements, in some cases, are based on specific classifiers or specific local energy functions. In these cases, those specific elements will be used. This implies an unfairness with respect to the comparisons. However, since the choice reflects inherent properties, it also reflects advantages or limitations of the approaches.