Filter and Filter Bank Design for Image Texture Recognition

Trygve Randen

A dissertation submitted in partial fulfillment of the requirements for the degree of

Doktor Ingeniør

Norwegian University of Science and Technology
Stavanger College

Stavanger, Norway
1997

Abstract

The focus of this dissertation is on the design of two-dimensional filters for feature extraction, segmentation, and classification of digital images with textural content. The features are extracted by filtering with a linear filter and estimating the local energy of the filter response. The dissertation gives a review covering broadly most previous approaches to texture feature extraction and continues with proposals of several new techniques.

Texture feature extraction using a quadrature mirror filter (QMF) bank is proposed, utilizing both critically sampled and full rate implementations. In the critically sampled case, tremendous computational savings can be realized. One of the major conclusions of the experiments is that it is possible to use sub-sampled filters with only a modest degradation in segmentation accuracy – realizing considerable computational savings. Furthermore, the commonly used octave band decomposition is evaluated against alternative decompositions, concluding that non-octave decompositions are generally superior. The QMF filter bank features are tested on benchmark images, on document segmentation, and on image content search.

The use of linear least squared prediction error filters is also proposed, yielding an optimal representation of the textures. Compared to non-optimized filter banks, this approach yields a low number of feature images for problems with a moderate number of textures. Still good classification results are obtained in most cases. However, being optimal with respect to texture representation does not guarantee optimality with respect to discrimination. Therefore, approaches for the design of filters with optimal energy separation are proposed. For two-texture problems, an exact closed form solution optimizing the relative distance between the average feature values is derived. Furthermore, an approximate solution for the Fisher criterion is also derived. For both of these solutions, the coefficient vector of the optimal filter is selected from the eigenvectors of the same matrix. A generalized criterion leading to the same eigenproblem is therefore introduced. Motivated by this generalized criterion, a scheme selecting the filter coefficient vector with respect to minimum feature histogram overlap is proposed. The methods are also extended to multiple texture cases and unsupervised problems. A model for the feature extraction process is required for the optimization and is developed and assessed.

The performances of the proposed methods are shown in extensive experiments. These experiments also incorporate methods broadly covering most approaches to texture feature extraction with filters, along with a few non-filtering approaches. The results clearly justify the new approaches.

Finally, new methods for automated seismic interpretation are proposed. In these techniques, the dominating orientation of the seismic is estimated. It is shown that applying filters tuned with respect to the dominating orientation in the seismic signal, powerful features for the detection of discontinuities in the seismic data, i.e., faults, may be extracted. It is furthermore shown how the orientation flow yields tools for analyzing stratum interfaces.

Preface

This dissertation is submitted in partial fulfillment of the requirements for the degree of *doktor ingeniør* at the Norwegian University of Science and Technology (NTNU), Trondheim, Norway. Professors John Håkon Husøy of Stavanger College (HiS), Stavanger, Norway, and Tor Audun Ramstad of NTNU have been the supervisors.

The work, including compulsory courses corresponding to one year full time studies, as well as one year of undergraduate lecturer duties, has taken place in the period of June 1993 to August 1997. The research was carried out at the Electrical and Computer Engineering Department of HiS and with the Research and Engineering Department of Schlumberger Geco-Prakla, Stavanger, Norway. Three months were spent as a visiting scholar at the Pattern Recognition and Image Processing Laboratory at Michigan State University, East Lansing, Michigan, USA. The work was funded by scholarships from NTNU, HiS, and the Norwegian Research Council (grant number 110914/410) and by financial support from Schlumberger Geco-Prakla.

Parts of the work presented in this dissertation have been presented earlier in reports, journals, and conferences [97, 47, 102, 101, 104, 105, 103, 106, 107, 6, 108, 99, 78, 98, 110, 109, 100, 111, 112]. A few elements of the work reported herein have been conducted in cooperation with graduate students for whom I have been supervisor. This is *parts* of the work reported in Sections 5.5 [5] and 5.3.3 [95]. The work presented on filter banks in Chapter 3 was novel when it was conducted in 1993, but later a multitude of similar and related approaches have been presented.

The problem of pattern recognition in seismic signals, Chapter 8, was introduced during the dr.ing. work. This part of the work was carried out in close cooperation with Schlumberger Geco-Prakla. Due to commercial interests, the work on seismic signals has not been strictly texture related. However, the approaches presented in Chapter 8 are related to the other approaches in the use of filters and techniques based on filtering.

Reproducibility is a central concern in the publication of this dissertation. Consequently, most of the images and software used for this dissertation are available on the world wide web at *http://www.hsr.no/sigproc/*.

Stavanger, 1997

Trygve Randen

To my wife Oddveig

Acknowledgements

This work would not have been possible without the help and support from my supervisor professor John Håkon Husøy, who, with his enthusiasm, insight, and new ideas, has been an invaluable support. I would like to thank professor Tor Audun Ramstad of the Norwegian University of Science and Technology for assistance on formal issues in connection with this work. I would also like to thank Lars Sønneland for useful ideas and assistance and for making my close cooperation with the team of Schlumberger Geco-Prakla, Research and Engineering department possible. Without this cooperation, my interesting work on seismic signal analysis would have been impossible. Furthermore, I would like to thank professor Anil K. Jain for making my stay at Michigan State University in the autumn of 1996 possible.

Parts of this work would not have been possible without the efforts of the graduate students Vidar Alvestad and Herindrasana Ramampiaro who I supervised. I would like to thank them for their effort.

I would like to thank my fellow doctoral student Thor Ole Gulsrud for joint jogging trips, support, and cooperation. I am also grateful to Ivar Austvoll, Dag Haugland, Jan Gustav Heber, Bernt Ribbum, Geir Egil Øien, and Sven Ole Aase, the wine wizard, and the other colleagues at Stavanger College for their help and support and for spoiling my time with questions on LaTeX and MATLAB... Special thanks go to the participants of the wine club. We have had several joint experiences in the joys of wine tasting and drinking. Furthermore, I would like to thank Satyavan B. Reymond for reading the sections on seismic interpretation and Jan Ove Hansen, Terje Iversen, Michael Nickel, Claude Signer, Ingve M. Skjeldal, Per Ola Tennbø, Jan Tveiten, Helene H. Veire, Ketil Vågbø, Morten Østebø, Victor Aarre, and the rest of my colleagues at Schlumberger Geco-Prakla for cooperation, help and support.

Last but not least, I would like to thank my wife Oddveig for being supporting and bear with my never-relaxed attitude, my two year old son Markus for reminding me that life has more to it than texture, and my parents Agnes and Oddvar for teaching me to understand the value of hard work and supporting me during my 21 years of education as well as in life in general.

Contents

Chapter 1

Introduction

Webster's Dictionary [137] gives the following definition of texture: "The characteristic structure of the interwoven or intertwined threads, strands, or the like, which make up a textile fabric [...]. The characteristic physical structure given to a material, an object, etc., by the size, shape, arrangement, and proportions of its parts [...]". Most natural surfaces and naturally occurring patterns exhibit texture. A texture recognition system will therefore be a natural part of many computer vision systems. In texture segmentation, the problem is to automatically identify the different regions with homogeneous textures, like in the images in Figure 1.1. For detailed discussions on what is and what is not texture, see [55, 88]. For more definitions of texture, see [20].

This introductory chapter is organized as follows: In Section 1.1, applications of texture analysis are discussed. A discussion of various categories of approaches to texture analysis is given in Section 1.2. The scope and major contributions are discussed in Section 1.3, while an outline of the dissertation is given in Section 1.4.

1.1 Applications of texture analysis

A wide range of examples of the successful applications of texture recognition systems exists. A few examples are discussed briefly in this section. For more information on applications, refer to the overview article by Tuceryan and Jain [128] or to several of the other papers in the reference list of this dissertation.

Medical imaging: Images in medicine arise from non-intrusive techniques as photography, x-ray, ultrasound, tomography, etc. Furthermore, some medical images also arise from intrusive techniques as microscopic photography of biopsies. In several of these image types, textural properties are important for diagnosis. Cancer is often characterized by characteristic textures with different medical imaging techniques. Breast cancer is a leading cause of cancer deaths among women [41]. Early detection is the most effective way to reduce mortality, and mammographic screening is currently the best method for early detection. Texture techniques have been applied successfully to the detection of lesions in mammographic images [41, 40]. Another example of images where cancer is characterized by texture is electron microscope images of thin slices of liver cells. Also for this kind of

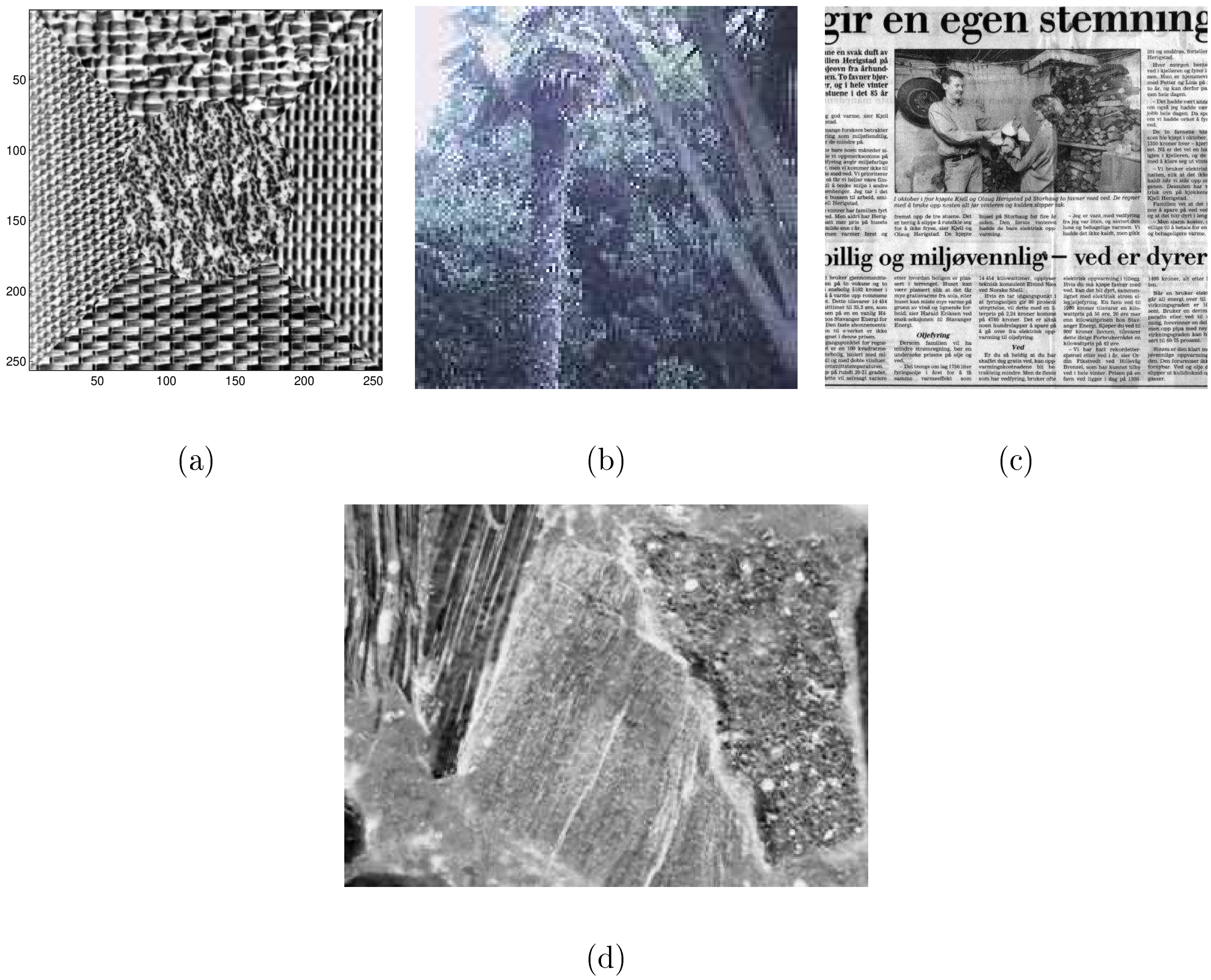


Figure 1.1: Examples of images with textural content. (a) Benchmark composite 256×256 5-texture image. (b) Vacation photo. (c) Scanned newspaper page. (d) Thin slice of a rock core sample from below the surface of the earth.

images, texture techniques have been applied successfully for cancer detection [4].

Remote sensing: Remote sensing is the measurement of properties of an object far away from the object. Remote sensing techniques include satellite photography, satellite multi-spectral measurements, seismic surveys, sonar surveys, etc. Numerous approaches to texture recognition in remote sensing have been presented. Applications include terrain classification [140, 120], cloud classification [61], sea floor mapping [63], and seismic pattern recognition [93].

Industrial inspection: In industrial processes, the detection of defects in manufactured products or in the raw material may be of a crucial importance. Manual visual inspection is often a tedious and laborious task, thus automation is of great interest. In many cases, the quality of a surface is well characterized by its texture. Hence, texture analysis plays an important role in inspection. Numerous approaches to inspection by texture techniques have been proposed. Two examples are quality inspection of painted metallic surfaces [32] and lumber defect detection [113].

Document segmentation: When viewed from a sufficient distance, the text regions of printed documents are by humans perceived as texture. For a particular language, font, etc., the text appears as a homogeneous texture, see for example Figure 1.1(c). By the use of texture analysis, it may be possible to segment regions of text from regions of graphics in scanned documents. Such segmentation is important for successful optical character recognition and may be utilized in for example digitizing of paper documents or periodicals. Examples are given in Section 7.1 and in references [49, 84].

Content based image search: Digital image data has been around on computers and on the Internet for a long time. However, locating an image with a specific content is not straight-forward. Without search tools, this requires manual labeling or browsing of the images. Hence, an evolving research area in computer vision is content based image search. Some content types, e.g., hair, leaves, sky, and grass, have texture as an important characteristic. Therefore, applications including texture recognition have been presented [34, 86, 88], see also Section 7.2.

Shape from texture: Texture analysis has also found its application in reconstructing three-dimensional shape from two-dimensional images. By analyzing how the texture pattern changes, the shape of an object may in principle be reconstructed from the texture [125].

Arbitrary-dimensional textures: The typical texture context is two-dimensional image processing, but it is also possible to extend the concept of texture to an arbitrary dimensionality. For instance, speech recognition systems using short-time spectral analysis can in this respect be regarded as texture analysis tools.

In order to measure the quality of a texture feature extraction scheme, numerous approaches have been devoted to the application on benchmark images. An example benchmark image is shown in Figure 1.1(a). Several experiments on benchmark images are presented in this dissertation, especially in the comparative study of Chapter 6. Most benchmark textures used in this dissertation, including Figure 1.1(a), are scanned from Brodatz [14].[1]

[1]The images from Brodatz [14] were scanned using a Hewlett Packard ScanJet II scanner with a 640×640 pixels resolution and eight bits per pixel. All test images are available on the Internet from http://www.hsr.no/sigproc/. It is noted that the scanning resolution will have significant impact on the performance of most texture features. Conclusions drawn from experiments are therefore highly dependent on scanning resolution.

1.2 Approaches to texture analysis

Approaches to texture feature extraction and recognition span a wide range of methods. Several books and articles give overviews of the available methodology [43, 48, 114, 128]. Four major categories may be identified [128];

Statistical: Co-occurrence and autocorrelation features.

Geometrical: Voroni tessellation and structural features.

Model based: Random field and fractal parameters.

Signal Processing: Spatial and spatial frequency filters and filter banks, wavelet transforms, wavelet packets, and wavelet frames.

The geometrical features are of limited practical utility [114] due to tight assumptions on the images. They are probably the least commonly applied features in texture recognition. The other three categories are included in experiments in this work.

1.3 Scope of this dissertation

The focus of this dissertation will be on the *signal processing approaches*. Multiple new filter bank and optimal filter and filter bank approaches will be proposed. Examples on the applications of the approaches will be presented and comparisons with previously published works will be given. For reference, experiments with statistical and model based features are also given. Briefly summarized, the major contributions of this dissertation are:

1. Introduction of the class of quadrature mirror filters (QMF) to texture feature extraction, along with the derivations of multi-rate local energy functions. Considerable improvements of the computational complexity are realized by the multi-rate (critically sub-sampled) filter banks.

2. It is shown how the multi-rate filter bank may be utilized for document segmentation and image content search.

3. Proposal of optimal prediction error filters for texture feature extraction. Class maps are formed by selecting the class corresponding to the minimum local prediction error.

4. Development of a powerful filter feature extraction model. The model includes expressions for estimating mean and variance of features extracted by a filter and a local energy function.

5. Proposals of several new filter optimization approaches for two-texture problems.

6. Proposal of a new multi-texture optimal filter bank design scheme.

7. Proposals of solutions to the problems of fault and stratigraphic pattern recognition in seismic signals.

A more detailed summary of the contributions is given in Section 9.1.

1.4 Outline of this dissertation

This dissertation can be divided into three parts; the introductory part, the methodology part, and the applications part. The methodology chapters constitute the core part of this dissertation. An overview of the chapters is given in this section.

1.4.1 Introduction

Chapter 1: Introduction to the problems and a brief overview of applications and approaches.

Chapter 2: A description of the principles of texture feature extraction with filters is given. Furthermore, a survey covering broadly most previous works on signal processing approaches to texture recognition is given.

1.4.2 Methodology

Chapter 3: An extension of the dyadic Gabor filter bank of Jain and Farrokhnia [50] to arbitrary sub-band decompositions is proposed. Furthermore, the class of QMF filter banks is introduced for texture filtering. In order to utilize sub-sampling of the QMF bank, a novel local energy function for sub-sampled filter banks is proposed.

Chapter 4: Two different filter optimization approaches are suggested in Chapters 4 and 5. In Chapter 4, an approach based on linear prediction error filtering is proposed. This approach is optimal with respect to representation of the textures, but not necessarily optimal with respect to discrimination.

Chapters 5: Several new FIR filter design techniques yielding filters optimal with respect to discrimination are proposed in Chapter 5. These are optimization approaches for two-texture problems with respect to several criteria, extensions to multiple textures, and an approach to unsupervised optimal filtering. Furthermore, necessary feature extraction models are developed.

Appendix A: The most involved details of the mathematical derivations and proofs of Chapter 5 are put in this appendix. Furthermore, a method to find the upper bound for one of the criteria, the Fisher criterion, is developed.

Chapter 6: Numerous approaches to texture feature extraction with filters are proposed in the literature and some novel techniques are proposed in this dissertation. However, no extensive comparisons covering most of the approaches have been presented previously. In Chapter 6, an extensive comparative analysis is compiled. These experiments cover broadly most published filters and filter banks for texture feature extraction. The study also includes the new approaches proposed herein. Special emphasis is put on making the experiments reliable and comparable, to be able to judge the quality of the different approaches. This chapter concludes the methodology part of the dissertation.

1.4.3 Applications

Chapter 7: The applications of the QMF filter banks of Chapter 3 to the problems of document segmentation and content based image search are addressed in Chapter 7.

Chapter 8: Another application, seismic pattern recognition, is addressed in Chapter 8. Solutions to two problems in seismic signal analysis, automated fault and automated stratigraphic boundary identification, are addressed. Not being strictly texture analysis, this chapter is somewhat on the side of the other chapters. However, since the approaches are based on feature extracting filters and other elements of the texture recognition systems, these approaches are closely related to texture approaches.

Chapter 9: The dissertation is summarized in Chapter 9, listing the major contributions and conclusions of this work and directions for future research.

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.

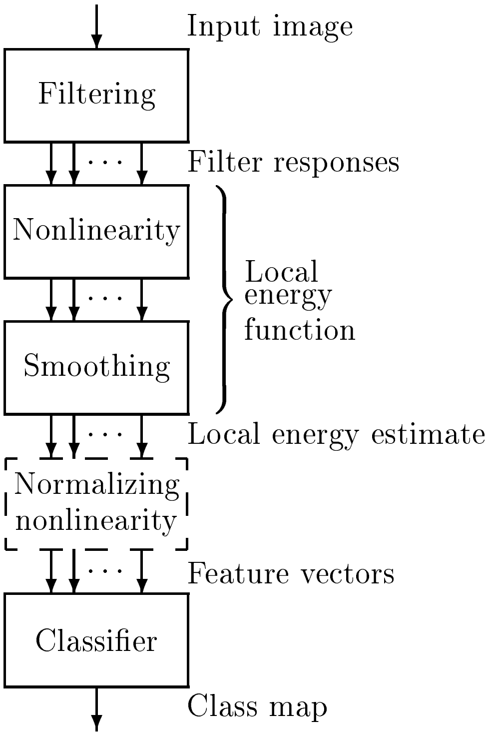


Figure 2.1: Experimental setup for most experiments in this dissertation. The dashed box is optional, see the text.

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

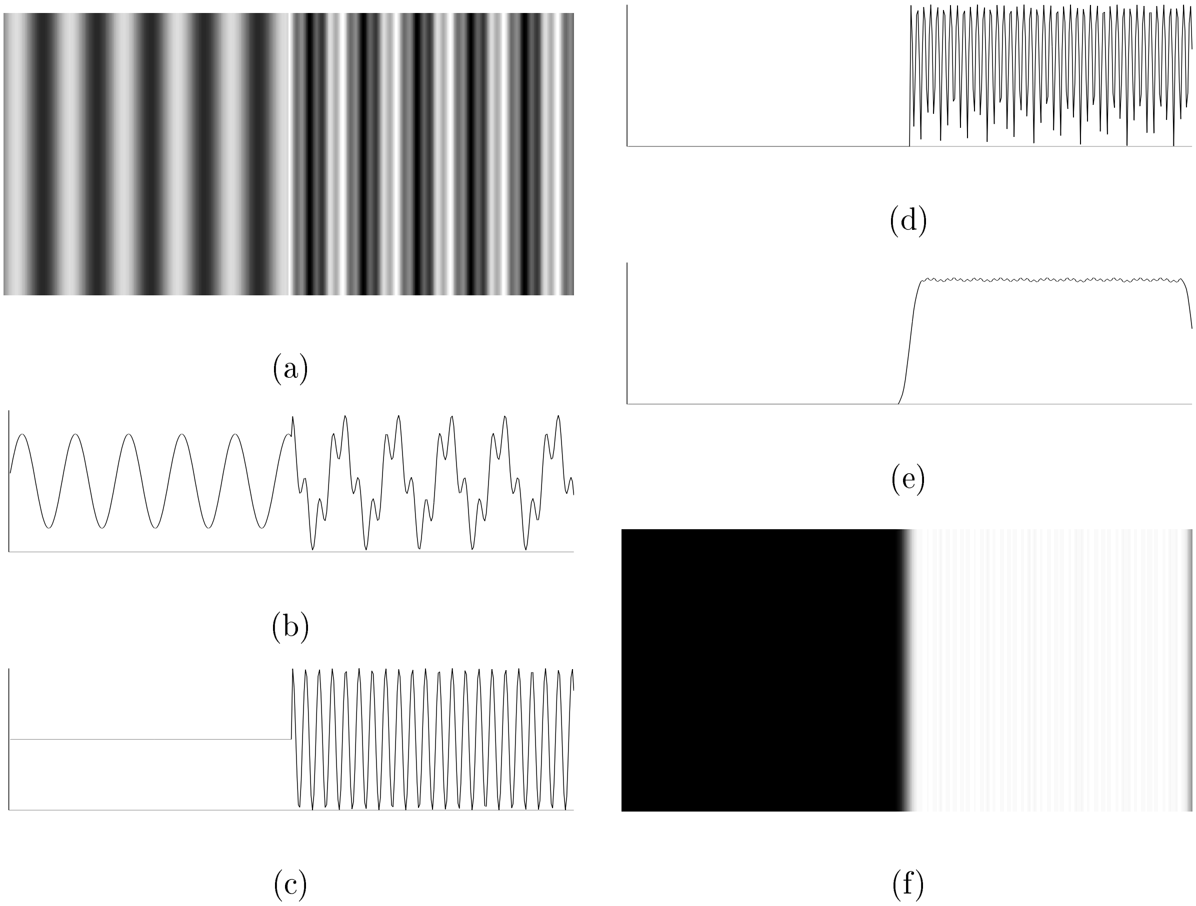


Figure 2.2: Illustration of a typical texture feature extraction process with filters. (a) Two synthetic textures, (b) a horizontal scan line through the image, (c) filtered, (d) nonlinearly transformed, (e) smoothed, and (f) the resulting 2D feature image.

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

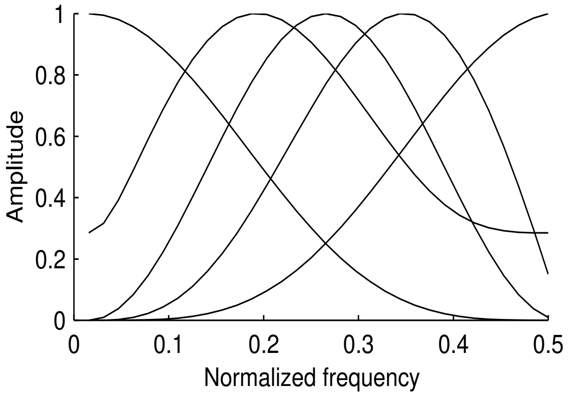
A basic assumption for most filtering approaches is that the energy distribution in the frequency domain identifies a texture. Hence, if the frequency spectrum is decomposed into a sufficient number of sub-bands, the spectral energy signatures of different textures are different. Utilizing this, several 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

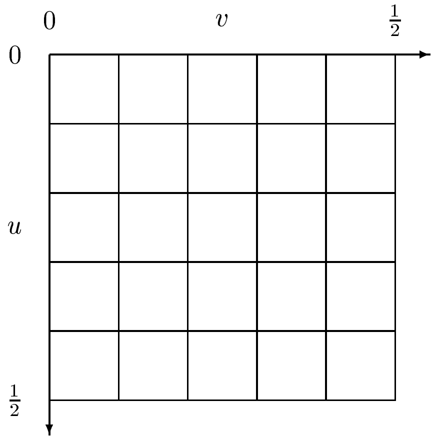
One of the first approaches to filtering for texture identification was presented in the work by Laws [62]. Laws suggested using a bank of separable filters, five in each dimension, i.e., a total of 25 filters. The filter masks suggested were $h_1 = [1, 4, 6, 4, 1]$, $h_2 = [-1, -2, 0, 4, 1]$, $h_3 = [-1, 0, 2, 0, -1]$, $h_4 = [-1, 2, 0, -2, 1]$, 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.

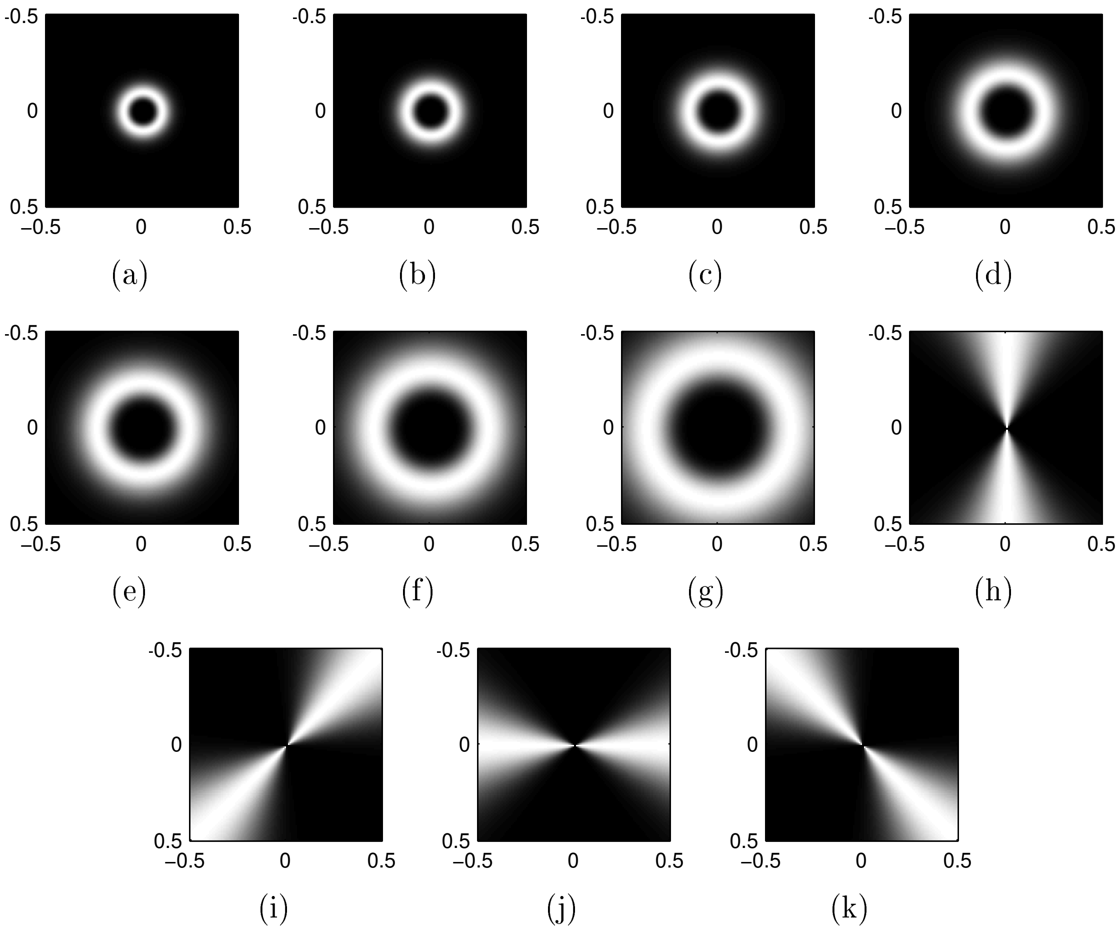


Figure 2.4: Amplitude responses of the ring and wedge filters [21]. The axis labels are normalized spatial frequencies.

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

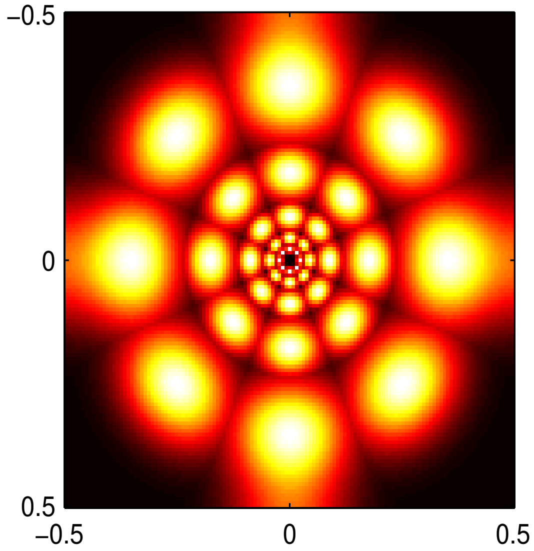


Figure 2.5: The frequency response of the dyadic bank of Gabor filters. The maximum amplitude response over all filters is plotted. Each filter is represented by one center-symmetric pair of lobes in the illustration. The axes are in normalized spatial frequencies.

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.

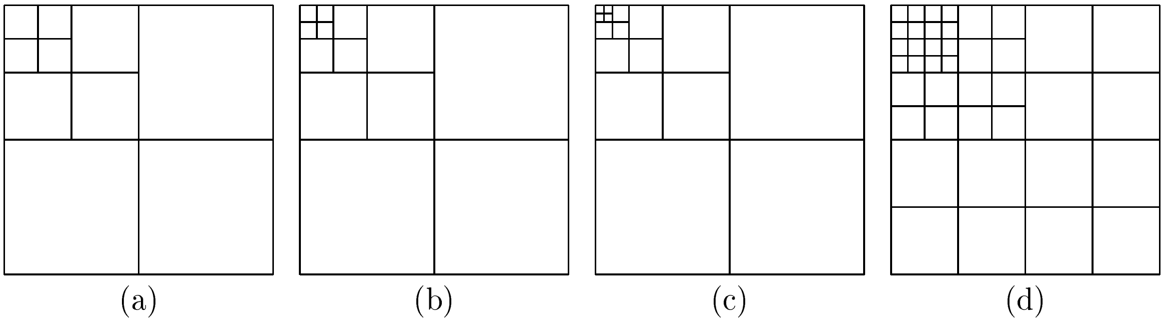


Figure 2.6: Sub-band decompositions evaluated in this dissertation. Decompositions (a-c) are dyadic (octave band), while (d) is not. The axes are the same as in Figure 2.3(b).

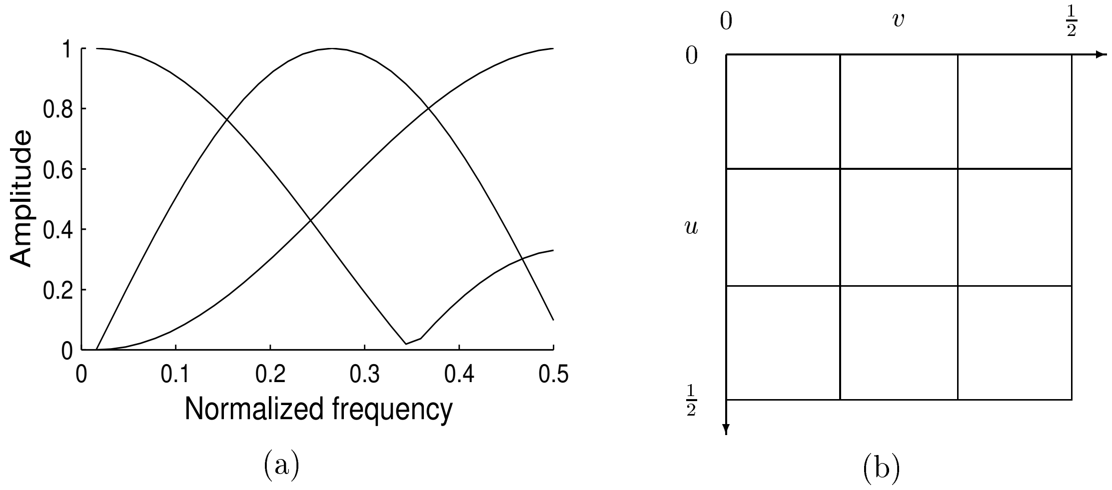


Figure 2.7: (a) One-dimensional equivalent of the amplitude response of the filter bank corresponding to a separable 3×3 Discrete Cosine Transform (normalized). (b) The resulting two-dimensional frequency band split. The axis labels are vertical and horizontal normalized spatial frequencies.

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

$$\text{ASM} = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} p_{ij}^2 \tag{2.9}$$

$$\text{Con} = \sum_{n=0}^{G-1} N^2 \left\{ \sum_{|i-j|=n} p_{ij} \right\} \tag{2.10}$$

$$\text{Cor} = \frac{1}{\sigma_x \sigma_y} \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} ij p_{ij} - \mu_x \mu_y \tag{2.11}$$

$$\text{Ent} = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} p_{ij} \log p_{ij}, \tag{2.12}$$

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

$$p_i^{(x)} = \sum_{j=0}^{G-1} p_{ij} \tag{2.13}$$

$$p_j^{(y)} = \sum_{i=0}^{G-1} p_{ij}. \tag{2.14}$$

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

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

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.

Chapter 3

Fixed filter banks

As we saw in Chapter 2, several approaches to multi channel filtering for texture segmentation are proposed. There are different, and to some extent conflicting, objectives for the choice of a filter bank. The principal objective is good feature separation, i.e., good classification or segmentation accuracy. However, issues as feature extraction complexity, feature dimensionality, memory requirements, and how easy the feature extractor may be tuned may also be important.

Two new multi channel filter approaches are presented in this chapter, addressing some of these objectives. In Section 3.1, alternatives to the Gabor filter bank sub-band decomposition of Jain and Farrokhnia [50] are suggested. The idea for suggesting new decompositions, is that they may yield improved segmentation accuracies. In Section 3.2, filters applied successfully to image compression [46] are introduced for texture feature extraction. Furthermore, the use of critically sampled filter banks is proposed. These approaches are advantageous with respect to their computational properties. It is therefore interesting to see if they can compete with the more computationally complex filter banks.

3.1 Non-dyadic Gabor filter bank decomposition

It has been common to decompose the frequency band by the octave band (dyadic) decomposition [50, 87, 126, 129]. This choice is motivated by the similarities between the octave band decomposition and the early vision of mammals. Octave band decompositions imply finer frequency resolutions in lower than in higher frequency regions. The work by Chang and Kuo [16], however, indicate that texture features are most prevalent in intermediate frequency bands. Hence, the octave band decomposition is not ideal for texture recognition. In this section, an extension of the scheme of [50] to more arbitrary decompositions is developed.

Some possible two-dimensional sub-band decompositions are given in Figure 2.6. The filter of Equation (2.6) is used and f_0 is set to the radial value of the center frequency of the pass-band. The filter orientation is set to the corresponding angle. Previously, filter orientations in the range $0°$ to $180°$ have been used [50]. The decomposition proposed

(a) (b)

Figure 3.1: Segmentation results with the texture mosaic of Figure 1.1(a) and different Gabor filter approaches, (a) with the decomposition of Jain and Farrokhnia [50] and (b) with the sub-band decomposition of Figure 2.6(d).

here will only make it possible to specify orientations from 0° to 90°.[1] To maintain some selectiveness of angles from 90° to 180°, filters whose frequency responses are the sums of the frequency responses in the range 0° to 90° and the frequency response of the orthogonal filter are used.

In the experiments, 6% of the feature vectors (see the discussion in Section 2.1.2) were used for training an LVQ classifier. The experimental results are illustrated in Figure 3.1. The classification errors were 5.4% with the decomposition of Figure 2.6(d) and 2.4% with the dyadic decomposition, indicating that the non-octave band decomposition is a good alternative. Remember that in this experiment, the dyadic decomposition was more flexible with respect to orientation. In more thorough experiments in Chapter 6, it is shown that the alternative decomposition outperforms the dyadic decompositions in several cases. When the same orientation limitations were imposed on both dyadic and non-dyadic decompositions, the non-dyadic decomposition yielded better or practically equal results in all test cases. The problem, however, remains the same for virtually all decompositions: to achieve a good segmentation quality we need a large number of sub-bands. This leads to a high computational complexity, as well as large memory requirements.

3.2 Quadrature mirror filter banks

In most approaches to multi channel texture segmentation, filter banks in which the sub-bands each *have the same number of image samples as the unfiltered image* are applied. Hence, after filtering, the amount of data is increased by a factor corresponding to the number of frequency channels in the filter bank. Sub-sampling of the frequency bands so that the total number of sub-band signal samples equals the number of signal samples in the input image can be performed without losing any information, as long as the filters are designed appropriately [133]. To achieve correct classification we should in principle not need a redundant representation. For critically sampled filter banks with

[1] The reason for this restriction is parallelism to the separable approach of Section 3.2.

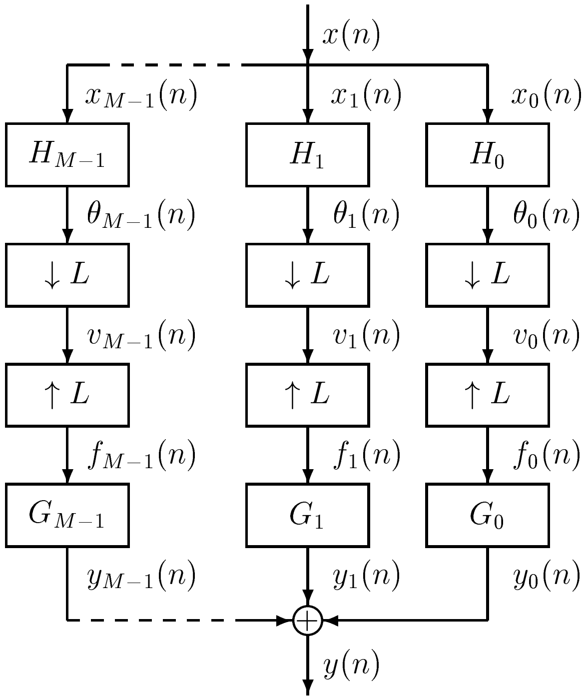


Figure 3.2: General analysis-synthesis filter bank.

perfect reconstruction, as used in image coding [96], the problem with redundancy is solved. It is examined what kind of segmentation can be obtained by operating on sub-sampled frequency channels when using critically sampled filter banks allowing perfect reconstruction of the input signal. By using this approach, tremendous computational savings over full rate approaches can be realized.

In Subsection 3.2.1 the basic theory for multi rate signal processing is reviewed and in Subsection 3.2.2 a local energy function for multi rate texture feature extraction is proposed. Computational characteristics are discussed in Subsection 3.2.3, and experimental results are given in Subsection 3.2.4.

3.2.1 Multi rate signal decomposition

Figure 3.2 shows a general *analysis - synthesis* filter bank. The symbol $\downarrow L$ represents *sub-sampling* by L, while the symbol $\uparrow L$ represents *up-sampling* by L. Sub-sampling by L is the process of discarding all but every L'th sample, while up-sampling by L is the process of inserting $L-1$ zeros between each sample.

The filter bank in Figure 3.2 is *critically sampled* if $L = M$, that is, if the sub-sampling factor is equal to the number of filters in the filter bank. A critically sampled filter bank has *perfect reconstruction* if the output of the synthesis filter bank is equal to the input to the analysis filter bank (with a possible delay). There exists an infinite class of filter banks with perfect reconstruction properties.

It is possible to implement the filtering operations of the filter banks at low sampling

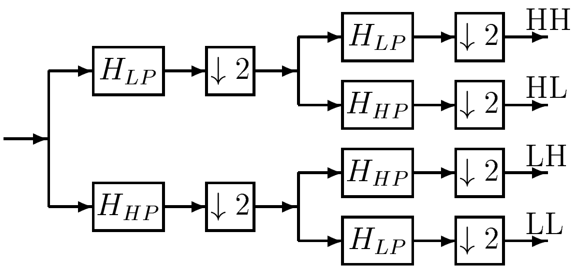


Figure 3.3: Tree-structured filter bank. The nodes correspond to filters and sub-samplers.

rate and thereby achieve big savings in computational complexity [2]. The theory of multi rate filter banks is covered in detail in [2, 133]. The further discussion is restricted to *separable* and *tree-structured* perfect and approximately perfect reconstruction filter banks. A filter bank is separable if it is possible to apply the filters in each dimension separately. A filter bank is tree-structured if it is built up as indicated in Figure 3.3 where each node in the tree is a filter bank.

Two types of filters will be used, *finite impulse response* (FIR) and *infinite impulse response* (IIR) filters. The IIR filters used are based on digital *all-pass* filters, given by

$$H_{LP}(z) = A_0(z^2) + z^{-1}A_1(z^2) \tag{3.1}$$
$$H_{HP}(z) = A_0(z^2) - z^{-1}A_1(z^2), \tag{3.2}$$

where

$$A_i(z) = \frac{a_i + z^{-1}}{1 + a_i z^{-1}}, \tag{3.3}$$

are the first order all-pass filters. First-order filters were empirically found to be sufficient. Two IIR filter bases will be used in this dissertation, these are "F_2_1_smpl" and "F_2_1_09" [46], with $a_0 = 0$ and $a_1 = \frac{1}{2}$, and $a_0 = 0$ and $a_1 = 0.6632$, respectively. See Johnston [53] for the coefficients of the FIR filters.

3.2.2 Multi rate local energy function

In Section 2.1.1, an experimental setup for multi channel filtering for texture segmentation with band-pass filters was presented. To be able to use sub-sampled filter banks successfully, a modification to the smoothing filter in the local energy function is needed. All other parts, except from the filter bank, remain the same.

Assuming no aliasing, when we decimate the signal by a factor N, a sinusoid with period M and frequency $f = 1/M$ will get the period M/N and the frequency $N/M = Nf$.[2] Hence, the smoothing filter designed for a given full rate signal will no further be

[2]Strictly speaking, restrictions for periodicity of discrete signals exist [18]. Particularly, the signals period must be a rational factor of the sampling interval. However, in this context, this issue is of purely theoretical interest.

appropriate for the same signal after sub-sampling. We remember that σ in Equation (2.2) was inversely proportional to the radial frequency f_0. Therefore the equation

$$\sigma = \frac{\sqrt{2}}{2} \frac{1}{N f_0} \tag{3.4}$$

may be used, where N is the sub-sampling factor and f_0 is the absolute value of the center frequency of the filter bank channel.

When a high-pass band of a filter bank is sub-sampled, the relation between signal frequency before and after sub-sampling is not as given above, due to aliasing [133]. However, the corresponding relationships are easily developed.

3.2.3 Computational characteristics

The filter bank basis defined by Equations (3.1), (3.2), and (3.3) has several properties that make it computationally very attractive, especially when subsampling is applied. First of all, the all-pass filters of Equation (3.3) have very few parameters. Furthermore, if subsampling is applied, only the first application of the low-pass and high-pass filters are at the full rate image and subsequent levels have sub-sampled data as input, see Figure 3.3. For example, for the decomposition of Figure 2.6(a) it can easily be shown that for an $M \times N$ image there will be only

$$4MN + 4\frac{M}{2}\frac{N}{2} + 4\frac{M}{4}\frac{N}{4} = 5.25MN \tag{3.5}$$

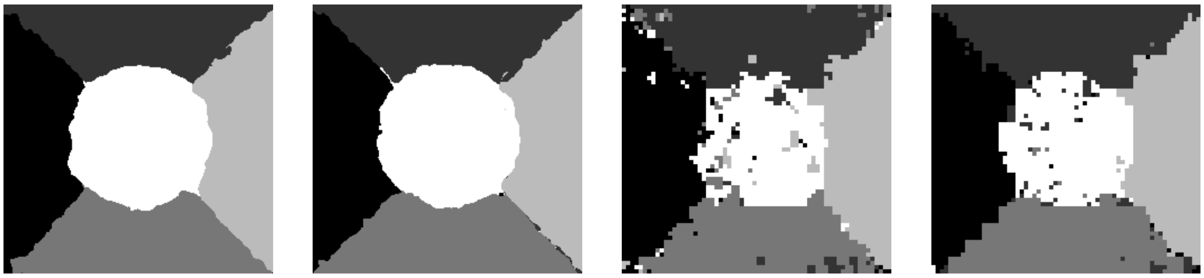
multiplications in the entire filter bank. Furthermore, for the filter basis "F_2_1_smpl", all of these multiplications are with numbers that are powers of two. Hence, these multiplications may be implemented by bitwise shifts.

The filter bases applied in this work give filter banks with good image compression capabilities [46]. If an image being examined is compressed with such a filter bank [96], only parts of the decompression have to be performed to extract the features.

Note that due to the subsampling, the complexity of this approach is significantly less than that of full rate filter approaches. With the decimated filter bank, the Gaussian smoothing filtering in the local energy function is performed on the sub-sampled dataset. It is thereby not very computationally complex. The computational characteristics of the FIR QMF filter banks are not as good as those of the IIR banks. However, with sub-sampling the complexity is still far less than that of full rate filter banks. Last, but not least, using sub-sampling implies classification with significantly reduced feature data sets.

3.2.4 Results

In order to study the effect of sub-sampling with respect to segmentation accuracy, tests with both full rate and sub-sampled filter banks have been performed. Some results are reported here and more results are given in Chapter 6. Segmentation using the perfect reconstruction IIR filter banks from Husøy [46] and the nearly perfect reconstruction FIR



(a) (b) (c) (d)

Figure 3.4: Segmentation results from experiments with the QMF filters. (a) Results from the full rate “F_2_1_09”, (b) full rate “f16b”, (c) critically sampled “F_2_1_09”, and (d) critically sampled “f16b” filters. The test image was the texture mosaic of Figure 1.1(a) and the sub-band decomposition was the one in Figure 2.6(d).

filter banks from Johnston [53] were tested. The squaring with logarithmic second nonlinearity, see Section 2.1.1, is used. Experiments on a large number of combinations of filters, images, and sub-band decompositions have been performed [97]. This study concluded that the FIR filter “f16b” [53] and the IIR filter “F_2_1_09” were quite representative of the different classes of filters.

Examples of the segmentation of the image in Figure 1.1(a) with these filters at full rate are shown in Figures 3.4(a) and 3.4(b). We see that perfect reconstruction filters are suited to texture segmentation when no sub-sampling of the channel filters is applied (full rate). The classification errors were 2.9% in both cases. The LVQ classifier [58] was applied, using 6% of the feature vectors for training.

However, by using full rate filters we do not achieve the full potential savings in computational complexity. We remember that due to perfect or nearly perfect reconstruction properties of these filters, practically all image information is retained in critically sampled sub-bands. Is it possible to use critically sampled filter banks and achieve good segmentation? Figures 3.4(c) and 3.4(d) show the results from two experiments. The classification errors were 10.8% with the “F_2_1_09” filter and 6.2% with the “f16b” filter. We see from the results that the segmentation accuracies in the sub-sampled cases are only slightly degraded relative to the full rate results. Remember that the computational savings are tremendous. For the current test image, we have a feature set of 2 621 440 feature samples in full rate and only 65 536 samples with sub sampling. This has large implications on the computational complexity and the memory requirements.

We see that the segmentation based on critical sampling has a stair-case effect near the edges. Further experiments show, as expected, that the stair-case effect is reduced if we use a lower sub-sampling rate, but at a higher computational cost. With a sampling rate of four times the critical sampling and the filter “F_2_1_09”, the classification error was 3.1% with drastically reduced stair-case effect, see Figure 3.5.

Segmentation from sub-sampled filter bank outputs show promising results when the filter banks have the perfect reconstruction property. However, when we are segmenting images, we do not necessarily need the perfect reconstruction property. What if we sub-sample filter banks without the perfect reconstruction property? Experiments [104] show

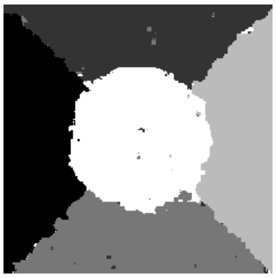


Figure 3.5: Segmentation result from four times critical sampling of the QMF filter “F_2_1_09” with with the texture mosaic of Figure 1.1(a) and the sub-band decomposition of Figure 2.6(d).

that significant degradation may occur if subsampling Gabor filters. Gabor filters do not have the perfect reconstruction property when sub-sampled. This, however, does of course not prove that non-perfect reconstruction filters yielding good segmentation do not exist.

3.3 Summary

In this chapter a new scheme for sub-band decomposition with the Gabor filter has been proposed and tested. Furthermore, the class of QMF filter banks has been introduced for multi channel filtering for texture segmentation. Several QMF and other filter banks have the prefect reconstruction property when sub-sampled, i.e., it is possible to reconstruct the input to the filter bank from the sub-sampled outputs. Using sub-sampling allows considerable savings in computational complexity. A computationally inexpensive scheme for local energy computation and classification using sub-sampling has been proposed in this chapter. The experiments have illustrated that other sub-band decompositions than the commonly used octave band decomposition may be viable. Furthermore, it is shown that it is possible to use critically sampled perfect reconstruction filter banks at the cost of only a modest degradation in segmentation quality.

Chapter 4

Optimal representation filters

As we have seen in Section 2.2 and Chapter 3, many approaches to texture feature extraction with filters apply banks of filters that are more or less determined irrespective of the textures to be discriminated. As we will see in Chapter 6, in some cases even filter banks with a relatively large number of sub-bands are incapable of yielding adequate segmentation of some of the test images. Hence, there is no guarantee that the bank of filters will work properly for a specific set of textures. Another unpleasant consequence of filter bank approaches is that they are to varying degree computationally demanding. For these reasons, the design of *optimal* filters for texture discrimination is gaining increased interest, see Section 2.3. Optimization offers the potential of reduced feature dimensionality and hence reduced computational complexity and/or better feature separation.

In this chapter, an optimization approach based on least mean squared linear prediction error filtering is presented. The linear prediction error filters are optimized with respect to the prediction error. This filter bank design approach is applicable to problems with arbitrary numbers of textures, and the number of filters is equal to the number of textures. Prediction error filtering is reviewed in Section 4.1, least squares auto regressive (AR) parameter estimation is presented in Section 4.2, and some experimental results are given in Section 4.4.

4.1 Prediction error filtering

Linear prediction is an important topic in digital signal processing having many practical applications [94]. A linear predictor in image processing is an equation on the form

$$\hat{x}(m,n) = -\sum_{(k,l)\in\mathcal{N}} \theta(k,l)x(m-k,n-l), \tag{4.1}$$

predicting a pixel value on the basis of neighboring pixels in the neighborhood $\mathcal{N}$ using the coefficients $\theta(k,l)$. The neighborhood $\mathcal{N}$ is not including the pixel itself, $(k,l) = (0,0)$. Assuming wide sense stationarity, a least squared error (LSE) predictor is a predictor minimizing

$$\varepsilon = \sum_{m,n} \left(\hat{x}(m,n) - x(m,n)\right)^2. \tag{4.2}$$

Consequently, the least squares error predictor for a texture will be the predictor yielding maximum similarity between the texture and the predictor output, in the squared error sense. Hence, the energy of the image

$$\begin{aligned} e(m,n) &= x(m,n) - \hat{x}(m,n) \\ &= x(m,n) + \sum_{(k,l)\in\mathcal{N}} \theta(k,l)x(m-k,n-l) \end{aligned} \tag{4.3}$$

is minimized. The image $e(m,n)$ is denoted the *prediction error image*. It is noted that Equation (4.3) expresses essentially a linear filtering operation, thus the term *prediction error filtering*.

Generally the LSE predictors for different textures are different. Hence, for any texture, the predictor yielding minimum mean error energy will be the predictor designed with respect to that texture. This may be used for classification, since the predictor yielding minimum local prediction error energy most likely corresponds to the underlying texture.

4.2 Least squares AR parameter estimation

If we arrange the $x(m-k,n-l)$ and $\theta(k,l)$ terms for all (k,l) in $\mathcal{N}$ in the vectors $\mathbf{x}(m,n)$ and $\boldsymbol{\theta}$, respectively, then Equation (4.3) may be written in vector form as

$$e(m,n) = x(m,n) - \mathbf{x}(m,n)^T\boldsymbol{\theta}. \tag{4.4}$$

If, for example, the region of support, $\mathcal{N}$, is a 3×3 rectangle, we have

$$\boldsymbol{\theta} = \begin{bmatrix} \theta(-1,-1) \\ \theta(-1,0) \\ \theta(-1,1) \\ \theta(0,-1) \\ \theta(0,1) \\ \theta(1,-1) \\ \theta(1,0) \\ \theta(1,1) \end{bmatrix} \tag{4.5}$$

and

$$\mathbf{x}(m,n) = \begin{bmatrix} x(m-1,n-1) \\ x(m-1,n) \\ x(m-1,n+1) \\ x(m,n-1) \\ x(m,n+1) \\ x(m+1,n-1) \\ x(m+1,n) \\ x(m+1,n+1) \end{bmatrix}. \tag{4.6}$$

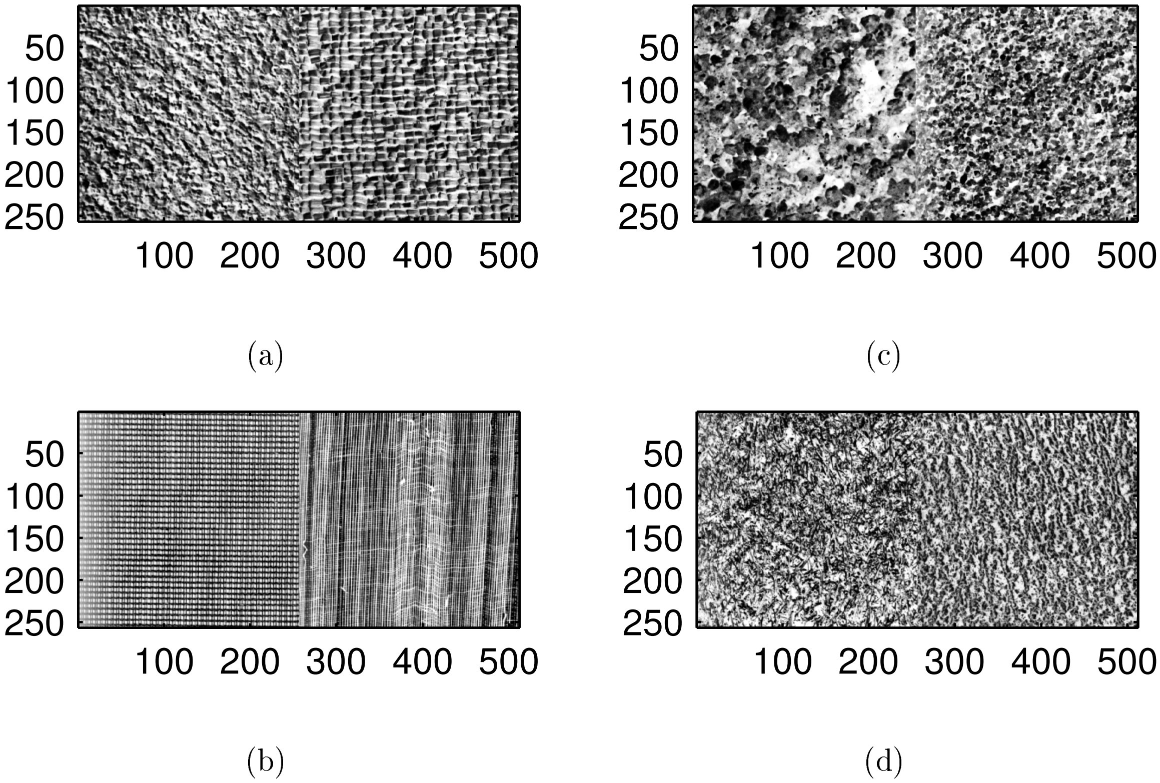


Figure 4.1: Textures pairs (a) D4-D84, (b) D21-D105, (c) D28-D29, and (d) D9-D24 [14] used in the experiments. These test textures have zero mean and have been histogram-equalized.

If we define $\mathbf{e}$ to be the vector of all $e(m,n)$, $\tilde{\mathbf{x}}$ to be the vector of all $x(m,n)$, and $\mathbf{X}$ to be the matrix with the $\mathbf{x}(m,n)^T$'s as the row vectors we may write the linear equation set determined by Equation (4.4) as

$$\mathbf{e} = \tilde{\mathbf{x}} - \mathbf{X}\boldsymbol{\theta}. \tag{4.7}$$

The $\boldsymbol{\theta}$ minimizing ε can then be found by minimizing $\mathbf{e}^T\mathbf{e}$. Finding the $\boldsymbol{\theta}$ yielding the minimum ε entails the solution of

$$\frac{\partial \mathbf{e}^T\mathbf{e}}{\partial \boldsymbol{\theta}} = 0, \tag{4.8}$$

which is found by solving the normal equations [44]

$$\mathbf{X^T X}\boldsymbol{\theta} = \mathbf{X^T}\tilde{\mathbf{x}}. \tag{4.9}$$

The normal equations can be solved effectively by Cholesky decomposition or other methods [38].

4.3 Texture feature extraction and classification

Now, we have established a framework for obtaining prediction error images, $e(m,n)$, one prediction error image corresponding to each texture. Denote the prediction error image for texture number i by $e_i(m,n)$. The first step of the feature extraction is to compute the $e_i(m,n)$-images by filtering. For the parts of an image containing texture number i, we will have

$$\sum e_i^2(m,n) \leq \sum e_j^2(m,n)\ \forall j \neq i. \tag{4.10}$$

In order to have a local estimate of this average, the $e_i^2(m,n)$-images are smoothed. It is noticed that this squaring and the subsequent smoothing correspond to the steps in the local energy function, Section 2.1.1. In the experiments reported here, the Gaussian low pass filter, Equation (2.1) with bandwidth empirically selected to be given by $\sigma_s = 8$ was applied.

Segmentation is obtained by assigning each pixel to the class corresponding to the minimum prediction error. In the experiments herein, the region of support of the predictor is circular with radius four pixels.

4.4 Results

The results given here will only be illustratory, i.e., chosen to illustrate the performance of the method. In order to make the illustrations simple, only two-texture images will be given here. Multiple texture examples are given in Chapter 6. The same texture pairs will be used in this and the next chapter for comparison purposes. Most texture pairs are well discriminated by this method and the methods of the next chapter, but a few texture pairs appear to be more problematic. The texture pairs in Figure 4.1 are used in the experiments. Figure 4.1(a) is easily discriminated by all tested approaches. The rest of the textures are chosen since they are problematic for some of the approaches, and thereby illustrate weaknesses well.

Figure 4.2 illustrates the performances of the prediction error energy features. The upper images are the features corresponding to minimum prediction error filtering for the left texture. The images in the second row correspond to the predictor for the right texture. Since the gray-levels are scaled to utilize the maximum range of the display, it is impossible to judge the distance between the features from, e.g., Figures 4.2(a) and 4.2(b). To illustrate these distances better, the graphs in the third row of Figure 4.2 are provided. These graphs show the average features per column for the left-texture predictor (solid) and the right-texture predictor (dotted). The final row of Figure 4.2 show the corresponding classification results.

The texture pairs of Figures 4.1(a) and 4.1(b) are the images with visually most distinct textures. As we see, these two textures are very well discriminated by the approach. The other two texture pairs are visually more similar and we see from Figures 4.1(k) and 4.1(o) that the predictor outputs are hardly different for the two textures. Hence, no good classification can be expected and the class maps in Figures 4.1(l) and 4.1(p) have little correspondence with the desired class maps.

A likely explanation of the breakdown for the last two images is that they are so similar that the predictor, which is optimal with respect to texture representation, do not yield discriminable features. A couple of other filtering approaches optimized with respect to texture representation have also been presented. These are the eigenfilter approach

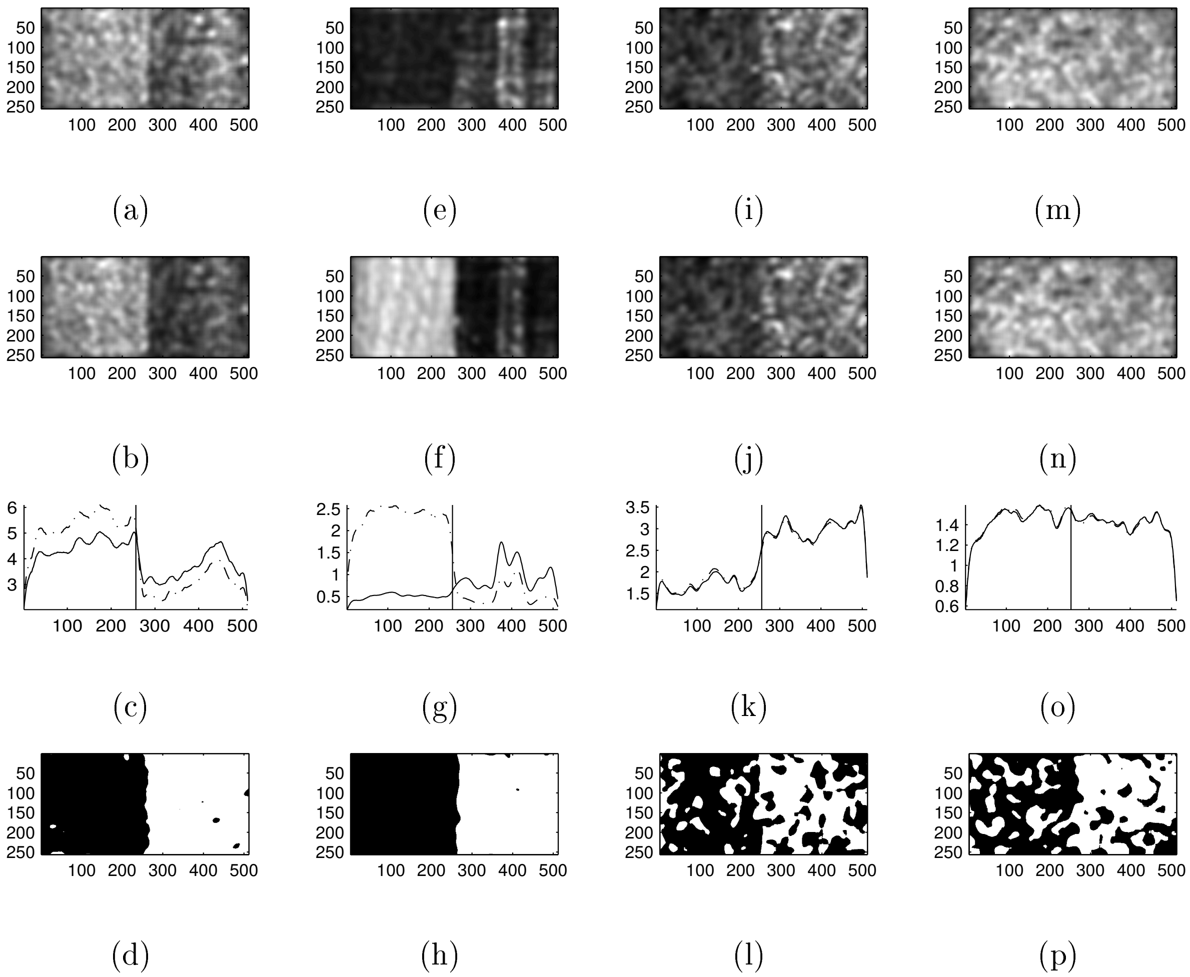


Figure 4.2: (a) Local prediction error energy for D4 in the pair D4-D84 and (b) for D84. (c) The features averaged per column (solid line corresponds to left texture, dotted to right), (d) and finally the corresponding class map. (e-h) Similar figures for D21-D105, (i-l) D28-D29, and (m-p) D9-D24. The vertical lines indicates the true texture boundaries.

of Section 2.3.1 and the optimal representation Gabor filter bank of Section 2.3.2. A discussion on possible implications of optimal representation versus optimal discrimination can be found in Section 2.3.1.

4.5 Summary

In this chapter, a new approach to texture feature extraction using optimal linear predictors has been proposed. The texture features are the local prediction error energies. Experimental results indicate that these features are suitable for segmentation for some textures, but inadequate for others. The method is optimal with respect to texture representation, thus it does *not guarantee* any *discriminable* difference in the prediction errors.

Chapter 5

Optimal discrimination filters

In this chapter, several approaches to the design of linear FIR filters with optimal energy separation are proposed. In Section 5.1 the approaches are outlined and in Section 5.2 the feature extraction model is presented. Various two-texture optimal filter design approaches are presented in Section 5.3, and in Section 5.4 a multi-texture approach is presented. In Section 5.5, an optimal filter design approach for unsupervised segmentation is presented. The chapter is summarized in Section 5.6. Mathematical details for the derivations found in this chapter are presented in Appendix A.

The feature extraction techniques developed in this chapter are advantageous in that they provide good feature separation along with less computational requirements than many other texture feature extractors. The application examples are on benchmark texture images, but the approaches should be applicable in most texture applications with a priori class knowledge.

5.1 Overview

The feature extraction system used in this chapter is illustrated in Figure 5.1. Basically, the purpose of the filter $h(m,n)$ is extraction of local frequencies where some of the textures have low signal energy and other textures have high. If this is accomplished, a composite texture image can be segmented and the regions classified by analysis of their energies. Hence, the first block of the feature extraction system illustrated in Figure 5.1 is the filter $h(m,n)$. The output, $y(m,n)$, is computed by

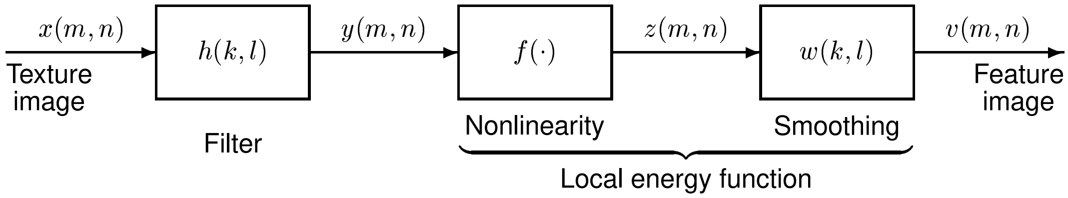


Figure 5.1: The experimental setup for the feature extractionexperiments with optimized filters.

$$y(m,n) = x(m,n) * h(m,n), \tag{5.1}$$

where $*$ denotes two-dimensional convolution. Assuming wide sense stationarity [83], the energy of a signal, y, is defined as $E\{y^2\}$ and a local energy measure is obtained by *smoothing* y^2. That is, the features are computed by nonlinearity,

$$z(m,n) = f\left(y(m,n)\right) = y^2(m,n), \tag{5.2}$$

and smoothing,

$$v(m,n) = z(m,n) * w(m,n). \tag{5.3}$$

A thresholding operation is applied for associating a class label with the extracted features.

In the system of Figure 2.1, an optional second nonlinearity was introduced. This nonlinearity is typically monotonous [132]. However, thresholding a monotonously transformed feature will yield exactly the same result as thresholding the non-transformed feature. Hence, the second nonlinearity will not have any impact on the classification result here.

In Section 5.2 a new model for the mean and variance of the features extracted with the system in Figure 5.1 is presented. This model is used for optimization of a single filter with respect to the relative distance between the average feature values,

$$J_U = \frac{\left(\mu_{v_1} - \mu_{v_2}\right)^2}{\mu_{v_1}\mu_{v_2}}, \tag{5.4}$$

for two textures. Here, μ_{v_1} and μ_{v_2} are the modeled feature means. This criterion was originally suggested by Unser [130] for designing optimal texture transforms, but has not been applied for texture filter design earlier. An *exact closed form* optimal filter solution is detailed in Section 5.3.1. A related optimization approach with respect to the ratio between the average feature values,

$$J_{MS} = \frac{\mu_{v_1}}{\mu_{v_2}}, \tag{5.5}$$

was suggested by Mahalanobis and Singh [66]. However, tests included in this chapter indicate that this is a considerably less robust criterion.

The main drawback with the *Unser criterion*, J_U, and the *Mahalanobis-Singh criterion*, J_{MS}, is that they do not take the variances of the extracted features into account. Consequently, a filter found to be optimal *may* yield a considerable *overlap* between the features. A criterion well known in the pattern recognition literature that takes the feature variances into account is the *Fisher criterion* [35],

$$J_F(\mathbf{h}) = \frac{\left(\mu_{v_1} - \mu_{v_2}\right)^2}{\sigma_{v_1}^2 + \sigma_{v_2}^2}, \tag{5.6}$$

where $\sigma_{v_1}^2$ and $\sigma_{v_2}^2$ are the feature variances. Now we need estimators for the feature means, μ_{v_i}, and variances, $\sigma_{v_i}^2$.

The framework for designing filters optimal with respect to this criterion is developed in Section 5.3.2. However, an exact closed form solution to this optimization problem has not been found, but an *approximate closed form* optimization solution is developed. Furthermore, the necessary mathematics for gradient search optimization is developed and some iterative optimization approaches are discussed in Section 5.3.3. In Section 5.3.4 common characteristics of the closed form two-texture solutions with respect to the different criteria are pointed out. A generalized solution is formulated on the basis of these common characteristics.

Inspection problems like defect versus non-defect detection may in some cases be formulated as two-texture problems. However, the range of applications of two-texture approaches is limited. To extend the usefulness, a multi-texture optimal filter bank design scheme is developed in Section 5.4, yielding an exact closed form solution based on optimization with respect to the mean values of the extracted features. Finally, an extension to unsupervised problems is suggested in Section 5.5, and the chapter is summarized in Section 5.6.

5.2 Feature extraction model

In this section, estimates of the mean and variance of the extracted features using the feature extraction scheme illustrated in Figure 5.1 are developed. The feature mean and variance models may be used for analyzing given filters or, as will be done later in this chapter, for designing optimal filters. In Subsection 5.2.1, the feature extraction process is expressed in a matrix/vector notation similar to the one we saw in Chapter 4. In Subsections 5.2.2 and 5.2.3, expressions for the feature mean and variance are developed, and in Subsection 5.2.4 assessed by experiments. We assume that the textures being filtered are wide sense stationary, but in Section 5.2.5 modeling of edge responses is discussed.

5.2.1 Vector formulation

Denote texture number i by $x_i(m,n)$ where m and n are the spatial indices to the texture image. Filtering $x_i(m,n)$ by the filter $h(m,n)$ yields

$$\begin{aligned} y_i(m,n) &= h(m,n) * x_i(m,n) \\ &= \sum_{k=0}^{M_h-1} \sum_{l=0}^{N_h-1} h(k,l) x_i(m-k, n-l), \end{aligned} \tag{5.7}$$

where $h(m,n)$ is an $M_h \times N_h$ filter. This operation corresponds to the first block of Figure 5.1. A more compact representation is obtained by introducing the following definitions,

$$\mathbf{h} = \begin{bmatrix} h(0,0) \\ \vdots \\ h(0, N_h - 1) \\ h(1,0) \\ \vdots \\ h(1, N_h - 1) \\ \vdots \\ h(M_h - 1, N_h - 1) \end{bmatrix}, \tag{5.8}$$

i.e., the vector formed by lexicographically ordering of the rows of $h(m, n)$, and

$$\mathbf{x}_i(m,n) = \begin{bmatrix} x_i(m,n) \\ \vdots \\ x_i(m, n - N_h + 1) \\ x_i(m-1, n) \\ \vdots \\ x_i(m-1, n - N_h + 1) \\ \vdots \\ x_i(m - M_h + 1, n - N_h + 1) \end{bmatrix}. \tag{5.9}$$

Then, the output at pixel (m, n) for texture i, Equation (5.7), may be rewritten as

$$y_i(m,n) = \mathbf{h}^T \mathbf{x}_i(m,n). \tag{5.10}$$

Similarly, the smoothing operation (see Figure 5.1) may be defined as

$$v_i(m,n) = \mathbf{w}^T \mathbf{z}_i(m,n), \tag{5.11}$$

where $\mathbf{w}$ corresponds to the smoothing filter, $w(m, n)$. This formulation may easily be extended to filter masks, $h(m, n)$, of any shape, but for simplicity, only filter masks with a rectangular region of support are considered here.

5.2.2 Feature mean

Assume that the input texture is wide sense stationary and that $\sum_{m,n} w(m,n) = 1$. Then, using the vector formulation, the expected feature value at the output of the filter is

$$\begin{aligned} \mu_{v_i} &= E\left\{v_i(m,n)\right\} = E\left\{w(m,n) * z_i(m,n)\right\} \\ &= E\left\{z_i(m,n)\right\} = E\left\{y_i(m,n) y_i(m,n)\right\} \\ &= E\left\{\left(\mathbf{h}^T \mathbf{x}_i(m,n)\right)\left(\mathbf{x}_i^T(m,n)\mathbf{h}\right)\right\} \\ &= (\mathbf{h}^T \mathbf{R}_{x_i x_i} \mathbf{h}), \end{aligned} \tag{5.12}$$

where

$$\mathbf{R}_{x_i x_i} = E\left\{\mathbf{x}_i(m,n)\mathbf{x}_i^T(m,n)\right\}. \tag{5.13}$$

By writing the elements of $\mathbf{R}_{x_i x_i}$ explicitly, it is readily seen that $\mathbf{R}_{x_i x_i}$ is constructed from the two-dimensional autocorrelation function of texture x_i. The derivative of μ_{v_i} is given by [118]

$$\frac{\partial \mu_{v_i}}{\partial \mathbf{h}} = \frac{\partial(\mathbf{h}^T \mathbf{R}_{x_i x_i}\mathbf{h})}{\partial \mathbf{h}} = 2\mathbf{R}_{x_i x_i}\mathbf{h}. \tag{5.14}$$

5.2.3 Feature variance

If we assume that $y_i(m,n)$ is Gaussian (assessed in Section 5.2.4), the variance of the feature image $v_i(m,n)$, $\sigma_{v_i}^2 = E\left\{(v_i(m,n) - \mu_{v_i})^2\right\}$, may be written

$$\sigma_{v_i}^2 = 2\sum_{k_1=0}^{M_w-1}\sum_{l_1=0}^{N_w-1} w(k_1,l_1)\sum_{k_2=0}^{M_w-1}\sum_{l_2=0}^{N_w-1} w(k_2,l_2)\left(\mathbf{h}^T\mathbf{R}_{x_i x_i}(k_1-k_2,l_1-l_2)\mathbf{h}\right)^2 - 2{\mu_{y_i}}^4, \tag{5.15}$$

where

$$\mathbf{R}_{x_i x_i}(k,l) = E\left\{\mathbf{x}_i(m,n)\mathbf{x}_i^T(m+k,n+l)\right\}, \tag{5.16}$$

$$\mu_{y_i} = \mu_{x_i}\mathbf{1}^T\mathbf{h}^4, \tag{5.17}$$

$\mu_{x_i} = E\{x_i(m,n)\}$, and $\mathbf{1} = [1\ 1 \cdots 1]^T$. The smoothing filter, $w(m,n)$, is of size $M_w \times N_w$. The derivation is given in Appendix A.2, along with the expression for the partial derivative of $\sigma_{v_i}^2$ with respect to $\mathbf{h}$.

In Appendix A.4 an alternate and more compact representation of the variance is given. This representation has, however, not proven successful for filter optimization yet.

5.2.4 Model assessment

The variance estimate above was based on the assumption that the output of the filter was Gaussian. This is an approximation which is generally not true. However, distributions for several filters and several real world textures from the Brodatz album [14] were tested by plotting the histograms of $y_i(m,n)$. Examples of some histograms for $y_i(m,n)$ are shown in Figure 5.2.[1] As we see, the histograms fit the Gaussian assumption (solid line) quite well.

How good are the total mean and variance models? In all experiments with the estimated feature mean, 1:1 matches between estimated and true were found. Slightly larger errors were encountered with the variance estimation. The true and estimated variances for a number of textures with several filters having Gaussian random drawn

[1] These distributions were generated by optimized filters of Section 5.3.5, Figure 5.8.

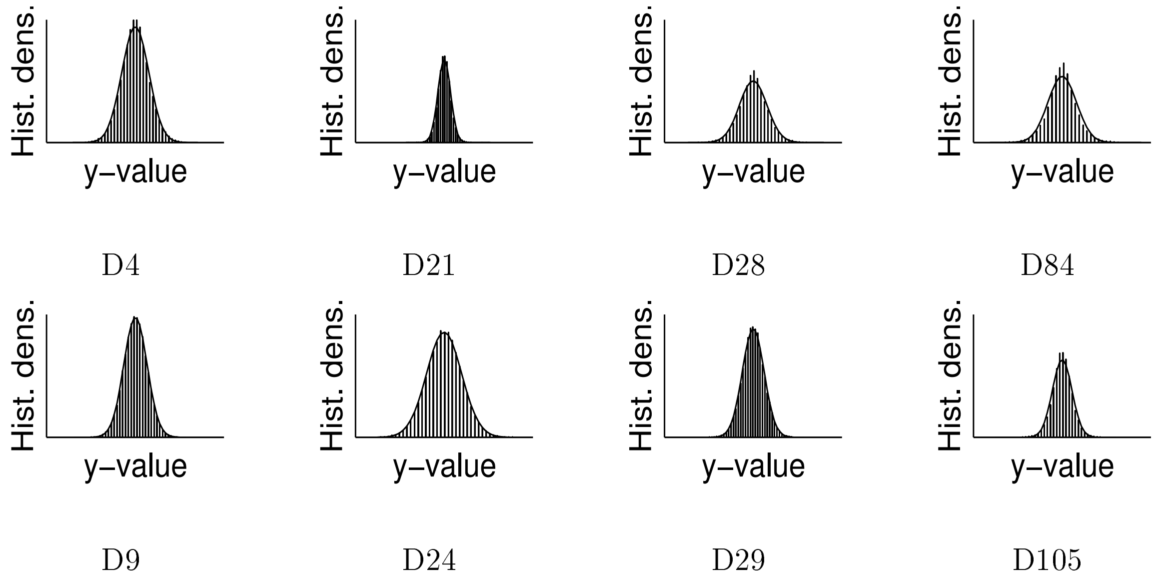


Figure 5.2: Filter response histograms from filters designed in the experiments illustrated in Figure 5.8, i.e., closed form optimization with respect to the Fisher criterion. The continuous curves illustrate the corresponding Gaussian distributions with the same means and variances.

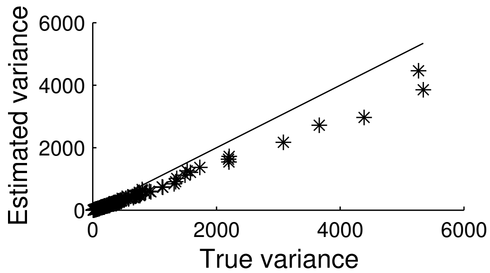


Figure 5.3: Estimated feature variance (vertical axis) versus true feature variance for a number of random drawn filters $h(m, n)$ and the Brodatz texture D4.

coefficients were tested. An example where this is apparent is shown in Figure 5.3. In this experiment, filters of size 7×7 were designed. The smoothing filter, $w(m, n)$, was Gaussian, Equation (2.1), with $\sigma_s = 8$ having a spatial extent of $M_w \times N_w = 33 \times 33$. The test image was the texture D4 [14]. As we see, the ratio between the true and modeled feature variance is approximately constant. A large number of textures were applied, and apparent linear relations were found in all cases, but the ratio was different for different textures. One possible explanation for this observation is that it is caused by problems with the estimates of the textures' autocorrelation functions. Since the autocorrelation function is typically estimated from an image of finite size, there is a certain variance in the estimate depending on the image size. Furthermore, real world textures are typically non-stationary, thus estimating the autocorrelation with the wide sense stationary assumption imposes some error in the autocorrelation estimate. Hence, reliable mean and variance estimates can only be expected if the autocorrelation function of the texture is estimated from a large stationary image.

The estimate for the mean value of $y_i(m,n)$, μ_{y_i}, was of a similar quality as the mean estimate for the features, $v_i(m,n)$. However, since the expression for $\sigma^2_{v_i}$ contains ${\mu_{y_i}}^4$ (see Equation (5.15)), even a small deviation in the estimate for μ_{y_i} may yield a considerable error in the estimate of $\sigma^2_{v_i}$. Due to the linearity of the filter, μ_{y_i} is a linear function of the mean of the input texture, see Equation (5.17). Hence, if the feature variance is small compared to the mean of the input texture, the variance estimate is likely to be unreliable. This is not considered to be a major limitation because if the mean values of two homogeneous patterns differ significantly, they are easily discriminated by smoothing. In such a case, texture techniques are not necessary. On the other hand, if the means are close, the average mean value may be subtracted prior to design and filtering.

5.2.5 Texture edge and boundary model

When a filter is optimized with respect to any of the proposed criteria, it is assumed that the textures are wide sense stationary. However, wide sense stationarity is at best approximated *within* a texture, but clearly, the edges between the textures impose non-stationarities. No optimization with respect to edge effects is done.

If the feature extraction system responds very strongly to the *edge* between two textures, the detected edge position will be biased and the region corresponding to the largest expected feature value will become too large. An example where this is apparent is shown in Figure 5.4, taken from the experiments of Section 5.5. In Figure 5.4(a) we see the feature image corresponding to an optimized filter. The true texture boundary is indicated by a black line. If examined carefully, we see that the feature response is *stronger* along the texture edge. Consequently, the class map generated by thresholding, Figure 5.4(b), is biased and the center texture is consistently too small. In this subsection, a technique for modeling the feature response near the texture edges is proposed. This model may be utilized for predicting the edge localization bias and determining means for correcting it.

From Section 5.2.2, we know that the mean feature value of texture i may be modeled as

$$\mu_{v_i} = \mathbf{h}^T \mathbf{R}_{x_i x_i} \mathbf{h}, \tag{5.18}$$

assuming wide sense stationarity. When all elements of $\mathbf{x}_i(m,n)$ are taken from one texture, $\mathbf{R}_{x_i x_i}$ is constructed from the two-dimensional autocorrelation function of the texture $x_i(m,n)$. However, when computing the expected feature value somewhere on the edge between two textures, the WSS assumption is not any more appropriate. We then have

$$\begin{aligned} \mu_{v_{edge}}(m,n) &= \sum_{k=0}^{N_w-1} \sum_{l=0}^{N_w-1} w(k,l) E\left\{z(m-k,n-l)\right\} \\ &= \sum_{k=0}^{N_w-1} \sum_{l=0}^{N_w-1} w(k,l) \mathbf{h}^T \mathbf{R}_{xx}(k,l) \mathbf{h}. \end{aligned} \tag{5.19}$$

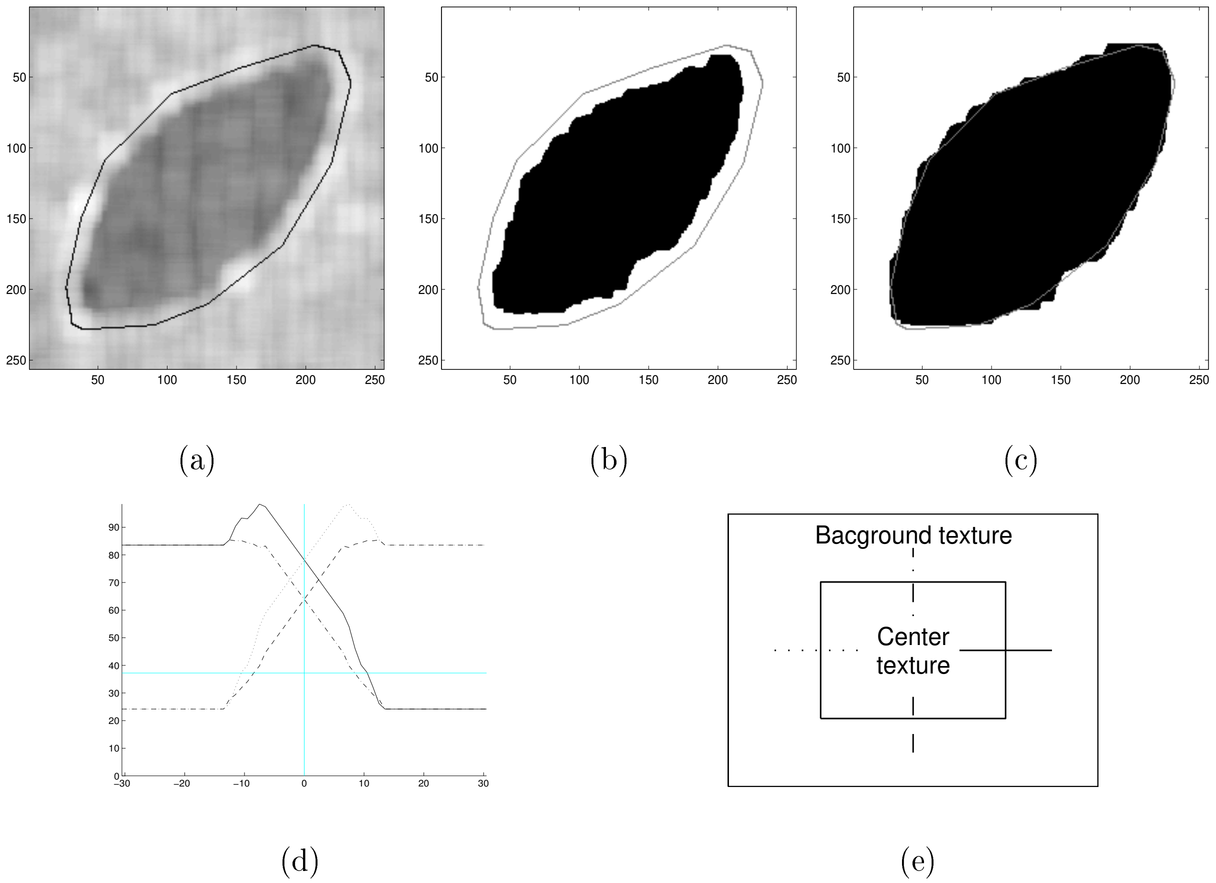


Figure 5.4: Illustration of edge modeling and compensation. (a) Feature image with biased edge response and (b) corresponding minimum error class map. (c) Region map images after edge effect compensation by morphological dilation. (d) Edge modeling of the feature values across a texture edge. The horizontal axes represent pixel numbers relative to the edge. The vertical lines indicate the true texture boundaries and the horizontal lines indicate the selected feature threshold values. (e) Descriptions of the line styles in (d): Center texture right of background texture (—), center left of background ($\cdots$), center below background ($-\cdot$), and center above background ($--$). Lines in the images indicate the true texture boundaries.

If the elements of $\mathbf{R}_{xx}(k,l)$ are written explicitly for a given edge location, it is readily seen that $\mathbf{R}_{xx}(k,l)$ may be created from the auto- and cross-correlation functions of the two textures. Using this, we can estimate the feature responses along edges of any desired orientation and shape. Examples of such edge response estimates are shown in Figure 5.4(c).

The class maps are typically determined from the optimal features by thresholding. The minimum error threshold in this case is illustrated by the horizontal line in Figure 5.4(c). If we look at these graphs, we see that vertical edges are biased by 8 pixels and horizontal by 11. To overcome this problem, the center region is extended with morphological dilation [39] with a rectangular kernel extending the region 8 pixels vertically and 11 horizontally, i.e., an $(8+1+8)\times(11+1+11) = 17\times 23$ rectangular kernel. The resulting class map for the example is shown in Figure 5.4(d), and the match with the true texture boundary is very good.

The strong feature response at the texture edges are due to the fact that the artificial texture boundary has abrupt pixel value transitions. Hence, from Fourier theory, we know that the edge has considerable high spatial frequency content. In this experiment, the texture filter had high-boost properties. Consequently, the boundary was emphasized. It is likely that many real world texture mosaics have less abrupt edges, yielding a smoother feature transition. Furthermore, in this experiment, the smoothing filter, $w(m,n)$, was a rectangular filter.[2] However, it is recommended to use a Gaussian smoothing filter instead of a rectangular, since the Gaussian will yield better edge accuracy. Consequently, for real world texture mosaics using Gaussian smoothing filters, the edge bias is probably less distinct. Hence, edge modeling may not be necessary in all circumstances.

5.3 Two-texture optimization approaches

In this section, several two-texture one-filter optimization approaches will be presented. In Subsection 5.3.1 an exact closed form solution for a filter maximizing the relative average feature distance is developed, in Subsection 5.3.2 an approximate closed form optimal solution with respect to the Fisher criterion is developed, and in Subsection 5.3.3 iterative approaches to optimization with respect to the Fisher criterion are discussed. Common characteristics of the closed form solutions are pointed out in Subsection 5.3.4 and finally in Subsection 5.3.5 some experiments are presented.

5.3.1 Optimization with respect to relative average feature distance

A classification algorithm is only successful if the input features for the different classes are discriminable. Optimizing with respect to the relative distance between the average features,

$$J_U(\mathbf{h}) = \frac{(\mu_{v_1} - \mu_{v_2})^2}{\mu_{v_1}\mu_{v_2}}, \tag{5.20}$$

is one way of attempting this. Optimization with respect to J_U is easily developed and entails the solution of

$$\frac{\partial J_U(\mathbf{h})}{\partial \mathbf{h}} = 0. \tag{5.21}$$

The chain rule for differentiation yields

$$\frac{\partial J_U(\mathbf{h})}{\partial \mathbf{h}} = \frac{\partial J_U(\mathbf{h})}{\partial \mu_{v_1}}\frac{\partial \mu_{v_1}}{\partial \mathbf{h}} + \frac{\partial J_U(\mathbf{h})}{\partial \mu_{v_2}}\frac{\partial \mu_{v_2}}{\partial \mathbf{h}}. \tag{5.22}$$

Combining Equations (5.12), (5.14), and (5.22), we get the eigenvalue equation

$$(\mathbf{R}_{x_2x_2}{}^{-1}\mathbf{R}_{x_1x_1})\mathbf{h} = \lambda\mathbf{h}, \tag{5.23}$$

[2] This choice was made in order to make a distinct edge modeling example feature image. A large edge bias is easier illustrated than a subtle one.

where

$$\lambda = \frac{(\mathbf{h}^T \mathbf{R}_{x_1 x_1} \mathbf{h})}{(\mathbf{h}^T \mathbf{R}_{x_2 x_2} \mathbf{h})}. \tag{5.24}$$

The optimal filter coefficient vector is found as the eigenvector of $(\mathbf{R}_{x_2 x_2}{}^{-1} \mathbf{R}_{x_1 x_1})$ with the maximum corresponding criterion value $J_U(\mathbf{h})$. The proof that this is a *valid* solution to the optimization problem, i.e., that all eigenvectors $\mathbf{h}$ will satisfy Equation (5.24) where λ is the corresponding eigenvalue is given in Appendix A.1.

It is noted that this approach yielded an exact closed form optimal solution. This guarantees that the global optimum is attained, in contrast to some of the other approaches that will be presented in this chapter. However, the major conceptual drawback with this method is the fact that the variances are not included in the criterion. Hence, the optimized filter may yield a large feature mean separation, but with large feature variances. Hence, the density functions for the features for the two textures may have a considerable overlap, yielding a poor classification.

Optimization with respect to the Mahalanobis-Singh approach, Equation (5.5), proceeds similarly [66], but yields far less robust features as indicated by the experiments, Subsection 5.3.5. One plausible explanation why the Unser criterion, J_U, yields better features than the Mahalanobis-Singh criterion is obtained by rewriting the criterion as $J_U(\mathbf{h}) = \mu_{v_1}/\mu_{v_2} + \mu_{v_2}/\mu_{v_1} - 2$. Even if it is inherently difficult designing a filter yielding high feature values for texture 1 and low for texture 2,[3] the Mahalanobis-Singh criterion will lead to the selection of a filter attempting to emphasize texture 1, i.e., maximize μ_{v_1}/μ_{v_2}. However, in this case, the Unser criterion may lead to the selection of a filter emphasizing texture 2 so that the term μ_{v_2}/μ_{v_1} is large.

5.3.2 Optimization with respect to the Fisher criterion

A good feature extractor should not only yield a large distance between the mean values of the extracted features, but also yield low feature variances. This is exactly what is expressed by the Fisher criterion [35],

$$J_F(\mathbf{h}) = \frac{(\mu_{v_1} - \mu_{v_2})^2}{\sigma_{v_1}^2 + \sigma_{v_2}^2}, \tag{5.25}$$

well known in pattern recognition.

As with the previous approach, in order to find the optimal filter, the partial derivative of the criterion, $\partial J_F(\mathbf{h})/\partial \mathbf{h}$, is equated to zero. However, by inserting the expressions for the mean, variance, and their derivatives, Equations (5.12), (5.14), (5.15), and (A.16), a general closed form solution has not been found. Then we have two alternatives, either to do some approximation and find an approximate closed form solution or to use some iterative search for the optimum.

[3] This may for example be the case if the energy of texture 1 is generally low compared to texture 2.

By approximating the textures as separable [48][4] autoregressive fields [83] of order one, AR(1),[5] a simplified variance expression is obtained in Appendix A.3,

$$\sigma_{v_i}^2 \approx 2\rho_i(\mathbf{h}^T\mathbf{R}_{x_ix_i}\mathbf{h})^2, \tag{5.26}$$

where ρ_i is a scalar with empirically $\partial\rho_i/\partial\mathbf{h} \approx 0$. Substituting the mean and approximated variance expressions and the corresponding derivatives into $\partial J_F(\mathbf{h})/\partial\mathbf{h}$ yields

$$\begin{aligned}&\left(\rho_2(\mathbf{h}^T\mathbf{R}_{x_2x_2}\mathbf{h})^2 + \rho_1(\mathbf{h}^T\mathbf{R}_{x_1x_1}\mathbf{h})(\mathbf{h}^T\mathbf{R}_{x_2x_2}\mathbf{h})\right)\mathbf{R}_{x_1x_1}\mathbf{h}\\ &= \left(\rho_1(\mathbf{h}^T\mathbf{R}_{x_1x_1}\mathbf{h})^2 + \rho_2(\mathbf{h}^T\mathbf{R}_{x_1x_1}\mathbf{h})(\mathbf{h}^T\mathbf{R}_{x_2x_2}\mathbf{h})\right)\mathbf{R}_{x_2x_2}\mathbf{h}.\end{aligned} \tag{5.27}$$

This equation may be simplified to

$$\left(\mathbf{R}_{x_2x_2}{}^{-1}\mathbf{R}_{x_1x_1}\right)\mathbf{h} = \lambda\mathbf{h}, \tag{5.28}$$

where

$$\lambda = \frac{(\mathbf{h}^T\mathbf{R}_{x_1x_1}\mathbf{h})}{(\mathbf{h}^T\mathbf{R}_{x_2x_2}\mathbf{h})}. \tag{5.29}$$

It is noted that this solution is identical to the eigen-problem in the previous approach. The optimal filter coefficient vector is found by calculating the eigenvectors of the matrix $\left(\mathbf{R}_{x_2x_2}{}^{-1}\mathbf{R}_{x_1x_1}\right)$ and selecting the vector maximizing the criterion $J_F(\mathbf{h})$. This approach is similar to the optimization with respect to $J_U(\mathbf{h})$ in that

1. the eigenvectors of $\left(\mathbf{R}_{x_2x_2}{}^{-1}\mathbf{R}_{x_1x_1}\right)$ are found and
2. the eigenvector yielding maximum criterion is selected.

However, different criterion functions are used to select the eigenvectors in the two approaches. Hence, different eigenvectors *may* be selected.

The separable AR(1) approximation is a very rude approximation, but at least it enables us to find a closed form solution which experimentally proves to be very useful. The discussion of the optimization attempt based on the alternative variance expression in Appendix A.4 should be considered for further research on this problem.

5.3.3 Iterative Fisher optimization

An alternative to finding the closed form solution to an approximate Fisher criterion function, as in the previous subsection, is to use some iterative approach optimizing a less simplified criterion function. *Gradient search* is one potential optimization technique. The fact that we have expressions for the derivative of the criterion function is a distinct advantage with the gradient search technique. However, besides time consumption, the

[4] A separable image may be expressed by separate one-dimensional generating equations in the vertical and horizontal dimensions.

[5] A one-dimensional AR(1) function $x(n)$ may be written $x(n) = -ax(n-1) + w(n)$, where $w(n)$ is white noise (often called innovation process), and a is a constant.

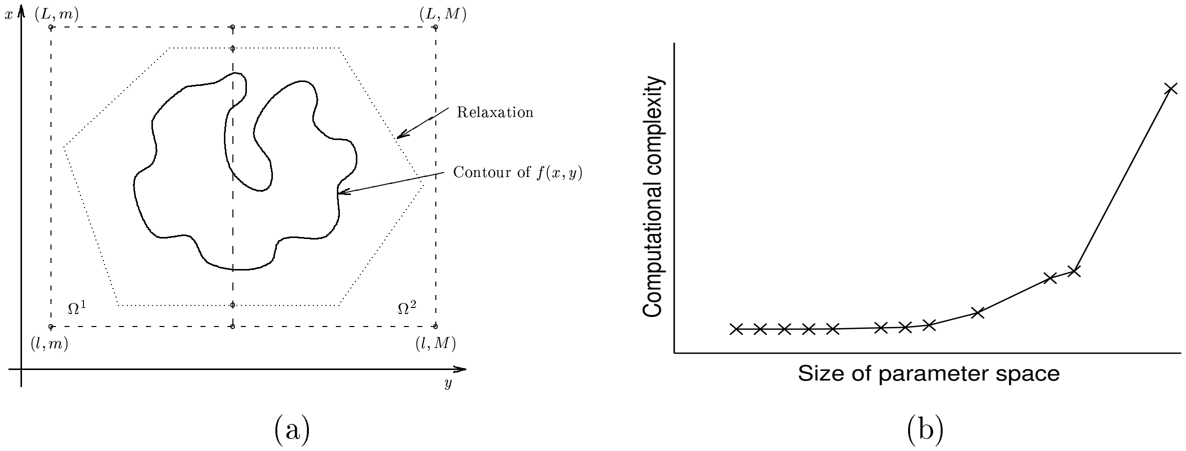


Figure 5.5: Exact iterative optimization: (a) Illustration of the principle of linear relaxation. (b) Computation time versus number of parameters in the system.

major drawback with gradient search is that it is likely to converge to a local optimum. Tests with random start vectors indicated that this is indeed the case. On the other hand, success using the closed form solution as starting point was obtained with results given in Subsection 5.3.5. Experiments with *simulated annealing* and *annealing evolution* (a combination of genetic algorithms and simulated annealing) have also been conducted [77, 78], but with less success.

Another iterative approach is *exact deterministic* technique, as used by Ramampiaro [95]. In that approach, the Fisher criterion is reformulated and represented by a quadratic programming problem. This allows search for the optimum by the branch and bound algorithm [69]. The branch and bound algorithm uses a linear relaxation. An example of a linear relaxation of a contour $f(x, y)$ is shown in Figure 5.5(a). This relaxation is used to compute the bounds on the object function. The solution is found by successively tightening the bounds and hierarchically partitioning the domain into several subspaces.

The complexity of the solution increases rapidly with the number of parameters, as illustrated in Figure 5.5(b). Both the sizes of the filter $h(m, n)$ and $w(m, n)$ contribute to the number of parameters [95]. Optimization with realistic sizes of the filters was practically impossible due to the computational complexity. In the experiments the technique was in practice limited to optimization of filters no larger than 3×3 with 3×3 smoothing filters. Such small smoothing filters are clearly inadequate and in the experiments larger filters were *applied*, although the design was with modeling of small smoothing filters. This limits the utility of the method, however, one interesting conclusion may be drawn from the experiments [95]: Using no prior knowledge of the closed form solution, the iterative technique yielded filters with about the same criterion values as the closed form solutions. This was, for example, far from the case with the gradient search, when random start vectors were used. Hence, we may conclude that the optimization approach has a large potential, but that further work is required to make it practically applicable.

5.3.4 Unified approach

As we have seen, the closed form solutions presented thus far yield the same eigen-problem, $(\mathbf{R}_{x_2x_2}{}^{-1}\mathbf{R}_{x_1x_1})\mathbf{h} = \lambda\mathbf{h}$, where the optimal filter coefficient vector is the eigenvector with maximum criterion value. Indeed, the same eigenproblem is also the solution to the *generalized* criterion function

$$J = \frac{\nu_1{\mu_{v_1}}^2 + \nu_2\mu_{v_1}\mu_{v_2} + \nu_3{\mu_{v_2}}^2 + \nu_4\sigma_{v_1}^2 + \nu_5\sigma_{v_2}^2}{\xi_1{\mu_{v_1}}^2 + \xi_2\mu_{v_1}\mu_{v_2} + \xi_3{\mu_{v_2}}^2 + \xi_4\sigma_{v_1}^2 + \xi_5\sigma_{v_2}^2}\mathbf{h}, \tag{5.30}$$

where the ν_is and ξ_is are arbitrary scalar constants. All the criterion functions considered thus far are special cases of this formulation. Given the same assumptions and following the same steps as in Section 5.3.2, the closed form optimization yields the eigenproblem

$$\left(\mathbf{R}_{x_2x_2}{}^{-1}\mathbf{R}_{x_1x_1}\right)\mathbf{h} = \frac{(\mathbf{h}^T\mathbf{R}_{x_1x_1}\mathbf{h})}{(\mathbf{h}^T\mathbf{R}_{x_2x_2}\mathbf{h})}\mathbf{h}. \tag{5.31}$$

As we see, this solution is independent of the ν_i's and ξ_i's. The proof is straightforward, but tedious. Note that if we omit the variance terms, the only assumption we have to make to obtain the closed form solution is that the input textures are wide sense stationary.

5.3.5 Experiments

Several homogeneous texture pairs have been used as test images with the methods presented in this chapter. The ones selected for presentation illustrate the performances well. The test images are shown in Figure 4.1. In all experiments, the analyzed textures have zero mean and have been histogram equalized. An $M_h \times N_h = 7 \times 7$ filter, $h(m,n)$, is designed, letting the smoothing filter, $w(m,n)$, be Gaussian, Equation (2.1), with $\sigma_s = 8$ and of spatial extent $M_w \times N_w = 33 \times 33$. Note that the size of the smoothing filter is a tradeoff between feature variations and edge accuracy. This smoothing filter was determined empirically.

The results with the Unser criterion are shown in Figure 5.6 and the Mahalanobis-Singh criterion in Figure 5.7. The Fisher criterion closed form and gradient search optimized results are shown in Figures 5.8 and 5.9, respectively. In the feature plots, the true texture border is indicated by a discontinuity in the surface. The class maps were made by thresholding with a threshold yielding minimum classification error. The results are summarized in Table 5.1 and a few feature histograms are shown in Figure 5.10. The gradient search overall doubled the estimated Fisher criterion value over the closed form solution. However, due to the generally marginal classification error, typically only marginal improvements were observed on the classification error. In one case the classification accuracy was even reduced. See Section 5.2.5 for a further discussions on edge responses.

As we can easily see, the Mahalanobis-Singh criterion optimal features have considerable overlap in Figures 5.7(e) and 5.7(g), while the Unser criterion optimal features have too much overlap in Figure 5.6(g). Clearly, these optimization approaches yield features that are non-discriminable for the D9-D24 texture.

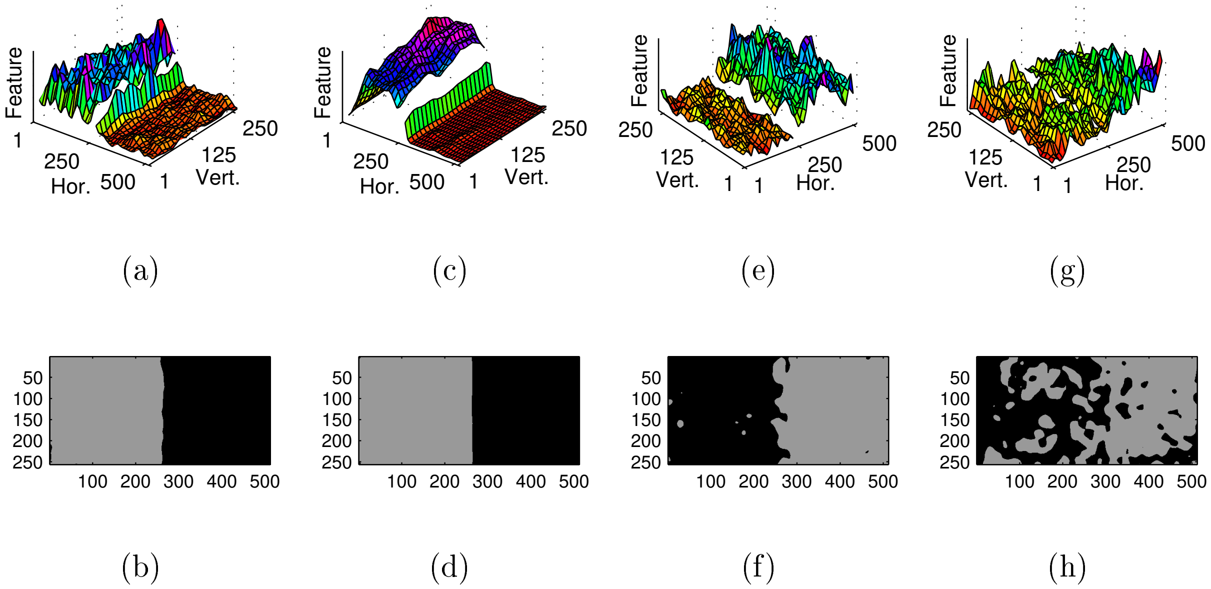


Figure 5.6: Features optimal with respect to J_U, Equation (5.4) for the textures (a) D4-D84, (c) D21-D105, (e) D28-D29, and (g) D9-D24 and their corresponding class maps (b), (d), (f), and (h). The true texture boundaries are indicated by discontinuities on the feature energy surfaces and the horizontal and vertical axes of the texture images, Figure 4.1, are indicated by "Hor." and "Vert." on the surface plots.

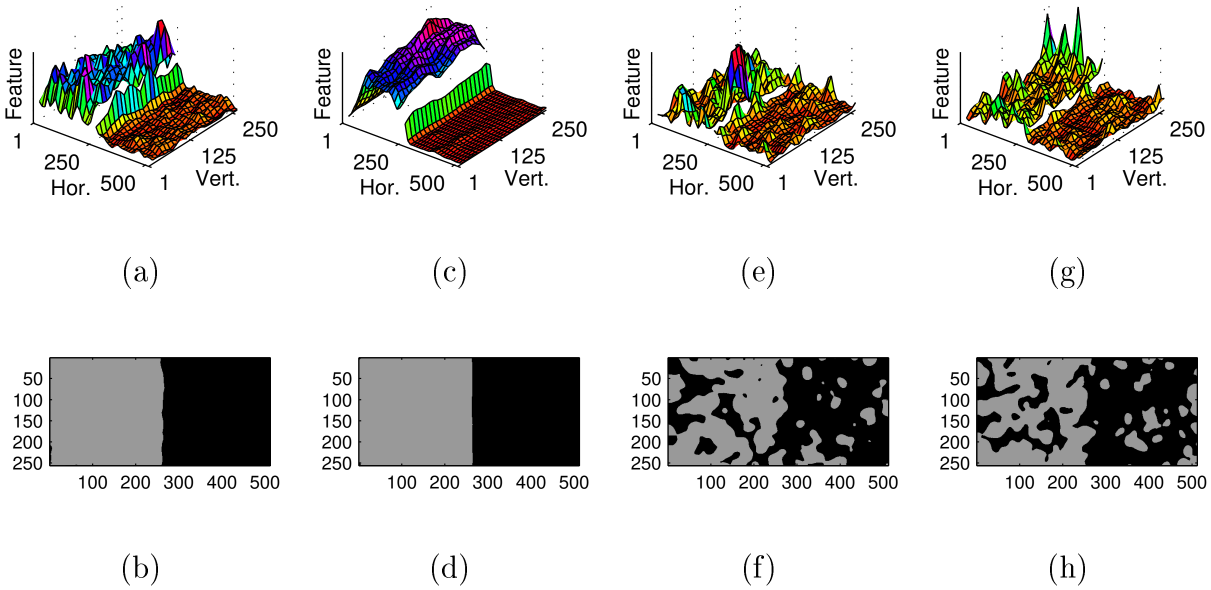


Figure 5.7: Features optimal with respect to J_{MS}, Equation (5.5) [66] for the textures (a) D4-D84, (c) D21-D105, (e) D28-D29, and (g) D9-D24 and their corresponding class maps (b), (d), (f), and (h).

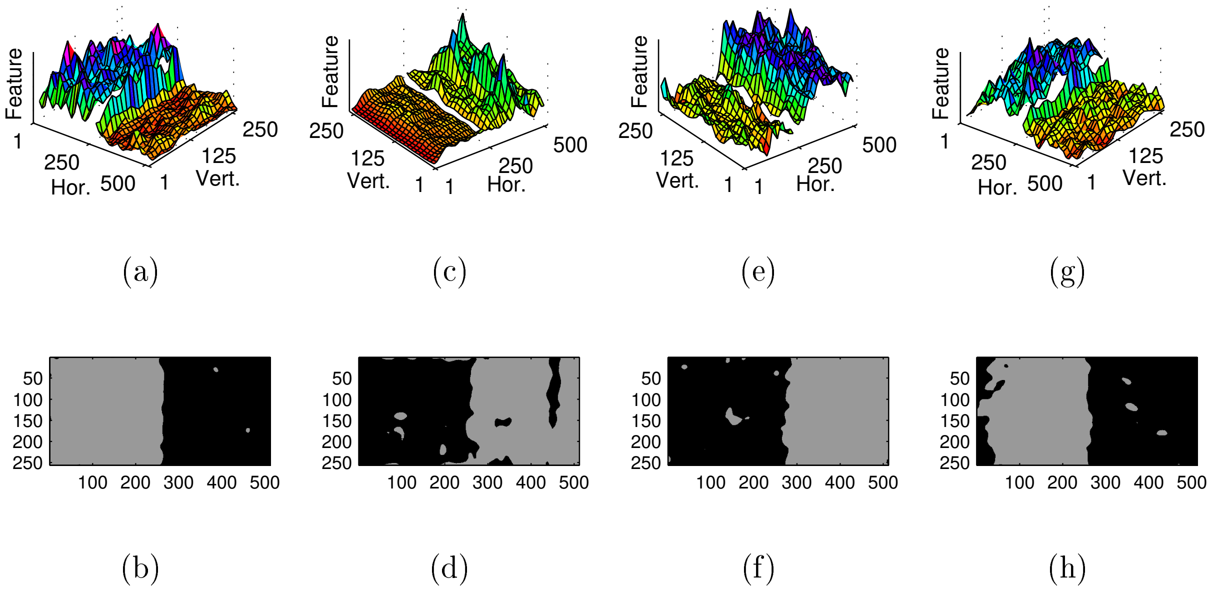


Figure 5.8: Features optimized with closed form solution for J_F, Equation (5.25), for the textures (a) D4-D84, (c) D21-D105, (e) D28-D29, and (g) D9-D24 and their corresponding class maps (b), (d), (f), and (h).

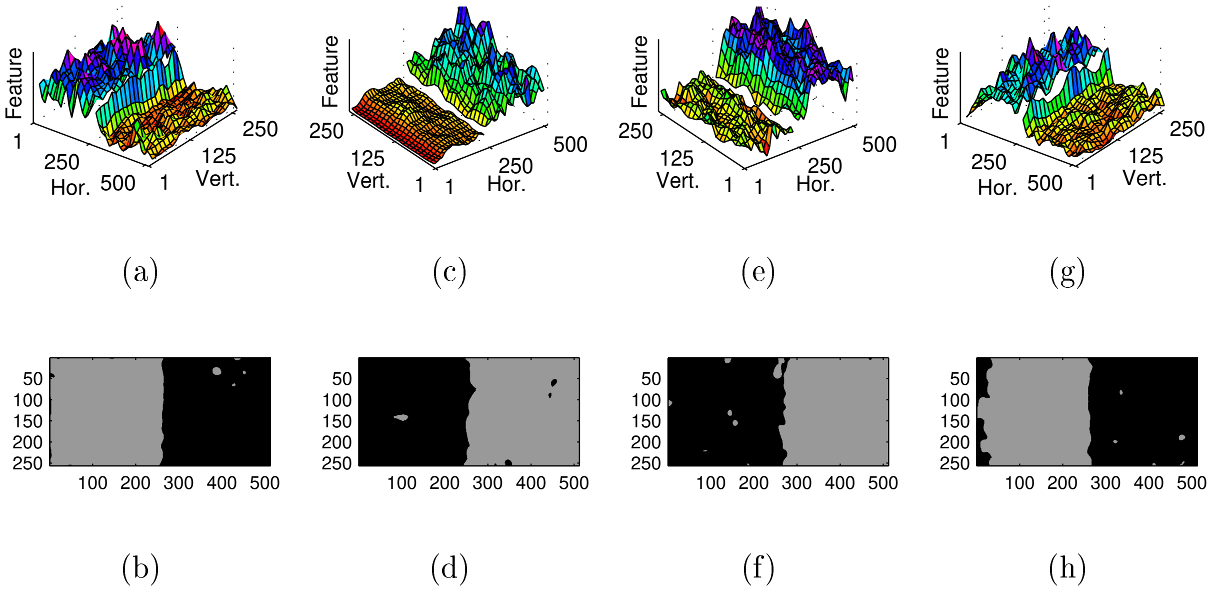


Figure 5.9: Features optimized with gradient search for J_F, Equation (5.25), using the closed form solution as the starting point, for the textures (a) D4-D84, (c) D21-D105, (e) D28-D29, and (g) D9-D24 and their corresponding class maps (b), (d), (f), and (h).

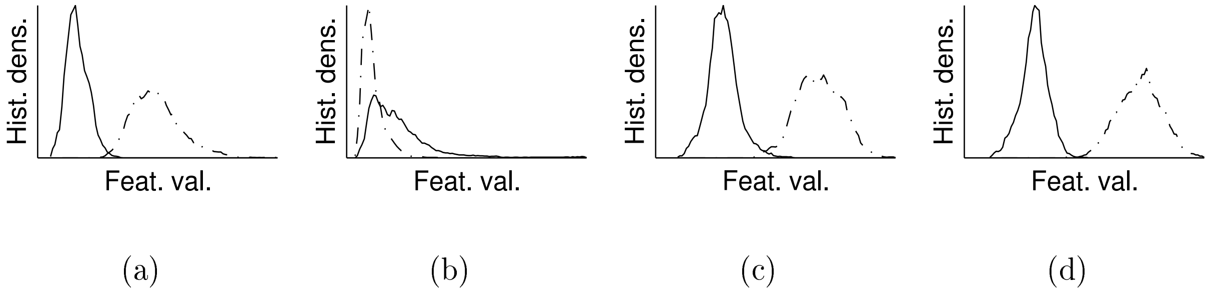


Figure 5.10: Feature histograms visualizing feature separation for texture pair D28-D29 for filters designed with respect to (a) J_U, (b) J_{MS}, (c) J_F closed form, and (d) J_F gradient search.

Criterion	D4-D84	D21-D105	D28-D29	D9-D24
Unser	1.3%	1.4%	2.7%	26.4%
Mahalanobis-Singh [66]	1.3%	1.4%	24.6%	23.9%
Fisher, closed form	1.4%	7.4%	4.6%	7.3%
Fisher, gradient	1.9%	1.6%	3.0%	5.2%
Fisher, exact iterative $M_h \times N_h = 3 \times 3$				6.0%
Min. histogram overlap	1.3%	1.4%	2.7%	7.3%
Dunn [30]	2.8%	1.0%	25.4%	18.7%

Table 5.1: Classification errors for all experiments reported in this chapter.

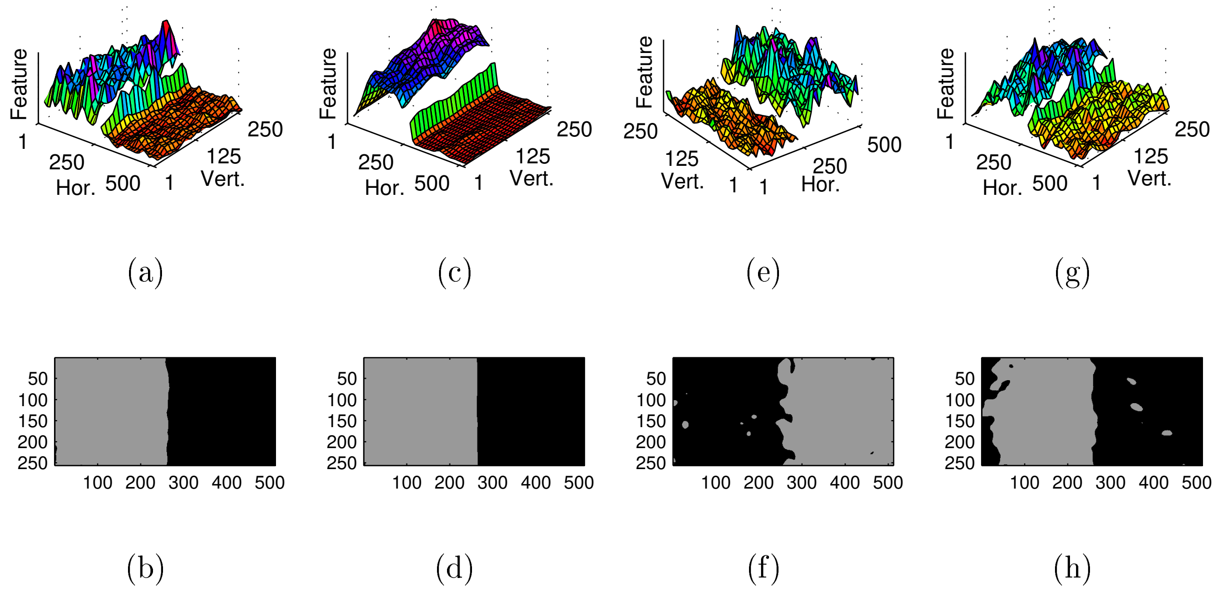


Figure 5.11: Features corresponding to optimal filters selected yielding minimum histogram overlap for the textures (a) D4-D84, (c) D21-D105, (e) D28-D29, and (g) D9-D24 and their corresponding class maps (b), (d), (f), and (h).

Not even the Fisher criterion is ideal for all cases, as is illustrated in the results for D21-D105, Figure 5.8(c). For this image, we see that the features extracted according to the J_U and J_{MS} criteria are significantly better separated than the features extracted according to the J_F criterion with closed form solution. The reason for the poor performance of the Fisher criterion in this case is unclear. However, it is certain that the Fisher solution utilizes larger lags in the autocorrelation function than the other approaches. Larger lags are more susceptible to noise due to the size of the training data and due inhomogeneities. This may yield inaccurate variance estimates, and consequently inaccurate criterion evaluation.

As we will see later, the Unser criterion actually yields best features in most cases, with the Fisher criterion in second place.[6] However, it is evident that none of the criteria presented thus far are better than the rest for selection of the eigensolution in *all* cases. An alternative approach may be to apply all filters corresponding to the eigenvectors of $\left(\mathbf{R}_{x_2x_2}{}^{-1}\mathbf{R}_{x_1x_1}\right)$ and select the one yielding *maximum separation of the feature histograms*. This solution is motivated by the unified solution in Subsection 5.3.4. The results using this approach are shown in Figure 5.11 and in Table 5.1. This approach will be equal to or better than any of the closed form experiments in all cases. However, filters optimized without the closed form solution, e.g., gradient search, may be better. The texture pair D9-D24 is an example of this. Here, the gradient search yields 5.2% error, while the best closed form solution yields 7.3%.

A limited experiment with the exact iterative optimization, Section 5.3.3, with respect to the Fisher criterion has also been performed. The results are shown in Figure 5.12. Due

[6]Probably due to the estimation inaccuracies

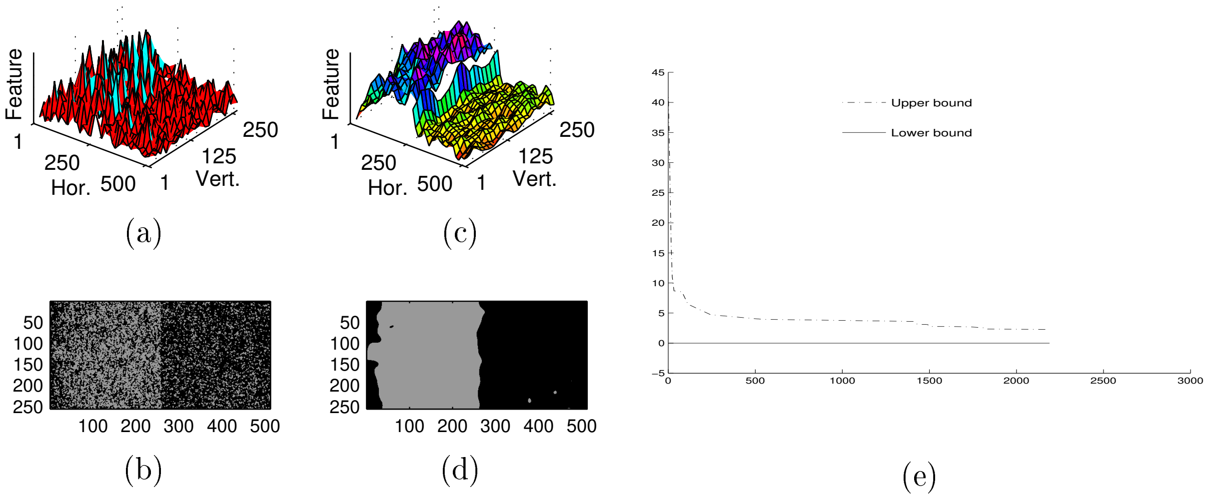
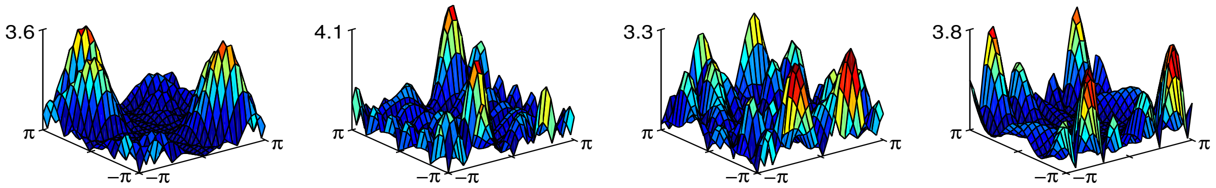


Figure 5.12: Features optimized with exact iterative optimization for J_F, Equation (5.25) for texture D9-D24. The optimal filter is of size $M_h \times N_h = 3\times3$, assuming a rectangular smoothing filter $w(m,n)$ of size only $M_w \times N_w = 3\times3$. (a) Feature surface and (b) class map using only the 3×3 rectangular smoothing filter. (c-d) The same using the same smoothing filter as in the other experiments, but still designing for the 3×3 smoothing filter. (e) The development of the bounds in the branch and bound algorithm as a function of number of iterations.

to the rapidly increasing complexity, the filter was restricted to the size $M_h \times N_h = 3 \times 3$, assuming a rectangular smoothing filter of size $M_w \times N_w = 3 \times 3$. This smoothing filter is clearly too small, as can be seen from Figures 5.12(a) and 5.12(b). To improve the results, the smoothing filter *applied* was increased to be of the same size as in the other experiments and the results are shown in Figures 5.12(c) and 5.12(d). The development of the bounds on the criterion value in the branch and bound iterations is shown in Figure 5.12(e).

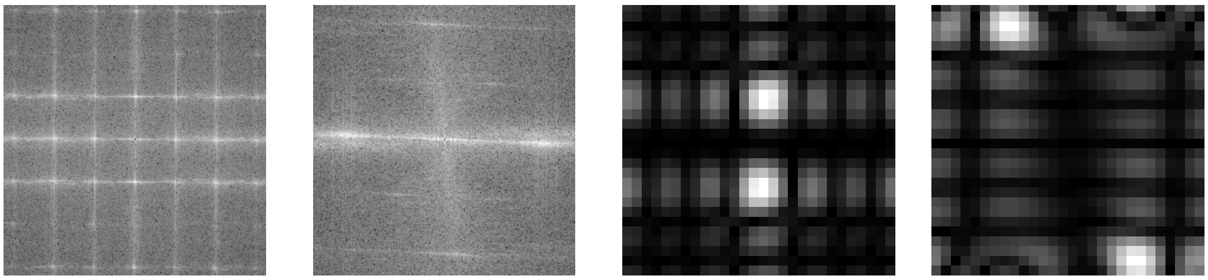
To illustrate what the optimal filters may look like, examples of amplitude responses are given in Figure 5.13. Two examples of amplitude spectra illustrated as gray-level images in conjunction with the amplitude spectra of the two corresponding textures are shown in Figure 5.14. This texture set, D21-D105, was chosen due to the clarity of the spectral peaks. If we consider the spatial frequency properties of an optimal filter, we would intuitively expect it to emphasize frequencies discriminating the textures well. That is, frequencies where one texture has strong components and the other weak. If we study Figure 5.14 carefully, we can see that this is the case for this example. For this texture set, the Fisher criterion did not yield very good features, as we intuitively understand from visual inspection – the Unser criterion is highlighting a more discriminative region of the frequency spectrum.

Results from the Gabor filter optimization approach by Dunn and Higgins [30] (Section 2.3.3) are given in Figure 5.15 for reference. The Gabor filters are sine modulated Gaussians and in the experiments filters with Gaussians with widths given by $\sigma = 8$ pixels are designed and applied. The nonlinearity and the smoothing filter were the same as in



(a) (b) (c) (d)

Figure 5.13: Amplitude response for the filter optimal with respect to J_F, Equation (5.25), closed form solution, for the texture pairs (a) D4-D84, (b) D21-D105, (c) D28-D29, and (d) D9-D24.



(a) (b) (c) (d)

Figure 5.14: Amplitude spectrum (log-scale) (a) for texture D21 and (b) for texture D105, and amplitude responses of the filters optimized with respect to (c) J_U, Equation (5.4) and (d) J_F, Equation (5.25).

the other experiments. As we can see from the figures, this approach was incapable of performing successful segmentation of the difficult images in Figures 4.1(c) and 4.1(d).

In an extensive experiment, the classification error was calculated for all approaches for 276 texture pairs. For computational reasons, the Gradient optimized filters were excluded from this analysis. Assume that errors below 5% correspond edge effects and are "acceptable". Then it was found that the Unser criterion did yield unacceptable features in 4% of the cases, Mahalanobis-Singh [66] in 29%, Fisher in 13%, the histogram-approach in 1%, and the Dunn [30] approach in 14%. The Mahalanobis-Singh [66] approach was clearly less robust than the other approaches. Some problems were also encountered with filters optimized with respect to the Fisher criterion and with the Dunn [30] approach. The most likely cause of the problems with the Fisher-optimal filters is the sensitivity of the variance estimate with respect to non-stationarities. The performances of the Unser- and Fisher-criterion optimized filters were generally complementary. That is, when one failed, the other succeeded in nearly 98% of the cases. The Unser criterion was yielding best features of these in about 60% of the cases.

The focus of this section and the experiments herein have been on segmentation of benchmark images. However, the introduced techniques have also been applied to other

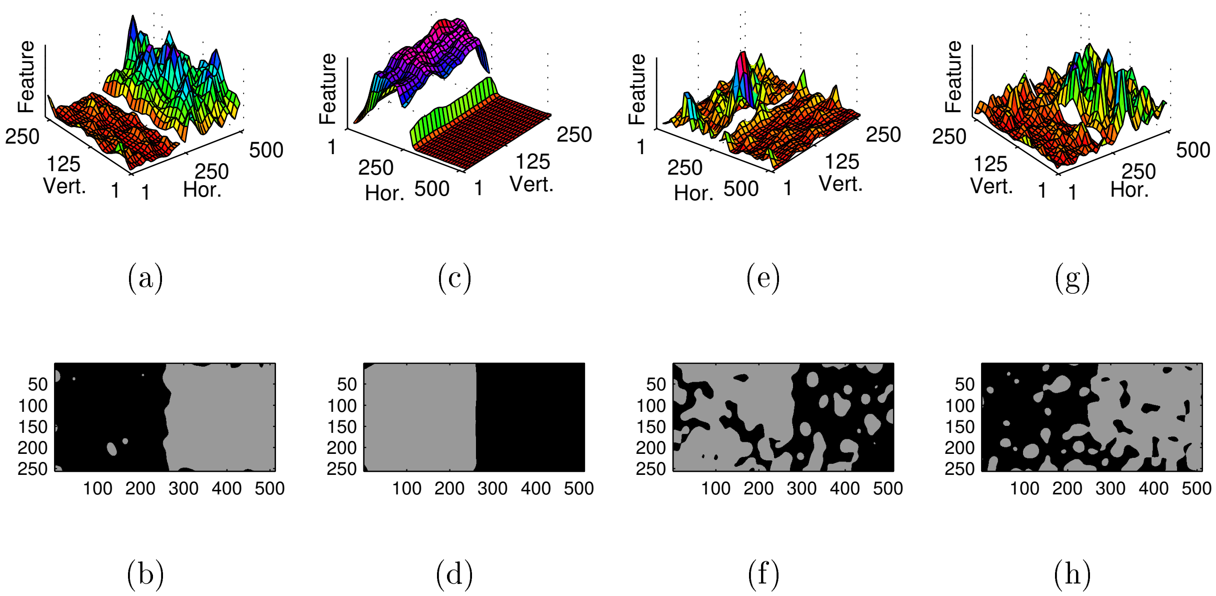


Figure 5.15: Features extracted with optimal Gabor filter [30] for the textures (a) D4-D84, (c) D21-D105, (e) D28-D29, and (g) D9-D24 and their corresponding class maps (b), (d), (f), and (h).

problems. A broad comparison of the techniques proposed here with techniques covering most approaches to filtering for texture separation is reported in Chapter 6. It turns out that these approaches are superior. Furthermore, Gulsrud et al. provide examples of these approaches successfully applied to detection of stellate lesions [41], circumscribed masses [41], and micro-calcifications [40] in mammograms.

5.4 Multi-texture optimal filter banks

In the previous section, we saw some optimal filter design approaches for two-texture discrimination. This may be interesting in some inspection application for discriminating normal from abnormal. However, many inspection problems and the majority of texture segmentation and classification problems involve more than two textures. Consequently, extensions of the solution to be applicable for multiple textures are desirable. To have reliable features for more than two textures, it is very likely that more than one filter is needed. The design and application of multiple filters for multiple texture problems is addressed in this section.

The straightforward approach for generalizing two-texture optimization approaches to solutions for more than two textures is to design one filter for each possible pairing of two textures in the problem. This approach was for instance applied to the filter design method of Dunn et al. [31] by Randen and Husøy [104]. This is a viable approach as long as the number of textures is moderate. However, as the number of textures increases, the number of filters gets very large, e.g. 10 textures give 45 filters, and 20 textures give 190 filters. Hence, we need another way of handling such problems.

Building on the two-texture approaches, a hierarchical feature extraction scheme will be proposed, where each filter separates one *group* of textures from another. The textures within the groups are separated by subsequent filters recursively. The problems that need to be solved are:

1. How do we design a filter for optimal feature separation between two *groups* of textures? This issue is addressed in Subsection 5.4.1.

2. How do we form texture groups allowing robust feature extractors to be constructed? This issue is addressed in Subsection 5.4.2.

Finally, the section is concluded in Subsection 5.4.3 with experimental results.

5.4.1 Optimal texture-group filters

We recall from Equation (5.12) that the mean value of the features extracted from texture $x_i(m,n)$ by the filter $h(m,n)$ can be written as $\mu_{v_i} = (\mathbf{h}^T \mathbf{R}_{x_i x_i} \mathbf{h})$. If the textures $x_1(m,n) \cdots x_I(m,n)$ are filtered by the same filter, the overall mean feature value for all textures will be

$$\begin{aligned} \mu_v &= \frac{1}{I}\left(\mu_{v_1} + \cdots + \mu_{v_I}\right) \\ &= \frac{1}{I}\left((\mathbf{h}^T \mathbf{R}_{x_1 x_1} \mathbf{h}) + \cdots + (\mathbf{h}^T \mathbf{R}_{x_I x_I} \mathbf{h})\right) \\ &= \mathbf{h}^T \frac{1}{I}\left(\mathbf{R}_{x_1 x_1} + \cdots + \mathbf{R}_{x_I x_I}\right) \mathbf{h} \\ &= \mathbf{h}^T \mathbf{R}_{x_a x_a} \mathbf{h}, \end{aligned} \tag{5.32}$$

where $\mathbf{R}_{x_a x_a}$ is the average $\mathbf{R}_{x_i x_i}$. This expression is identical to the expression for the mean of one texture if we replace $\mathbf{R}_{x_i x_i}$ with the average $\mathbf{R}_{x_i x_i}$'s, $\mathbf{R}_{x_a x_a}$, of all the textures in the group. Hence, by replacing $\mathbf{R}_{x_1 x_1}$ and $\mathbf{R}_{x_2 x_2}$ with the $\mathbf{R}_{x_a x_a}$'s for the two texture *groups* in the approaches of Section 5.3.1, the same filter design approaches may be used. Including the exact closed form solution, this relation can easily be shown. A similarly simple expression for the variance of a group of textures has not been found, even with the assumptions made in Appendix A.3. Consequently, a closed form solution for the Fisher criterion for multiple textures is not found. Still we may use iterative search, but due to complexity issues this is not very practical and has not been tested.

If a texture is non-stationary (non-homogeneous), it may in some cases be considered as being composed of multiple different textures. Then the autocorrelation function estimated over the entire texture image will be approximately equal to the average of the autocorrelation functions of its components, at least for small lags. Hence, $\mathbf{R}_{x_i x_i}$ for the entire texture is approximately equal to the mean $\mathbf{R}_{x_i x_i}$'s of the components. A consequence of Equation (5.32) is that the feature mean estimated using the autocorrelation function for the entire texture is approximately equal to the average feature means of its components. The same is not true for the variance.

5.4.2 Texture grouping

If the textures in one group share some characteristics that are less prevalent in another group, good filters may probably be designed. In order to make such a grouping, some measure of the similarity is needed. It can not be expected that very dissimilar textures may be grouped with success. Of course, filters corresponding to all possible group pairs of the textures may be designed and the expected feature statistics computed for each texture to determine which grouping is best. With many textures, such an approach becomes exhaustive. For instance, with 16 textures there are $2^{15} - 1 = 32767$ different group pairs. Hence, another way of determining the group pairs is desired.

In order to form two texture groups that may be successfully separated, we must ensure that the maximum of some distance measure between any two textures *within* a group is less than the minimum distance between any two textures *between* the groups. More formally, assume that the I textures $x_{i_1}(m,n) \cdots x_{i_I}(m,n)$ are one texture group and the J textures $x_{j_1}(m,n) \cdots x_{j_J}(m,n)$ are another. The conditions for any group pair to yield successful filters are then

$$\max\{d_{i_k,i_l}\} < \min\{d_{i_p,j_q}\} \;\; \forall\; k,l,p < I,\;\; q < J \tag{5.33}$$

and

$$\max\{d_{j_k,j_l}\} < \min\{d_{j_p,i_q}\} \;\; \forall\; k,l,p < J,\;\; q < I. \tag{5.34}$$

The measures d_{i_k,i_l} and d_{j_k,j_l} are within group distances and d_{i_p,j_q} and d_{j_p,i_q} are between. As similarity measure, the criterion value for the optimal filter for the two textures is used, e.g. if the filter is optimized with respect to the Unser criterion the similarity criterion is $d = -J_U$.

A disadvantage with the above conditions, is that the textures are fragmented into a lot of groups since typically only small groups satisfy conditions (5.33) and (5.34). To be able to separate two groups, one filter is needed. With many groups the number of group pairs will be large, hence a large number of filters must be applied. To reduce the number of filters, textures are therefore added to the groups as long as all the modeled feature means for one of the groups are above the means for the other. In this approach, no account is made for the variances. However, this may easily be added.

5.4.3 Experiments

A result from this approach with the Unser criterion for the textures D4, D9, D19, D21, D24, D28, D29, D36, D37, and D38 [14] is shown in Figure 5.16. There is just a moderate amount of misclassified regions, even though some of the textures are visually very similar. The texture grouping yielded 17 texture group pairs where four were duplicates, i.e., 13 filters were designed. The filtering hierarchy is illustrated in Figure 5.16(c). Note that due to the hierarchical nature of the solution, it is not necessary to compute filter responses for all image positions. It is, for instance, not necessary to apply the filter separating between classes 0 and 5 to regions of the image that are already recognized as belonging to one of the classes 1 or 6. As a consequence, the effective number of filters is less than 13

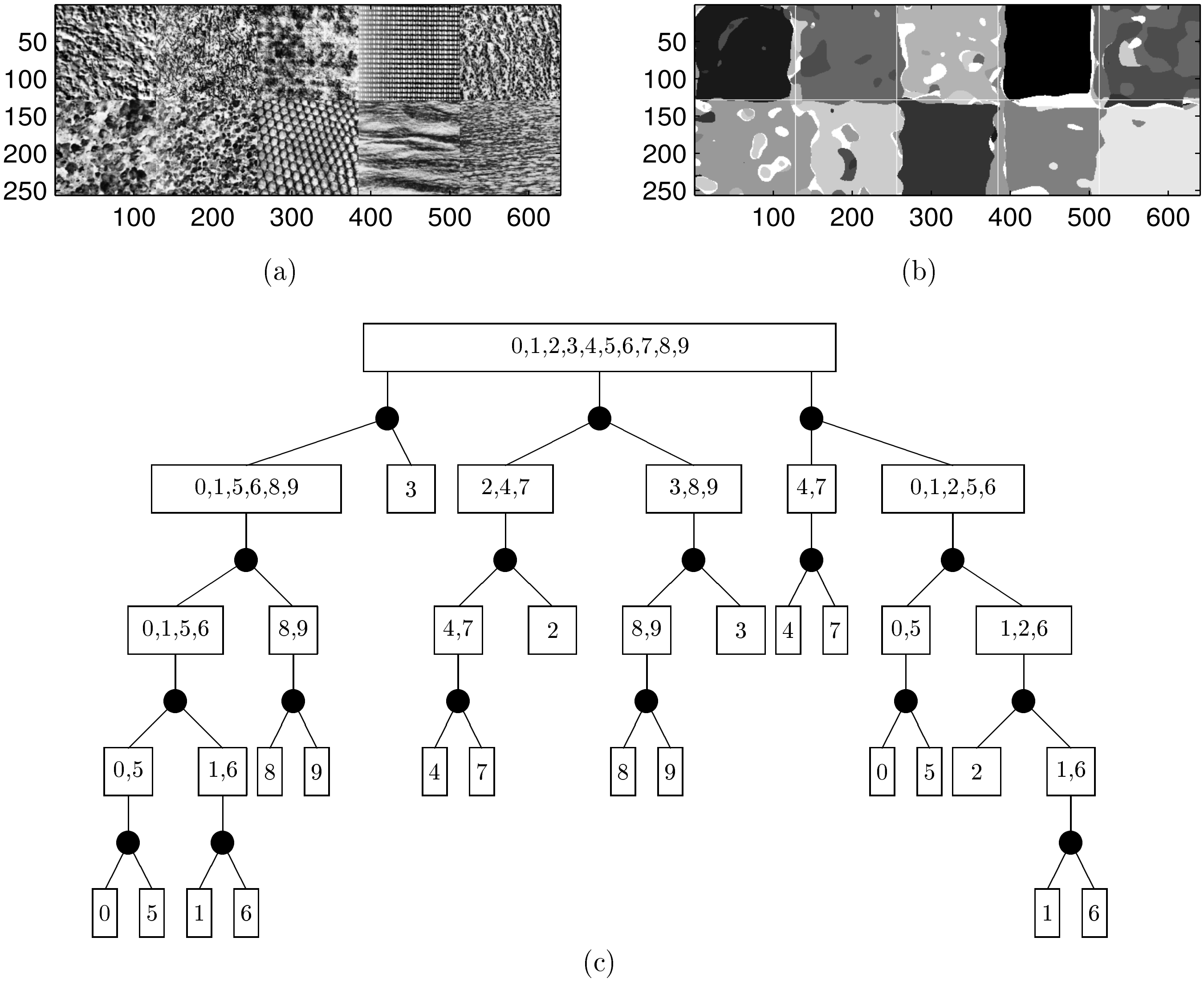


Figure 5.16: Multi-texture filter optimization experiment. (a) Ten-texture image, (b) segmentation by the multi-texture approach, and (c) texture grouping hierarchy where each black disc corresponds to one filter. The numbers in (c) refer to texture number, where the textures in (a) are numbered 0 to 4 from left to right in the first row and 5 to 9 in the second.

in this case, which is very low compared to many filter bank approaches, see Sections 2.2 and 3.2. Further results are reported in Chapter 6. From these results, we see that this approach yields very good segmentation compared to most other filtering approaches.

In an artificial mosaic of textures as the one in Figure 5.16(a), the edges between textures are very discontinuous, i.e., having large high-frequency contents. The feature extraction filter may have high-pass characteristics. Hence, it may have a boosted response at the edge instead of a smooth transition between the texture regions. In combination with the local energy function, this implies reduced edge accuracy. This effect is evident e.g. around the second texture from the right in the top row of Figure 5.16(b). Such a boosted response will probably not occur to a great extent in a real world mosaic, since texture edges in real world images are likely to be much smoother. However, if strongly discontinuous edges are present, this problem can be alleviated by edge modeling and compensation as in described in Section 5.2.5 and illustrated in Section 5.5. Other alternatives are neighborhood examination [139] and application of smaller filters or smaller smoothing windows in the regions near the edges.

5.5 Unsupervised optimal filters

Most optimal filter design approaches are restricted to supervised problems, but the work by Teuner et al. [127] shows that optimal filter design may be useful even for unsupervised problems. In their approach, the image is partitioned into cells and decomposed by a pyramidal Gabor transform with dyadic decreasing cell sizes. The cells are examined to find the most significant spectral components. Cells that are similar with respect to spectral content are grouped. The corresponding frequencies form the basis for designing the Gabor filters. However, as we have seen, the Gabor filter may in some cases be too restricted to yield appropriate features. In this section an unsupervised texture feature extraction and segmentation technique using the schemes of Section 5.3 or 5.4 is proposed. These filters are not designed using a single significant frequency per filter, but rather using the autocorrelation functions of the textures. Consequently, the pyramidal Gabor transform analysis of Teuner et al. [127] is not appropriate to get the texture properties for designing these optimal filters. A block-based clustering scheme where cells of the image are clustered by a cell similarity measure related to the filter design parameters is proposed in Sections 5.5.1 and 5.5.2. Once the cells are clustered, the autocorrelation functions are estimated for the major clusters and the corresponding optimal filters are designed. Experiments with the approach are reported in Section 5.5.3.

This section is intended to be an example of a possible extension of the optimization approaches in this chapter to unsupervised problems. Several modifications and improvements are possible. Only two-texture problems with the Fisher criterion function are considered in the experiments, but application of other criterion functions and extension to multiple texture cases are readily possible.

5.5.1 Cell similarity function

The first step in the unsupervised feature extraction and segmentation scheme proposed herein is block-based clustering of the input image. The image is divided into equally sized cells. Cells that are similar according to some similarity measure are assumed to belong to the same cluster. It is desirable to compare cells globally to determine if they are similar and should be merged. However, due to the nature of the cell comparison technique developed in this section, the similarity is calculated pairwise between cells. Since the number of cell-pairs increases rapidly with the number of cells, the similarity is only computed for neighboring cells. When the image is divided into a presumably small number of clusters of cells, all clusters are compared on a global basis and similar clusters are merged. Finally, the auto correlation functions are estimated from the clusters and used for filter optimization as in Section 5.3.2.

As we saw in Section 5.3.2, the optimal filter for a texture pair is found from one of the eigenvectors of the matrix $\left(\mathbf{R}_{x_2x_2}{}^{-1}\mathbf{R}_{x_1x_1}\right)$. If $\mathbf{R}_{x_1x_1} = \mathbf{R}_{x_2x_2}$, all the eigenvalues of $\left(\mathbf{R}_{x_2x_2}{}^{-1}\mathbf{R}_{x_1x_1}\right)$ are equal to one, so if $\mathbf{R}_{x_1x_1}$ and $\mathbf{R}_{x_2x_2}$ are estimated from the same texture, all eigenvalues of $\left(\mathbf{R}_{x_2x_2}{}^{-1}\mathbf{R}_{x_1x_1}\right)$ are expected to be very similar. On the other hand, if $\mathbf{R}_{x_1x_1}$ and $\mathbf{R}_{x_2x_2}$ are estimated from two different textures, the eigenvalues of $\left(\mathbf{R}_{x_2x_2}{}^{-1}\mathbf{R}_{x_1x_1}\right)$ will be more spread. Hence, if we estimate $\mathbf{R}_{x_1x_1}$ and $\mathbf{R}_{x_2x_2}$ from two cells in an image, a large spread will indicate that two cells probably contain two different textures and a small spread will indicate similarity.

A suitable function measuring the spread of the eigenvalues is the ratio of the geometric to the arithmetic mean of the eigenvalues [5]

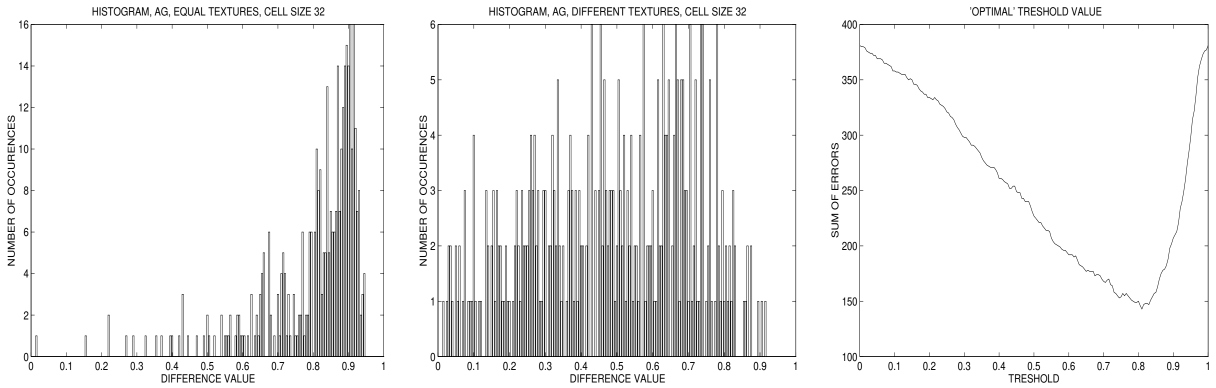
$$d_{i,j} = \frac{\frac{1}{N}\sum_{n=1}^{N}\lambda_n}{\left(\prod_{n=1}^{N}\lambda_n\right)^{\frac{1}{N}}}. \tag{5.35}$$

This function is equal to unity if all eigenvalues are equal (indicating cells containing equal textures) and approaches zero as the eigenvalue spread increases (indicating different texture cells). The threshold value ξ is introduced to decide whether or not two cells contain similar textures. If $d_{i,j} < \xi$, the two cells are assumed to contain different textures and if $d_{i,j} > \xi$, they are assumed to contain the same.

Distance histograms were computed using 20 different textures from Brodatz [14]. Cells from the same texture were obtained by extracting cells from different sub-images of the texture images. Cell sizes of 32×32 were used in the experiments. The histograms for cells from the same texture and from different textures are shown in Figures 5.17(a) and 5.17(b), respectively. The error of assuming similar when we have different textures and assuming different when we have similar textures is plotted in Figure 5.17(c) as a function of the threshold.

For cell sizes 128×128, 64×64, and 32×32 pixels, the optimal thresholds, ξ, are 0.935, 0.845, and 0.825, corresponding to error probabilities of 4.3%, 13.7%, and 17.6%.[7] Hence, the similarity functions have different optimal threshold values for different cell

[7] 128×128 and 64×64 cells are clearly too large for the images used in this dissertation. Nevertheless, the figures illustrate the point well – decreasing cell sizes yield increased decision errors.



(a) (b) (c)

Figure 5.17: Histograms of the cell similarity measure with cell size 32×32 pixels for (a) cells containing similar textures and (b) cells containing different textures, and (c) classification error vs. cell similarity.

sizes. The error increases as the cell size decreases, due to an increased variance in the parameter estimates.

5.5.2 Labeling connected cells

A cell has four horizontal and vertical neighbors and four diagonal neighbors. Hence, a complete comparison of all the neighbors involves computing eight similarity values. The image is scanned cell by cell from left to right and from top to bottom. If all the labeled neighbors of a cell have similarity values below ξ, the cell is not similar to its surrounding cells and is therefore assigned a new label. If two or more labeled neighbors are similar (i.e., $d_{i,j}$ greater than ξ), one of the labels is assigned to the current cell and a note of the equivalences is made. If only one neighbor is similar, its label is assigned to the current cell.

After a complete scan of the image, the similar label pairs are represented in similarity classes and unique labels are assigned to each cluster. However, one problem still remains. If a texture has been split into several parts, the method identifies the texture as different textures. To overcome this problem, all regions are compared to each other using the average autocorrelation functions. Region pairs having a similarity measure above the threshold are merged.

If a cluster contains just one or two cells after the cluster merging, it is assumed that these cells cover transition zones between neighboring texture regions. Hence, the cells are omitted in the subsequent filter design. The number of classes is consequently equal to the number of clusters composed by more than two cells.

5.5.3 Experiments

In all experiments, an empirically selected $M_w \times N_w = 20 \times 20$ averaging (i.e., rectangular) filter is used as smoothing operator. This smoothing operator has approximately the same cut-off frequency as the Gaussian smoothing filter used in the previous experiments. The experiments will illustrate both the unsupervised segmentation and the edge modeling and compensation of Section 5.2.5. One of the reasons for selecting a different smoothing filter here than in the previous experiments was to illustrate the edge effects more clearly.[8] Furthermore cell sizes of 32×32 pixels are used in the cell clustering steps.

As described in Sections 5.5.1 and 5.5.2, the cell clustering is a two-step process. In the first step, each cell is only compared to its neighbors. However, the input image may contain the same texture in several disconnected regions of the image. Furthermore, since the autocorrelation estimates for small cells have high variances, cells belonging to the same class may be identified as belonging to two different clusters. Consequently, some homogeneous texture regions may be split into several clusters.

In the experiments here, images constructed from textures textures D55 and D56, Figure 5.18(a), and by D17 and D6, Figure 5.18(b), from Brodatz [14] are used. The results of applying the first cell clustering step are shown overlayed. In all figures in this section, the left figure will be the results corresponding to Figure 5.18(a) and the right figure will be the results corresponding to Figure 5.18(b).

To overcome the problem of fragmented texture regions illustrated in Figure 5.18(b), a merging procedure was suggested in Section 5.5.2. In this step, all clusters are compared to each other and clusters found to be similar are merged. The results after this merging are shown in Figures 5.18(c-d). Note especially that the background texture, which was split into two large clusters in Figure 5.18(b), has been merged to one cluster in Figure 5.18(d). The merged clusters are used as basis for determining the autocorrelation functions of the textures in the images. These autocorrelation functions are then used to determine the filters optimized with the closed form approximate Fisher criterion solution, described in Section 5.3.2. The extracted feature images are shown in Figures 5.18(e-f).

Using a minimum error threshold classifier,[9] the class maps are as shown in Figures 5.19(a-b). As we can clearly see from Figures 5.18(e-f) and 5.19(a-b), the edges are severely biased due to the feature extraction responses at the edges. Modeling the edge responses as described in Section 5.2.5, we get the scan lines of the expected feature values for horizontal and vertical texture edges illustrated in Figures 5.19(c-d). Through the modeling it is found that the center texture region is expected to be 11 pixels too small at each vertical border and 8 at each horizontal border for the image of Figure 5.18(a). That is, the curves of Figure 5.19(c) cross the threshold 11 and 8 pixels off the edges. Correspondingly, the center texture of Figure 5.18(b) is too large by 3 pixels at vertical borders and 5 pixels at horizontal borders. Incorporating these numbers into a morphological dilation kernel as described in Section 5.2.5, we get the class maps of Figures 5.19(f-g). We can clearly see that we now get texture boundaries that are very close to the true texture boundaries.

The approach by Teuner et al. [127] has also been applied to the dataset used in this chapter [5] and the results were as illustrated in Figure 5.20. We clearly see that the proposed approach is better for these two images. Further experiments, supporting this finding, are given by Alvestad [5]. Notice that in the approach by Teuner et al., more

[8] The rectangular filter has worse edge properties than the Gaussian low-pass filter.

[9] The error is minimized with respect to supervised classification of the cell clusters.

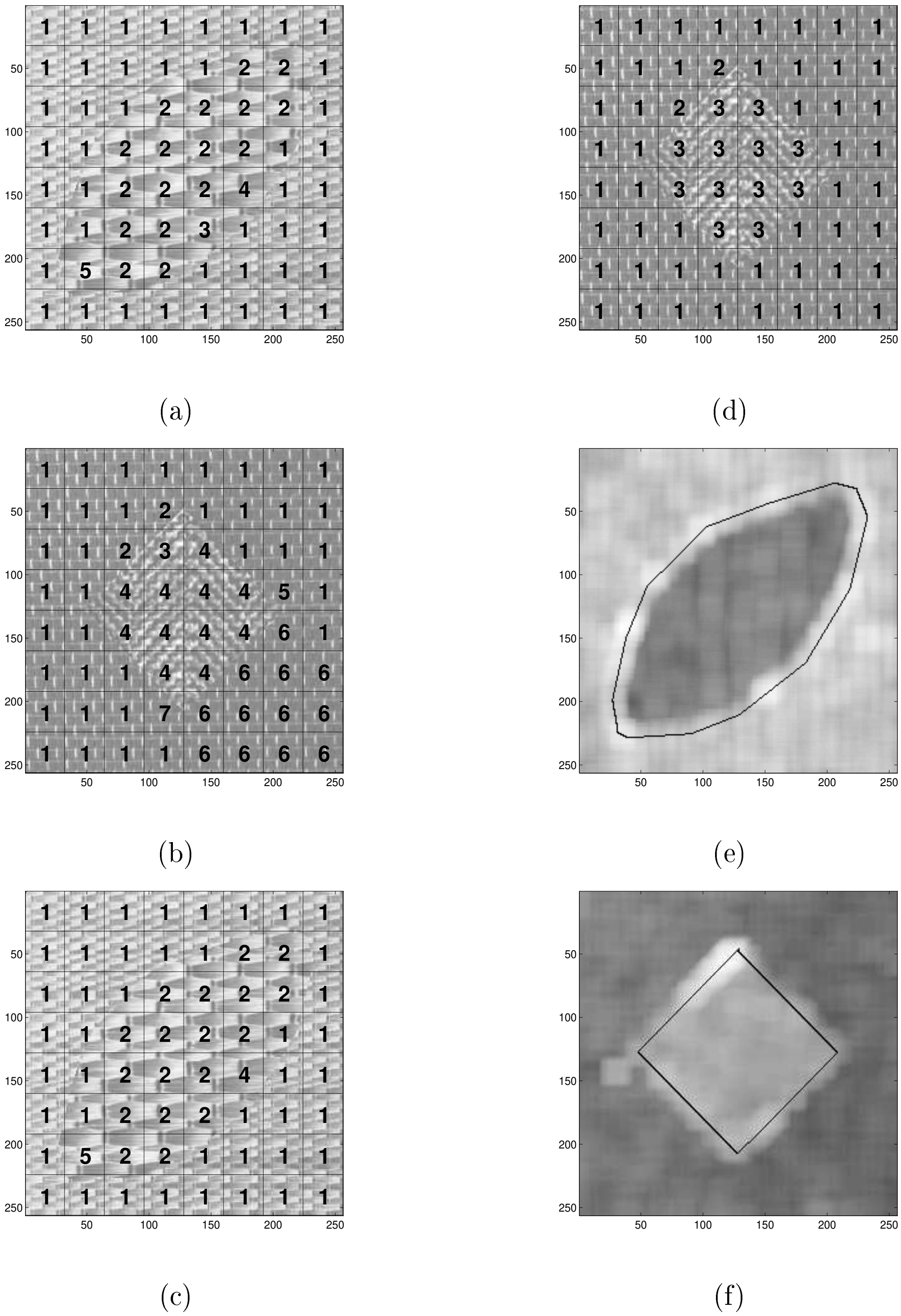


Figure 5.18: Unsupervised experiments. Texture images used in the experiments; (a) D55 and D56 [14] and (b) D6 and D176 [14]. The clustering results for the cell clustering are shown overlayed. (c-d) The clustering after merging similar cell clusters. (e-f) The feature images corresponding to Fisher-optimal filtering, discriminating the two major clusters of (c-d).

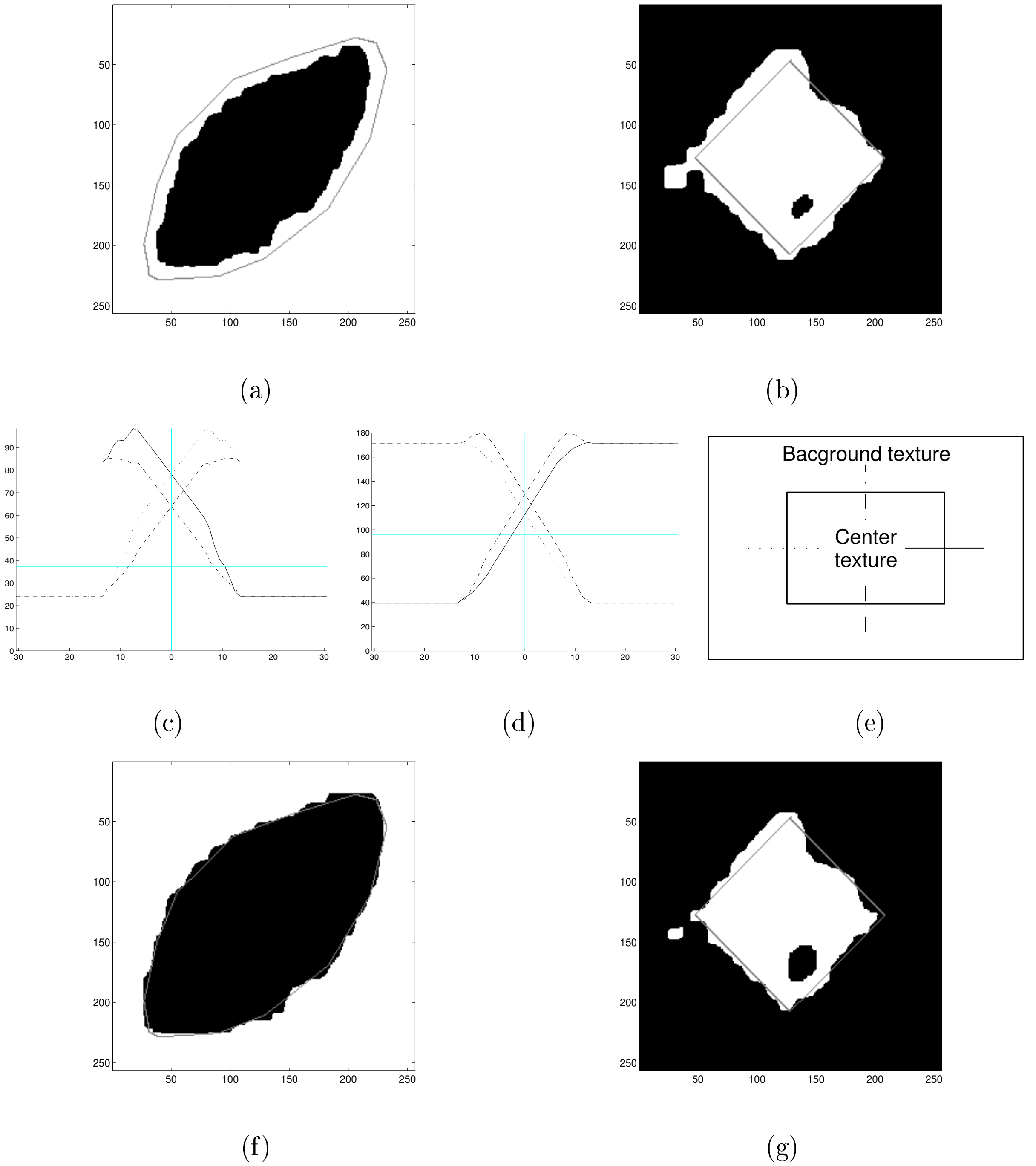
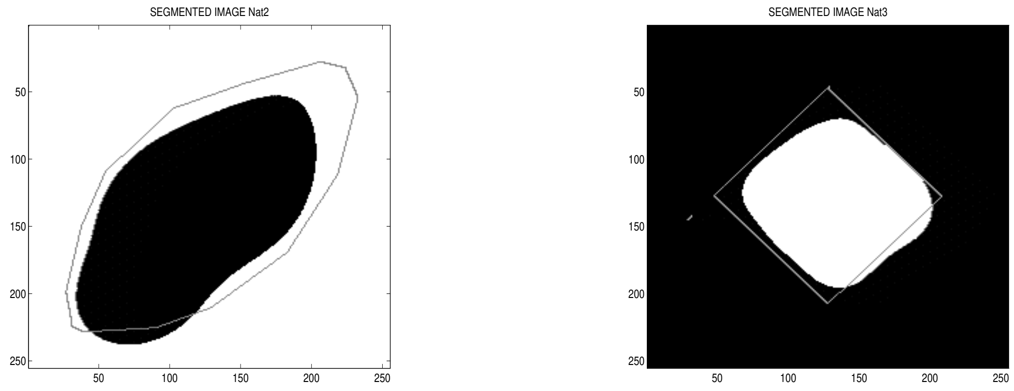


Figure 5.19: (a-b) Class map images resulting from minimum error thresholding of the feature images of Figures 5.18(e-f). The lines in the images indicate the true texture boundaries. (c-d) Edge modeling of the features of Figures 5.18(e-f), representing expected feature values across a texture edge. The horizontal axes represent pixel numbers relative to the edge. The vertical lines indicate the true texture boundaries and the horizontal lines indicate the selected feature threshold values. (e) Descriptions of the line styles in (c-d): Center texture right of background texture (—), center left of background (···), center below background (–·), and center above background (– –). (f-g) Region map images after morphological dilation.



(a) (b)

Figure 5.20: Unsupervised segmentation results for the images in Figures 5.18(a-b) using optimal Gabor filters as proposed by Teuner et al.

than one filter was used, whereas in the proposed approach, only one filter was used.

As the experiments indicate, the proposed cell clustering technique works well for these texture images. However, due to the fact that the the cell-clustering is based on region growing, it is possible that very different cells may be combined into one cluster through a series of neighbors. A direction for future research is therefore to find a more robust way of clustering cells.

5.6 Summary

In this chapter, estimators for the mean and variance of texture features extracted with a texture filter, a quadratic nonlinearity, and a smoothing filter have been proposed. The mathematical models for these estimators formed the basis for the following filter optimization procedures:

1. Optimal exact closed form two-texture filter design with respect to the Unser criterion.

2. Optimal approximate closed form two-texture filter design with respect to the Fisher criterion.

3. Optimal approximate closed form two-texture filter design with respect to a generalized criterion. Motivated by this development, a filter selection scheme selecting the filter yielding minimum histogram overlap was developed.

4. Extension to optimal multiple texture filter bank design.

5. Extensions to unsupervised problems.

In order to find closed form solutions to the optimization problems incorporating the variance, an approximate variance model had to be used. The two-texture design techniques were thoroughly tested on several texture images. The performances of the design techniques based on the Unser and Fisher criteria were very good in most cases. Best performance was, however, achieved with the histogram-based technique resulting from the unified approach. It is important to keep in mind that none of the criteria are guaranteed to have a direct relation to the classification error, the ultimate feature goodness criterion.

Issues that are not addressed in detail in this dissertation are how to determine the ideal filter size and the ideal smoothing filter size and type. These parameters have impact on the feature extraction results, and should be addressed in future work.

Generally, compared to using fixed filter banks as in Sections 2.2 and 3.2, the main disadvantages with an optimal approach are the need for better insight and a more complex design. The main advantages are the optimality, fewer filters and hence fewer features, and much simpler classification (thresholding with one feature is far less complex than applying a neural net or statistic classifier with multiple features). Furthermore, compared to the commonly used Gabor filters [50, 31, 30], successful results are obtained with significantly smaller, i.e., less computationally complex filters.

Chapter 6

Texture segmentation: Comparative experiments

Although similar in concept, several quite different filtering schemes have been presented, see Sections 2.2 and 2.3 and Chapters 3, 4, and 5. One of the pioneering approaches was the approach by Laws [62], Section 2.2.1, where a bank of band-pass filters was applied. Subsequent works have focused on different filter bank bases, different sub-band decompositions, and on optimization of the filters for texture feature separation. The large number of different test images applied, along with different system setups, have made comparison of the filtering approaches based on published results very difficult.

A few comparisons between texture feature extraction schemes have been presented. Weszka et al. [140] compared the Fourier power spectrum, second order gray-level statistics, co-occurrence statistics, and gray-level run length statistics features. They concluded that the co-occurrence features were the best of these features. The co-occurrence features were also the best features in a study by Conners and Harlow [24], when compared with run length difference, gray-level difference density, and power spectrum.

du Buf et al. [28], on the other hand, reported that several features, were performing approximately similarly. They evaluated co-occurrence features, fractal dimension, transform and filter bank features, dominant local frequency and orientation features, number of gray-level extrema per unit area, and curvilinear integration features. Filtering features have been compared to the co-occurrence features in some studies, with different conclusions. Strand and Taxt [123] concluded that the co-occurrence features were performing best, while Laws [62], Pietikäinen et al. [92], and Clausi and Jernigan [19] had the opposite conclusion. Different setups, different test images, and different filtering methods may be the reasons for the contradicting results. Ojala et al. [81] compared gray-level difference, Laws filter, covariance, local binary patterns, and complementary feature pairs. They concluded that the gray-level difference features were performing best and noted a particularly poor performance of the Laws features.

It is noted that no extensive evaluation of filtering approaches has been performed. The aim of this chapter is to provide such a comparative study. The system setup will be as illustrated in Figure 2.1. In order to make the results comparable, the focus will be on the filtering part of this system, keeping the other system parameters to a large extent the same for all methods. However, some variation is allowed due to inherent properties

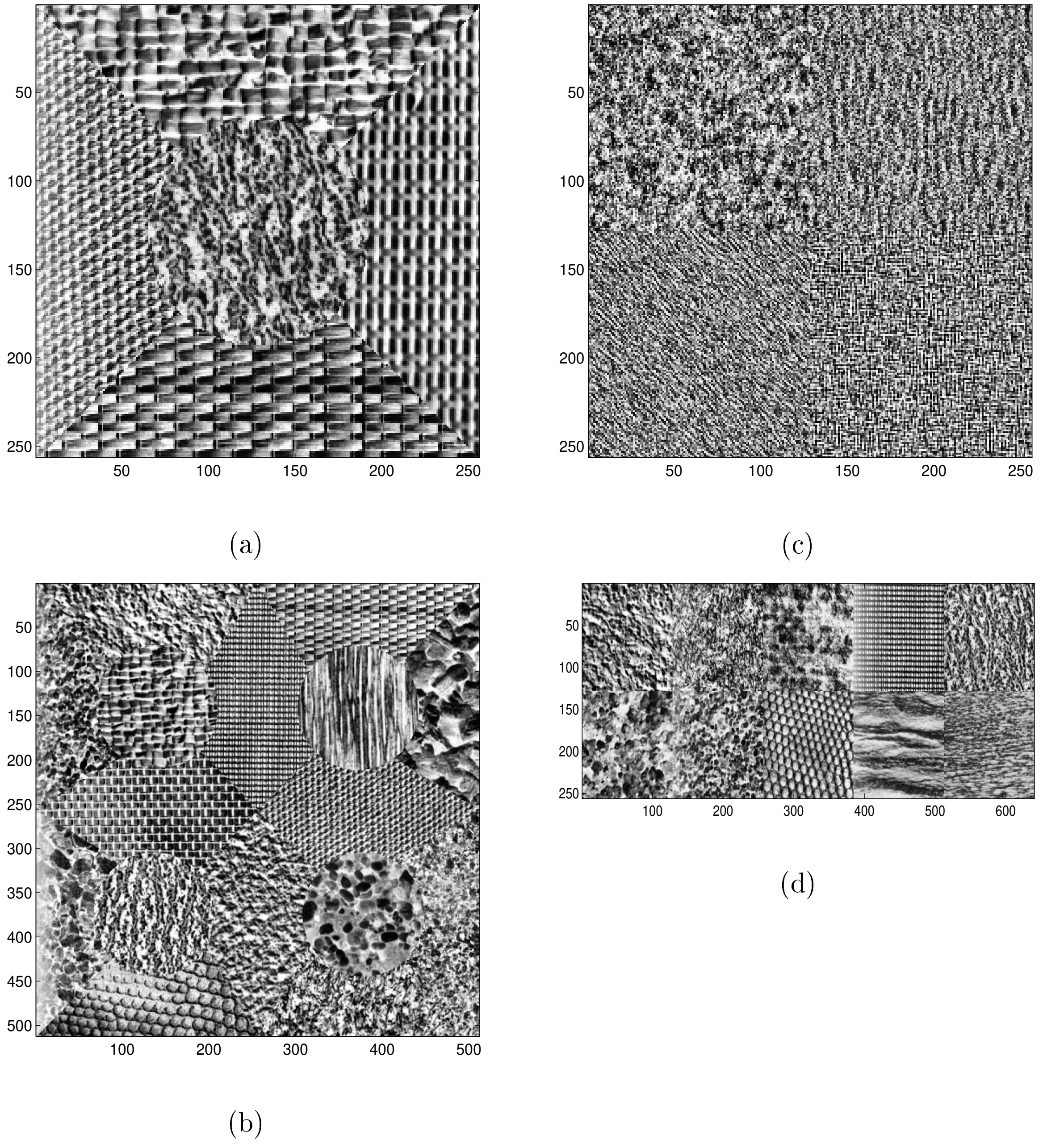
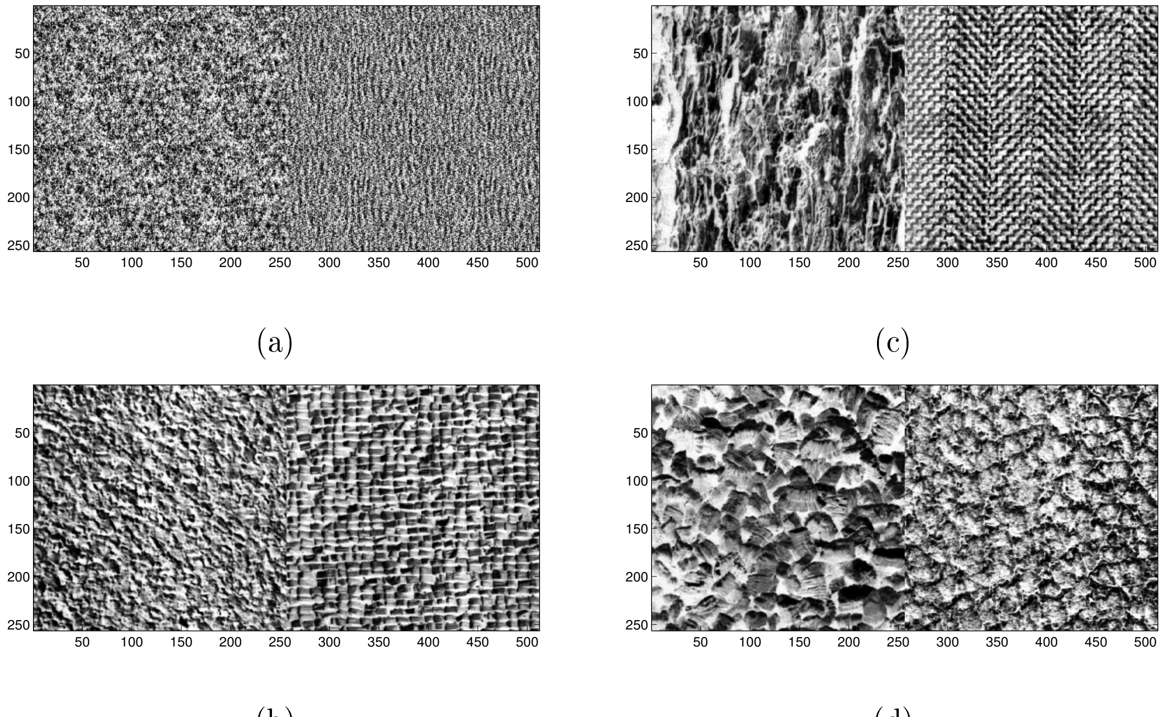


Figure 6.1: Composite texture images used in the experiments. (a) 256×256 pixels 5-texture image consisting of D77, D84, D55, D53, and D24 [14] (the same as shown in Figure 1.1(a)). (b) 512×512 pixels 16-texture image consisting of D3, D4, D5, D6, D9, D21, D24, D29, D32, D33, D54, D55, D57, D68, D77, and D84 [14]. (c) 256×256 pixels 4-texture image consisting of synthetic Gauss Markov random samples [50]. (d) 256×640 pixels 10-texture image consisting of D4, D9, D19, D21, D24, D28, D29, D36, D37, and D38 [14].



(a) (c)

(b) (d)

Figure 6.2: 256×512 pixels two-texture images used in the experiments, (a) the upper two textures of Figure 6.1(c) (replicated to be 256×256 pixels each) (b) D4 and D84 [14], (c) D12 and D17 [14], and (d) D5 and D92 [14].

of the approaches. For reference, the filtering approaches will also the compared with two classes of non-filtering features, model based and statistical features. In the experiments, supervised segmentation results for images with 2 to 16 textures with borders ranging from simple to difficult are compared. Special emphasis is put on making the results realistic, thus the design and test data sets are disjoint. Interestingly enough, this is far from always the case in experiments on supervised texture segmentation, see Section 2.1.2. This also makes the experiments different from experiments in earlier chapters.

This chapter is organized as follows. In Section 6.1, the experimental setup is described, in Section 6.2, the experimental results are presented and discussed, and in Section 6.3 the results are summarized.

6.1 Experimental setup

In the experiments, the different feature extraction approaches are evaluated by performing supervised segmentation on several test images of varying complexity, Figures 6.1 and 6.2. As feature quality criterion, the classification error, "the most common measure of performance for a recognition system" [80], is used. All filter optimization and classifier training are performed on 256×256 subimages of the texture images that are not part of the test images. For the classifier design, features were extracted from these 256×256 images individually and used for training. However, due to the availability of image data,

Identifier	Smoothing filter	Classifier
Laws filters	$\sigma_s = \frac{1}{2\sqrt{2}f_0}$	LVQ
Ring/wedge filters	$\sigma_s = \frac{1}{2\sqrt{2}f_0}$ and $\sigma_s = 16$	LVQ
Dyadic Gabor filter bank	$\sigma_s = \frac{1}{2\sqrt{2}f_0}$	LVQ
Arbitrary Gabor filter bank	$\sigma_s = \frac{1}{2\sqrt{2}f_0}$	LVQ
DCT	$\sigma_s = \frac{1}{2\sqrt{2}f_0}$	LVQ
Co-occurrence	N/A	LVQ
AR	$\sigma_s = 3$	LVQ
Daubechies n	$\sigma_s = \frac{1}{2\sqrt{2}f_0}$	LVQ
f16b	$\sigma_s = \frac{1}{2\sqrt{2}f_0}$	LVQ
F_2_1_smpl	$\sigma_s = \frac{1}{2\sqrt{2}f_0}$	LVQ
F_2_1_09	$\sigma_s = \frac{1}{2\sqrt{2}f_0}$	LVQ
Eigenfilter	$\sigma_s = 8$	LVQ
Opt. repr. Gabor filter bank	$\sigma_s = \frac{1}{2\sqrt{2}f_0}$	LVQ
Prediction error filter	$\sigma_s = 8$	Minimum value selection
Optimal Gabor filter $\sigma =?$	$\sigma_s = 2\sigma$	Threshold
n-filter optimal Gabor filter bank	$\sigma_s = 2\sigma$	LVQ
J_{MS}	$\sigma_s = 8$	Threshold
J_U	$\sigma_s = 8$	Threshold
J_F	$\sigma_s = 8$	Threshold
BackProp. NN, mask size n	N/A	Feed-forward neural net

Table 6.1: Details on the smoothing filters and classifiers used in the experiments. For the smoothing filters, the σ_s-parameter of a Gaussian lowpass-filter is given. Any f_0 is the sub-band center frequency and σ is the parameter of a Gabor filter in the filter bank.

it was not possible to optimize and train on a separate dataset for the images in Figures 6.1(c) and 6.2(a). This makes the results with these images less reliable, but since interesting results are seen with these images, they are still kept.

The local energy function used is the one described in Section 2.1.1, i.e., nonlinearity and smoothing with a Gaussian smoothing filter. For sub-sampled filters, the modification of Section 3.2.2 is used. For filters not having a narrow pass-band, Gaussian low-pass filters with $\sigma_s = 8$ (empirically determined) are used. The nonlinearity used is the squaring with logarithmic normalization after filtering, since this combination was found to be optimal [132]. As described in Section 2.1.2, the classifier is LVQ [58] in most cases. However, with the optimal Gabor filters [30], Section 2.3.3 and the optimal FIR filters, Sections 5.3 and 5.4, thresholding is used. With the optimal prediction error filters, Chapter 4, the image is classified by assigning the class label associated with the minimum error predictor. The classifiers and energy functions applied are summarized in Table 6.1. The method identifiers used in this and subsequent tables are defined in Table 6.2.

A normalization by a global normalization factor is applied for each feature. The global normalization factors are determined as the factors yielding unity variance for the image in Figure 6.1(a).

Identifier	Max number of classes	Description	References
Laws filters	∞	Section 2.2.1	[62]
Ring/wedge filters	∞	Section 2.2.2	[21]
Dyadic Gabor filter bank	∞	Section 2.2.3	[50]
Arbitrary Gabor filter bank	∞	Section 3.1	
DCT	∞	Section 2.2.5	[79]
QMF	∞	Section 3.2	
Co-occurrence	∞	Section 2.4.1	[42]
AR	∞	Section 2.4.2	[128]
Daubechies n	∞	Daubechies n-tap wavelet, Section 2.2.4	[25, 16, 60] [115, 131]
f16b	∞	f16b QMF-filter, Section 3.2	
F_2_1_smpl	∞	F_2_1_smpl QMF-filter, Section 3.2	
F_2_1_09	∞	F_2_1_09 QMF-filter, Section 3.2	
Eigenfilter	∞	Section 2.3.1	[1]
Opt. repr. Gabor filter bank	∞	Section 2.3.2	[12, 11]
Prediction error filter	∞	Section 4.1	
Optimal Gabor filter $\sigma =?$	2	Section 2.3.3	[30]
n-filter optimal Gabor filter bank	∞	Section 2.3.4	[138, 139]
J_{MS}	$2/\infty$	Sections 2.3.5, 5.3, and 5.4	[66]
J_U	$2/\infty$	Sections 5.3 and 5.4	
J_F	2	Section 5.3	
BackProp. NN, mask size n	∞	Section 2.3.7	[51]

Table 6.2: Summary of identifiers used to identify the methods in the result tables.

6.2 Results

The presentation of the result is divided in three parts. First, in Subsection 6.2.1, heuristicall designed, i.e., fixed, filter banks are considered. Next, in Subsection 6.2.2, critically sampled filter banks (including wavelet transform/packets) are discussed. In Subsection 6.2.3, results with the optimized filters and filter banks are discussed. Finally, in Subsection 6.2.4, a discussion of the test images is given.

6.2.1 Heuristically designed filter banks

The results for all the heuristically designed feature extractors are presented in Table 6.3. The best results from the wavelet frame, arbitrarily decomposed Gabor filter, and QMF filter experiments are shown in the rows labeled "Wavelet", "Arbitrary Gabor" and "QMF", respectively. The results from these filter classes are detailed in Table 6.4.

In order to get an increased insight from the experiments, there are multiple "dimensions" in the results. Some approaches allow variations of the filter basis and sub-band decomposition, some are fixed on basis, some on decomposition, and some on both. The

	Test image (figure number)							
Method	6.1(a)	6.1(b)	6.1(c)	6.1(d)	6.2(a)	6.2(b)	6.2(c)	6.2(d)
Laws filters	18.6%	48.3%	6.3%	41.9%	2.9%	24.8%	3.3%	8.6%
Ring/wedge filters	14.6%	43.8%	6.1%	44.5%	5.5%	3.0%	4.5%	18.4%
Dyadic Gabor filter bank	10.7%	60.1%	10.6%	32.3%	0.9%	3.6%	2.0%	11.3%
Arbitrary Gabor (best, Tab. 6.4)	8.2%	54.8%	14.3%	39.7%	1.0%	6.5%	1.2%	15.6%
DCT	13.2%	40.9%	7.0%	38.2%	6.9%	6.4%	2.2%	2.5%
Wavelet fr. (best, Tab. 6.4)	8.7%	38.2%	5.4%	40.9%	0.7%	4.4%	0.4%	4.7%
QMF (best, Tabs. 6.4)	8.7%	36.4%	3.5%	41.7%	0.7%	6.7%	0.8%	6.0%
Co-occurrence	9.9%	49.6%	5.3%	35.3%	2.0%	1.9%	4.8%	3.3%
AR	19.6%	58.0%	13.9%	56.7%	3.5%	11.3%	1.9%	3.0%

Table 6.3: Classification errors for different heuristically designed texture feature extractors. The numbers for the wavelet frame, arbitrarily decomposed Gabor, and QMF methods represent the best numbers for different setups.

number of feature images also vary much, from eight to 40 features, having considerable impact on the complexity. The complexity, sub-band decomposition, and filter basis issues will be discussed later, but first an overall discussion of the results is given.

Overall discussion

No clear hierarchy of classification performances is observed in Table 6.3, thus no approaches are clear winners. For some images, some approaches are good, for different images, others are good. No methods are consistently poor, but many methods have examples with poor relative performance.

Considering Table 6.3, evidently the "old" Laws and ring/wedge filters not very good. They never stand out as clear winners, and even fail completely in some cases: The Laws approach fails for the images in Figures 6.1(a), 6.1(b), 6.2(b), and 6.2(d). The ring/wedge filters fail with Figures 6.1(a), 6.1(d), 6.2(a), 6.2(c), and 6.2(d). Also the commonly referenced dyadic Gabor filter bank show rather poor performance for some of the test images, Figures 6.1(c), 6.1(d), and 6.2(d). Similar comments apply to the arbitrary Gabor filter bank. The DCT approach did also show poor relative performance for a few images, Figures 6.1(a), 6.2(a), and 6.2(b). However, no "breakdown" of the DCT features was observed.

With the QMF and wavelet approaches, multiple experiments were performed for each entry in Table 6.3. These results are summarized in Table 6.4. The QMF and wavelet frame approaches are among the best for most images. Examples of the classification results for the best QMF filter bank are shown in Figure 6.3. These results correspond to the textures in Figures 6.1(a) and 6.1(b). Most of the best results were obtained with the same sub-band decomposition, Figure 2.6(d), but the performances of the filter bases were less consistent. Further discussions on bases and decompositions follow later.

The co-occurrence method was quite good for most images, with poor relative results only for Figures 6.1(b) and 6.2(c). The AR features, on the other hand, were poor for most images. However, the best filtering approaches yield classifications that are as good as or better than the co-occurrence and AR methods in practically all cases.

It is a quite common practice in texture segmentation to pick training data from

Image in Figure 6.1(a)

Basis	Subband decomp. (fig. no)			
	2.6(a)	2.6(b)	2.6(c)	2.6(d)
Gabor	14.9%	13.0%	18.6%	8.2%
Daubechies 4	15.6%	18.0%	23.8%	8.7%
Daubechies 6	14.8%	18.6%	30.5%	10.7%
Daubechies 8	14.7%	43.5%	34.4%	12.1%
Daubechies 10	18.2%	24.3%	39.3%	15.9%
f16b	9.5%	13.2%	28.4%	8.7%
F_2_1_smpl	8.9%	15.0%	22.8%	12.3%
F_2_1_09	11.2%	21.2%	80.7%	15.7%

Image in Figure 6.2(a)

Basis	Subband decomp. (fig. no)			
	2.6(a)	2.6(b)	2.6(c)	2.6(d)
Gabor	3.0%	1.0%	1.0%	1.1%
Daubechies 4	2.4%	0.7%	2.3%	1.4%
Daubechies 6	1.3%	1.9%	4.0%	2.6%
Daubechies 8	2.5%	3.5%	6.4%	3.4%
Daubechies 10	4.2%	4.8%	7.8%	4.4%
f16b	1.1%	0.5%	1.2%	0.7%
F_2_1_smpl	2.6%	2.5%	3.6%	2.1%
F_2_1_09	2.5%	2.3%	3.4%	2.0%

Image in Figure 6.1(b)

Basis	Subband decomp. (fig. no)			
	2.6(a)	2.6(b)	2.6(c)	2.6(d)
Gabor	60.4%	60.1%	62.7%	54.8%
Daubechies 4	48.8%	46.7%	60.1%	38.2%
Daubechies 6	46.4%	50.8%	63.3%	39.7%
Daubechies 8	46.5%	53.2%	68.9%	43.8%
Daubechies 10	47.3%	57.4%	72.8%	48.7%
f16b	42.4%	45.9%	55.1%	36.4%
F_2_1_smpl	40.7%	43.8%	60.8%	37.1%
F_2_1_09	44.3%	47.0%	58.8%	39.8%

Image in Figure 6.2(b)

Basis	Subband decomp. (fig. no)			
	2.6(a)	2.6(b)	2.6(c)	2.6(d)
Gabor	9.2%	6.9%	7.9%	6.5%
Daubechies 4	15.5%	19.2%	13.1%	5.7%
Daubechies 6	16.1%	20.3%	11.9%	4.7%
Daubechies 8	14.8%	17.8%	13.0%	4.9%
Daubechies 10	15.4%	17.9%	14.5%	4.4%
f16b	16.0%	21.6%	16.7%	8.1%
F_2_1_smpl	15.3%	20.1%	13.8%	7.2%
F_2_1_09	14.8%	21.2%	18.6%	6.7%

Image in Figure 6.1(c)

Basis	Subband decomp. (fig. no)			
	2.6(a)	2.6(b)	2.6(c)	2.6(d)
Gabor	20.5%	18.2%	17.1%	14.3%
Daubechies 4	8.2%	8.9%	12.7%	5.4%
Daubechies 6	9.8%	12.1%	15.1%	9.8%
Daubechies 8	13.9%	15.5%	23.5%	13.3%
Daubechies 10	18.2%	20.1%	27.0%	18.1%
f16b	7.9%	9.1%	13.9%	3.5%
F_2_1_smpl	10.2%	8.9%	13.7%	4.9%
F_2_1_09	9.6%	10.0%	24.6%	4.6%

Image in Figure 6.2(c)

Basis	Subband decomp. (fig. no)			
	2.6(a)	2.6(b)	2.6(c)	2.6(d)
Gabor	2.0%	1.5%	9.5%	1.2%
Daubechies 4	1.3%	1.0%	4.0%	0.6%
Daubechies 6	0.9%	0.7%	4.6%	0.6%
Daubechies 8	1.0%	0.4%	3.3%	1.3%
Daubechies 10	0.9%	0.5%	2.4%	1.2%
f16b	0.3%	0.4%	6.2%	0.8%
F_2_1_smpl	1.5%	2.4%	4.3%	2.3%
F_2_1_09	1.2%	2.3%	6.2%	2.1%

Image in Figure 6.1(d)

Basis	Subband decomp. (fig. no)			
	2.6(a)	2.6(b)	2.6(c)	2.6(d)
Gabor	45.9%	43.8%	53.7%	39.7%
Daubechies 4	41.2%	44.5%	47.0%	40.9%
Daubechies 6	43.2%	47.0%	50.1%	44.3%
Daubechies 8	44.8%	51.9%	53.5%	45.0%
Daubechies 10	46.4%	51.9%	56.9%	48.5%
f16b	47.0%	48.9%	46.6%	39.8%
F_2_1_smpl	47.2%	46.8%	42.9%	41.7%
F_2_1_09	47.1%	49.4%	49.6%	44.2%

Image in Figure 6.2(d)

Basis	Subband decomp. (fig. no)			
	2.6(a)	2.6(b)	2.6(c)	2.6(d)
Gabor	20.9%	16.5%	27.5%	15.6%
Daubechies 4	13.4%	13.5%	6.5%	8.2%
Daubechies 6	12.3%	11.4%	6.4%	6.1%
Daubechies 8	9.3%	8.0%	6.9%	4.7%
Daubechies 10	9.0%	7.4%	7.2%	5.1%
f16b	11.2%	10.9%	4.1%	8.2%
F_2_1_smpl	10.4%	10.4%	6.9%	7.4%
F_2_1_09	10.6%	10.1%	7.6%	6.0%

Table 6.4: Classification errors for different wavelet, Gabor, and QMF full rate filter banks.

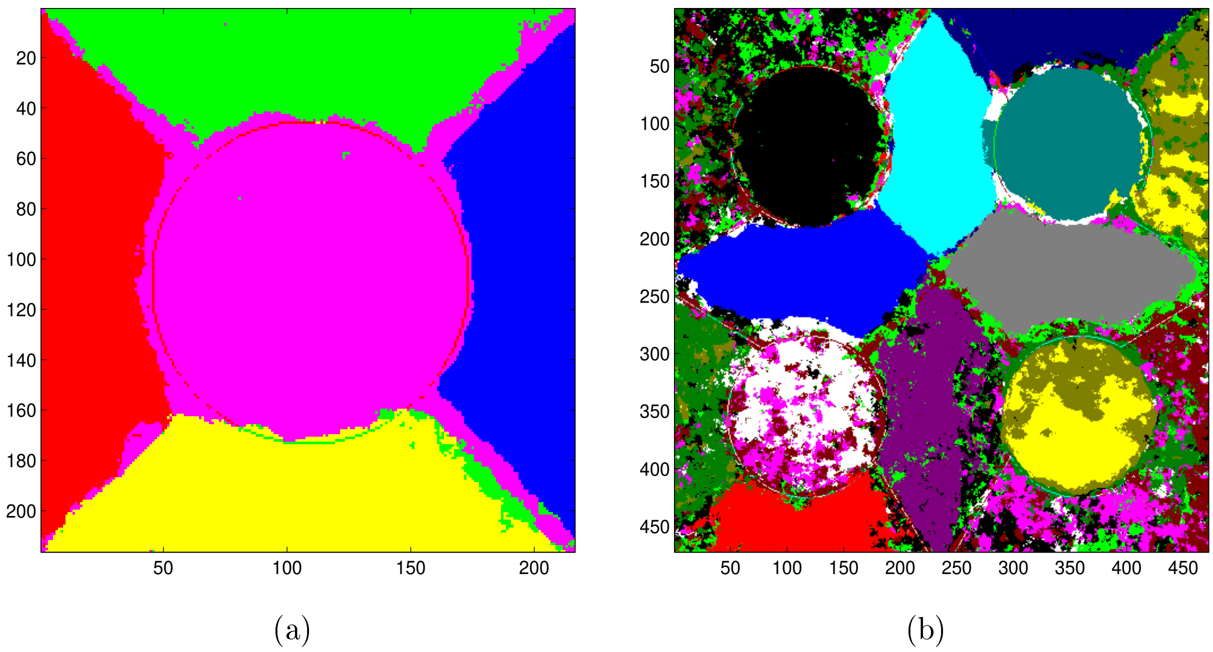


Figure 6.3: Classification results for the QMF “f16b” filter bank for the textures in Figures 6.1(a) and 6.1(b).

the test features, e.g., [32, 51, 104, 123]. This practice has primarily not been used in these experiments. In order to simulate a real application environment, all classifier training has been performed on features from separate subimages. Furthermore, no edge or texture border specific training data are picked. These issues must be kept in mind when comparing the results with previously published results. For a further analysis of the impact of this choice, see Section 2.1.2.

Filter basis

Table 6.4 gives the numbers for the performances of the wavelet frames, QMF, and arbitrary Gabor filter banks. For each column in any of the sub-tables of Table 6.4, the decomposition is constant. Hence, any variation in performance is due to differences in the filter basis. For some of the image/decomposition combinations, there is hardly any impact of the filter basis at all, e.g., Figure 6.2(a). However, for most multi-texture images and some two-texture images, there is a noticeable performance variation as a function of filter basis. For some images, some filter bases are consistently poor, e.g. the Gabor filter for Figure 6.2(d). In other cases, the performance of some of the bases is very dependent on the decomposition, see next subsection.

It can be noted that the larger Daubechies bases are rather poor. This may be caused by two effects. First, the stop-band attenuations of the Daubechies filters are not as good as for the QMF filters. Next, the response is not symmetric, yielding alignment problems with the filter responses and less accurate texture boundary localization.

Many papers propose filtering approaches based on the Gabor basis, e.g., [11, 12, 13, 30, 31, 50, 127, 138, 139]. However, from these experiments, we find no evidence that the Gabor basis is superior. If we may draw any conclusion on the appropriateness of the Gabor basis from these experiments, it is rather the opposite.

Sub-band decomposition

For some images, there is hardly any impact of the decomposition, while for others, the impact is considerable. For the image in Figure 6.1(a), the decomposition has large impact, whereas for Figure 6.2(a), there is hardly any. It appears that for the most complex or difficult images, the impact of the decomposition is large, while it is small for simpler images.

For most images and filter bases, the best results were obtained with the decomposition of Figure 2.6(d). The drawback with this decomposition is the high dimensionality of the feature vectors; 40. This dimensionality imposes high computational complexity. However, for many of the images, the relatively low-dimensional decomposition of Figure 2.6(a) yielded good results.

Another conclusion that may be drawn from the experiments is that increasing the depth of a dyadic decomposition do not improve the results. All the decompositions of Figures 2.6(a-c) are dyadic. Of these, the decomposition of Figure 2.6(a), the most shallow, is overall best

Complexity issues

When comparing the results, complexity issues should be kept in mind. Clearly, most of the filtering approaches are significantly less computationally complex than the co-occurrence and AR methods. For the filtering approaches, filtering and classification are the main contributors to the total complexity of the system. The DCT approach has low filtering complexity (short separable filter masks) and low feature dimensionality, yielding relatively low classifier complexity. However, the results are only moderately good. The Laws, ring and wedge, Gabor, and some of the wavelet and QMF filter banks have high filtering complexities, whereas the IIR QMF filters have low complexities.

The Laws approach yields 25 features, the ring/wedge 11, the dyadic Gabor 20, the DCT 8, and the wavelet, arbitrary Gabor, and QMF filter banks yield from 10 to 40 features. Low feature count yields low classification complexity.

Summary

From Tables 6.3 and 6.4, no single feature extraction method is consistently superior and the impression is rather confusing. The main insight we may learn from this, is that when designing a filter bank for a particular texture problem, it is advantageous to test multiple approaches.

An inherent assumption made when filter banks are used for texture feature extraction, is that the spectral signatures of different textures are different,[1] see Section 2.1. For problems with many textures or spectrally similar textures, it is reasonable to assume

[1] Analog discussions do also apply to other representation spaces than the frequency space.

that the spectral decomposition has a significant effect. This intuitive idea is supported by the experiments, and we see great importance of the bases and decompositions for difficult two-texture images and for most multi-texture images.

In many circumstances, the effort in testing numerous filter banks may be too large. For these applications, one filter bank should be recommended. This filter bank should have a reasonably good performance in most cases. One such choice is the bank with the QMF filter "F_2_1_smpl" using the sub-band decomposition in Figure 2.6(d). This filter bank has very low filtering complexity, see Section 3.2.3, but the feature dimensionality is quite high. An alternative, with lower feature dimensionality and overall second best results, is the decomposition of Figure 2.6(a).

6.2.2 Critically sampled filter banks

Considering the wavelet frame approaches, Table 6.4, relative to the results from the wavelet transform and wavelet packet approaches, Table 6.5, a few conclusions may be drawn. First of all, the filter responses for the wavelet transforms and packets are simply subsampled versions of the ones obtained by the overcomplete wavelet frames. We see that, as expected, using decimated filter outputs overall degrades the results slightly. The major cause is worse edge resolution due to the subsampling. However, the significantly decreased computational complexity (discussed in Section 3.2.3) should be kept in mind.

Next, we see that the trend in the results is the same with the critically sampled approaches as it was with the full rate approaches. That is, image, base, and decomposition combinations that yielded poor relative results in full rate do also yield poor results when critically sampled, and vice versa.

6.2.3 Optimal filters

As has been stressed earlier, a heuristically designed filter bank may yield a tremendous computational complexity. It may also give unsatisfactory results. It is, for instance, noted that the best filter banks in many cases had feature dimensionalities as high as 40. Some reduction in dimensionality may be obtained by feature selection [52], but for good results, a relatively high dimensionality is still needed [52].

Another approach is filter or filter bank optimization. The results from the optimal filtering experiments are presented in Table 6.6. Some of the optimization approaches are only applicable for two-texture cases, and the multi-texture table entries for these are marked by "N/A". The impression with the optimization approaches is much more clear than it was with the heuristic approaches. Some approaches are less powerful and some are rather good.

Despite the fact that the training times for the *back propagation designed masks* were significant, the results were not useful for *any* of the images in this test. Useful results were only obtained for some very simple texture pairs. One example is shown in Figure 6.4. Improvements using a modified scheme have been reported [51], but has for implementational and complexity reasons not been tested here.[2]

[2] The results produced here were actually produced using the same software that was applied for the

Image in Figure 6.1(a)

Basis	Subband decomp. (fig. no)			
	2.6(a)	2.6(b)	2.6(c)	2.6(d)
Daubechies 4	22.8%	22.6%	22.7%	11.9%
Daubechies 6	17.6%	15.2%	29.7%	12.3%
Daubechies 8	21.4%	27.2%	32.6%	13.9%
Daubechies 10	21.0%	26.0%	36.5%	18.9%
f16b	17.2%	15.2%	25.4%	10.8%
F_2_1_smpl	15.5%	16.5%	26.1%	17.6%
F_2_1_09	17.1%	19.8%	26.4%	21.4%

Image in Figure 6.1(b)

Basis	Subband decomp. (fig. no)			
	2.6(a)	2.6(b)	2.6(c)	2.6(d)
Daubechies 4	58.1%	52.5%	66.5%	41.5%
Daubechies 6	56.2%	55.2%	65.8%	44.8%
Daubechies 8	56.7%	58.2%	62.6%	46.8%
Daubechies 10	56.4%	59.1%	65.7%	53.9%
f16b	50.2%	51.5%	54.4%	39.1%
F_2_1_smpl	51.1%	52.9%	58.6%	41.4%
F_2_1_09	52.9%	54.3%	61.3%	45.3%

Image in Figure 6.1(c)

Basis	Subband decomp. (fig. no)			
	2.6(a)	2.6(b)	2.6(c)	2.6(d)
Daubechies 4	15.9%	10.4%	11.9%	4.8%
Daubechies 6	15.9%	12.2%	12.3%	6.4%
Daubechies 8	18.1%	17.1%	20.0%	12.3%
Daubechies 10	20.4%	22.7%	28.8%	14.0%
f16b	13.2%	9.5%	12.0%	4.1%
F_2_1_smpl	16.2%	15.4%	24.5%	11.2%
F_2_1_09	16.8%	16.9%	20.6%	12.0%

Image in Figure 6.1(d)

Basis	Subband decomp. (fig. no)			
	2.6(a)	2.6(b)	2.6(c)	2.6(d)
Daubechies 4	49.9%	53.0%	52.6%	48.0%
Daubechies 6	47.1%	47.4%	56.4%	46.6%
Daubechies 8	51.0%	53.2%	54.9%	49.7%
Daubechies 10	52.5%	56.9%	58.1%	50.1%
f16b	48.2%	43.3%	45.6%	45.1%
F_2_1_smpl	48.3%	47.0%	51.8%	49.9%
F_2_1_09	50.9%	48.2%	54.6%	49.1%

Image in Figure 6.2(a)

Basis	Subband decomp. (fig. no)			
	2.6(a)	2.6(b)	2.6(c)	2.6(d)
Daubechies 4	4.5%	1.3%	0.4%	0.6%
Daubechies 6	4.5%	0.2%	1.2%	1.2%
Daubechies 8	4.9%	2.6%	1.6%	2.2%
Daubechies 10	7.1%	3.6%	3.1%	2.1%
f16b	4.5%	0.8%	0.9%	0.6%
F_2_1_smpl	5.1%	3.4%	5.8%	3.3%
F_2_1_09	4.3%	4.1%	5.8%	2.5%

Image in Figure 6.2(b)

Basis	Subband decomp. (fig. no)			
	2.6(a)	2.6(b)	2.6(c)	2.6(d)
Daubechies 4	22.5%	25.7%	23.4%	11.0%
Daubechies 6	22.2%	23.3%	19.7%	11.2%
Daubechies 8	21.5%	25.9%	21.7%	8.5%
Daubechies 10	18.8%	19.8%	20.4%	8.2%
f16b	22.2%	24.2%	17.5%	10.5%
F_2_1_smpl	22.3%	22.4%	19.4%	11.7%
F_2_1_09	20.6%	22.7%	18.4%	10.6%

Image in Figure 6.2(c)

Basis	Subband decomp. (fig. no)			
	2.6(a)	2.6(b)	2.6(c)	2.6(d)
Daubechies 4	2.8%	1.1%	2.4%	1.3%
Daubechies 6	2.7%	1.8%	3.6%	1.4%
Daubechies 8	2.2%	1.3%	1.8%	1.3%
Daubechies 10	2.9%	0.8%	1.6%	1.5%
f16b	1.0%	0.8%	1.0%	0.5%
F_2_1_smpl	2.4%	3.1%	6.7%	3.9%
F_2_1_09	2.3%	3.2%	7.1%	3.4%

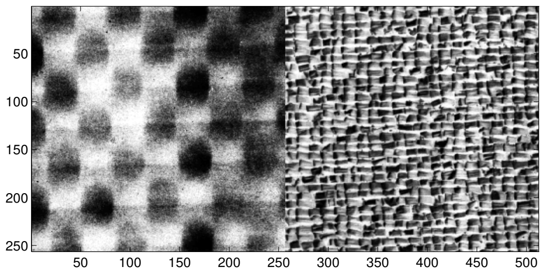
Image in Figure 6.2(d)

Basis	Subband decomp. (fig. no)			
	2.6(a)	2.6(b)	2.6(c)	2.6(d)
Daubechies 4	15.5%	15.6%	9.7%	10.0%
Daubechies 6	12.9%	13.2%	12.1%	7.9%
Daubechies 8	10.6%	8.1%	7.3%	6.8%
Daubechies 10	8.5%	8.4%	9.1%	7.3%
f16b	10.7%	11.0%	7.8%	7.8%
F_2_1_smpl	9.5%	9.7%	9.0%	6.9%
F_2_1_09	9.9%	11.4%	11.3%	7.4%

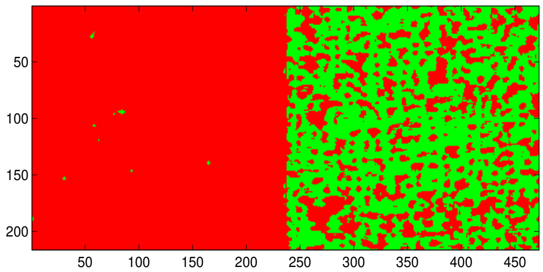
Table 6.5: Classification errors for different wavelet transforms/packets and critically decimated QMF filters.

Method	Test image (figure number)							
	6.1(a)	6.1(b)	6.1(c)	6.1(d)	6.2(a)	6.2(b)	6.2(c)	6.2(d)
Eigenfilter	12.1%	56.1%	6.9%	37.1%	4.2%	4.0%	3.7%	4.7%
Opt. repr. Gabor filter bank	7.2%	37.2%	18.6%	41.9%	4.8%	5.5%	5.8%	36.2%
Prediction error filter	14.7%	34.7%	2.0%	44.6%	0.6%	0.9%	4.6%	31.7%
Optimal Gabor filter $\sigma = 2$	N/A	N/A	N/A	N/A	5.0%	10.4%	3.6%	10.8%
Optimal Gabor filter $\sigma = 4$	N/A	N/A	N/A	N/A	4.7%	5.3%	3.3%	11.6%
Optimal Gabor filter $\sigma = 8$	N/A	N/A	N/A	N/A	8.2%	5.7%	5.2%	11.4%
Optimal Gabor filter $\sigma = 16$	N/A	N/A	N/A	N/A	13.3%	14.0%	13.7%	24.2%
4-filter optimal Gabor filter bank	24.7%	54.0%	24.3%	52.8%	7.0%	4.3%	3.8%	18.7%
6-filter optimal Gabor filter bank	16.3%	49.4%	11.2%	50.3%	6.7%	3.8%	3.8%	17.1%
10-filter optimal Gabor filter bank	16.1%	40.7%	10.0%	39.9%	6.7%	4.1%	3.9%	12.8%
J_{MS}	16.9%	47.3%	10.0%	59.7%	0.9%	0.7%	8.3%	28.6%
J_U	12.7%	45.6%	4.0%	35.9%	0.9%	0.7%	2.1%	5.1%
J_F	N/A	N/A	N/A	N/A	0.9%	0.7%	2.1%	5.1%
BackProp. NN, mask size 11	47.4%	61.1%	74.1%	69.3%	45.3%	32.1%	30.0%	43.6%
BackProp. NN, mask size 21	72.4%	84.9%	74.7%	76.5%	48.5%	30.8%	28.8%	48.6%

Table 6.6: Classification errors for different optimized texture feature extractors. See Table 6.2 for an overview of the method descriptions.



(a)



(b)

Figure 6.4: (a) Texture pair D8-D84 [14] and (b) supervised segmentation by the back propagation designed mask [51].

The *eigenfilter* approach is never emphasized with either remarkably good or remarkably bad results. However, it must be remembered that the number of filters is relatively high with this approach. For two-texture images, the number of filters is typically in the range 14-18, while for multi-texture images it is 40 or close to 40. Hence, the argument of low complexity versus the heuristic filters is not applicable.

The *optimal representation Gabor* filter bank was the only optimized filtering approach doing a really good job on the image of Figure 6.1(a). However, for most of the other test images, it is only giving similar, and even in some cases significantly worse results than the best optimal filtering approaches, despite higher feature dimensionality. The poor performances for the images of Figures 6.1(c) and 6.2(d) are particularly noted.

original article [51], provided by Kalle Karu. The results should therefore be representative. The software for the modified scheme was not available.

The best results for the images in Figure 6.1(b), 6.1(c), and 6.2(a) in this study were obtained with the *prediction error filters*. However, despite these nice results, a remarkably poor result was obtained with the image of Figure 6.2(d) and a relatively poor result with the image of Figure 6.1(a). Hence, this approach is not very robust. The cause for the problem is most likely that the approach is optimal with respect to representation, not separation, as discussed in Section 2.3.1.

The results from the *optimal Gabor filters* and *optimal Gabor filter banks* are generally good, but the results indicate problems with similar textures – especially when using only one or a few filters. In particular, the results for the images of Figures 6.1(a), 6.1(c), 6.2(a), and 6.2(d) are noted. On the other hand, these approaches yield one-dimensional feature spaces for two textures. Hence, class labels may be assigned to the feature values by thresholding. Thresholding has a very low computational complexity compared to classifiers like the LVQ. Furthermore, training the threshold classifier means determining *one* scalar parameter per feature image. This is clearly a simpler task than training the parameters of, say, the LVQ classifier.

FIR filters optimized with respect to J_{MS} show remarkably poor performance for some images, in particular the texture pair of Figure 6.2(d) is noted. Furthermore, the results are unacceptable for the images in Figures 6.1(a), 6.1(b), 6.1(d), and 6.2(c). From this, we may conclude that these filters are not very robust. The number of resulting filters for the images in Figure 6.1 are 4, 46, 3, and 12, respectively.

Considerably better results are obtained by filters optimized with respect to J_U and J_F. For two-texture images, these filters are clearly the winners of this study. Overall, they outperform all other approaches, irrespective of number of features, for texture-pair images. Since these approaches yield only one feature image that may be classified by thresholding, they are also the least computational demanding for two-texture images.

Even for multi-texture images, filters optimized with respect to J_U are among the best optimal filters, but the performance is not of the same extra-ordinary quality. However, performance issues should be kept in mind. For the multi-texture images of Figure 6.1, there are 4, 25, 5, and 15 filters in the filter bank, respectively. Classification is done by thresholding each feature image once, thus the classification complexity is low. No multi-texture approach has yet been developed for the J_F criterion.

6.2.4 Discussion of the test images

Clearly, some images yield more differences between the methods than others. Specifically, this applies for the images in Figures 6.1(a), 6.1(c), 6.2(b), and 6.2(d). Apparently, the textures in these images are difficult to discriminate between, and some of the least robust or powerful methods experience problems.

6.3 Summary

We have seen how various filtering approaches yield different results for different images. No single approach did perform best or very close to the best for *all* images, thus no single approach may be selected as the clear "winner" of this study. In addition to the classification error issue is the issue of computational complexity. If we take the classification error and computational complexity into consideration, the following conclusions may be drawn:

- Much of the focus in the filter *optimization* approaches is on a low feature count, thus many of the optimalization schemed yield nice computational characteristics. In most cases, some of the optimal techniques were best or very close to the best techniques. Especially the filters optimized with respect to J_U or J_F had low classification errors. Hence, these filters should be selected for many applications. However, some problems with multi-texture images can be observed.

- For multi-texture images, the optimal prediction error filter approach may be a good candidate. It yields low feature extraction complexity and very low classification complexity. However, there is a risk that some textures are not discriminable by the approach.

- If the performances of the optimal filters or filter banks are too poor for a specific application, several heuristic filter banks should be tested. The results are not very conclusive on which approach to select, but both QMF and wavelet frame filter banks should be considered. However, compared to the optimal approaches, this implies increased computational complexity.

- The classification error increases slightly when using critical sampling, as opposed to overcomplete representations for the wavelet and the QMF approaches. However, the computational complexity decreases formidably. A critically sampled representation may therefore be selected before an overcomplete, if the increased classification error can be tolerated.

- The dyadic decomposition (also known as wavelet transform decomposition or octave band decomposition) may yield features with good separating properties. However, increasing the depth (i.e., adding more sub-bands) do not improve the classification results. Hence, we may conclude that important texture information is located in other sub-bands than the low-pass band, supporting the findings of Chang and Kuo [16].

- The computational complexities of the filtering approaches are generally low compared to the two other approaches tested here. Since the results were equally good or better for the filtering approaches, the filtering approaches should therefore be preferred.

Chapter 7

Document segmentation and image queries

In this chapter, two application examples of the QMF filter bank features of Section 3.2 will be presented. The motivation is to show the applicability of the texture features of Section 3.2, not to present complete systems with all pre- and post-processing for the applications. In Section 7.1, an approach to document segmentation is presented and in Section 7.2 an application to image content search is presented. A common denominator for the two approaches presented here is that the search is for known textures in images containing other textures not known a priori. Hence, several commonly used classifier design approaches [35] are not applicable here. Furthermore, for the same reasons, the filter optimization approaches presented in Chapters 4 and 5 are not applicable.

7.1 Document segmentation

An enormous amount of printed material is available on paper. The digital computer and computer networks have made it possible to search for and retrieve electronically stored documents in seconds, no matter where in the world they are stored. This is not the reality for documents stored as paper copies and there is a considerable interest in digitizing paper documents.

The straightforward approach to document digitization is to scan the whole document and store it as a bit-map representation. This approach was taken for the CD-ROM accompanying the proceedings of the IEEE 1994 International Conference on Acoustics, Speech & Signal Processing (ICASSP'94). Unfortunately, this requires a huge storage capacity (even with modern image compression techniques) and does not enable search for textual content in the documents.

A more efficient way of representing documents is to separate the text and the graphics and store the text in a character representation form. Therefore, several approaches to text-graphics segmentation have been proposed. The most popular approaches may broadly be classified as either *top-down* or *bottom-up* approaches [122]. One of the most common top-down techniques is the *run length smoothing* method [17, 135] (also known as the *constrained run length* method). With the run length smoothing algorithm, the document image is processed along vertical and horizontal runs (scan-lines). If the number of white pixels between two black pixels is less than a threshold, the white pixels are converted to black. Thereby, white gaps within and between words are converted to black, while white gaps between columns remain white. A similar discussion applies for lines. Hence, the run length smoothing method is sensitive to font-size, character spacing, line and column spacing and to some extent to the orientation of the document.

Another common top-down technique is the *recursive projection profiles* method [59, 76, 136] (also known as the *recursive X-Y cuts* method). The basic idea underlying the projection profiles method is that the numbers of black pixels in the columns and rows of the image are projected onto the horizontal and vertical axes, respectively. Hence column and line spacings generate detectable valleys in the projection profiles and it is possible to extract the block structure of the document. The recursive projection profiles method is restricted to rectangular blocks of known orientation.

The bottom-up methods are typically variants of the *connected components* method [33]. The connected components method basically extracts connected components, like a character or a connected part of a drawing. It then performs some kind of analysis on each connected component and merges connected components that are close to each other. The drawbacks with the connected components method are that it is character size dependent, sensitive to inter-line and inter-character spacings, and sensitive to the digitizing resolution.

In addition to the already mentioned restrictions with these methods, they typically operate on thresholded images. If the image capturing conditions are difficult, appropriate thresholding may be difficult.

An interesting ability of humans is the ability to tell what is and is not text in a document viewed from a distance too far away for letter discrimination. This suggests that textural clues are important for humans in separating text regions from non-text regions. In this section, an approach to segmentation of text and graphics in scanned documents is proposed. The approach is based on the assumption that the text in a document may be viewed as one texture, while the graphics is represented by different textures. Then, the documents are segmented with a texture segmentation scheme using the sub-sampled feature extraction scheme of Section 3.2. Both unsupervised and supervised segmentation schemes are proposed. While most traditional text-graphics segmentation schemes require a priori knowledge of the input, the unsupervised scheme is independent of document layout, typeface, font size, scanning resolution, etc. However, the unsupervised segmentation scheme shows a somewhat limited performance for difficult documents. By sacrificing some of the flexibility, it is shown that the segmentation is improved by using a supervised classifier. In Subsection 7.1.1, the unsupervised segmentation scheme is proposed and in Subsection 7.1.2, the supervised segmentation scheme is proposed. In both cases experimental results illustrating the performances of the suggested approaches are presented.

An approach similar to the unsupervised approach [49], uses the Gabor filter and a local energy function as the feature extractor, see Section 2.2.3. One major drawback with the Gabor approach is its computational complexity. The approach presented in [49] uses a subset of eight filters from the total set of 20 suggested in [50]. However, even when this restricted subset is used, a total data amount of eight times the input is used by the classifier. The approach presented here has much lower complexity, without any observed sacrifices in segmentation quality.

In another similar approach, supervised segmentation using eigenfilters derived by the use of neural networks are examined [84]. By this approach, good results are obtained, realizing some savings over the previous approach. However, the extension to unsupervised problems is not straight forward.

7.1.1 Unsupervised segmentation

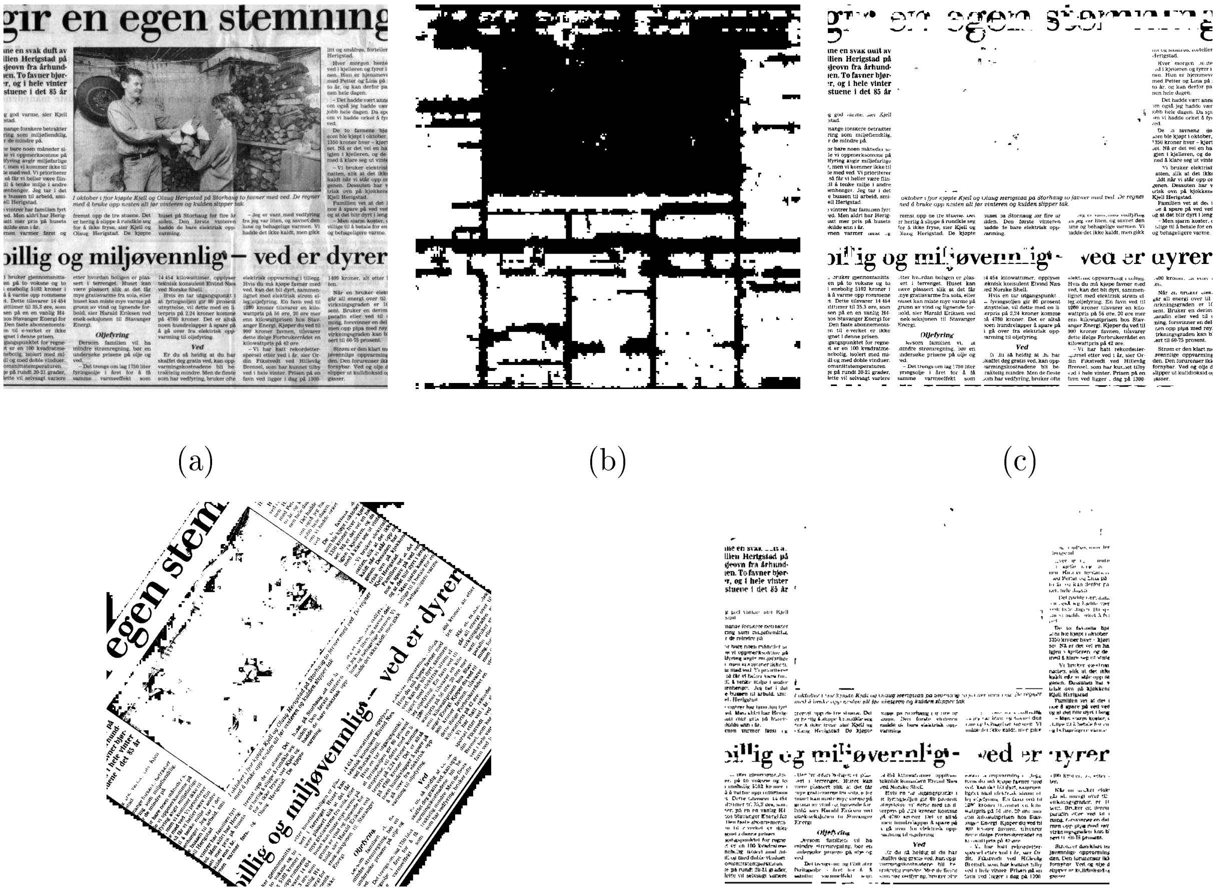
From the results of Chapter 6, it is evident that the critically sampled IIR QMF filter banks of Section 3.2 give quite good results. Furthermore, the computational complexities are lower than for most other filter bank approaches, see Section 3.2.3. Consequently, a critically sampled IIR QMF filter bank, using the filter structure of Figure 2.6(d), will be used as feature extractor in these experiments. Furthermore, the sigmoidal nonlinearity and Gaussian smoothing filter are used in these experiments. Using the critically sampled filter bank implies calculation of only a few feature samples. The total size of the feature set is therefore identical to the size of the input image. Critical sampling introduces some edge inaccuracies, see Section 3.2.4, but the computational and memory-requirement savings are tremendous. The approach by Jain and Bhattacharjee [49], for example, implies a feature set eight times the size of the input image. An A4 page scanned at 300 dots per inch, for example, consists of more than eight million pixels, thus the size of the feature set is important.

If the features give rise to correct unsupervised segmentation, we will not have to make any assumptions about font size, scanning resolution, column layout, orientation, etc., as we will see in this section. The clustering algorithm chosen is the k-means algorithm [116] with a predetermined number of clusters.

It has been chosen to use parts of pages scanned from the March 2, 1994, issue of the newspaper “Stavanger Aftenblad” (SA), Stavanger, Norway as test images. For comparison, some images used in [49, 128] have also been included. Figures 7.1(a-c) shows a part of page 8 in SA and the quite successful segmentation of the document. Except for some problems with the heading and spurious dots in the text regions, the segmentation is as desired. Note that no a priori information on font size, scanning resolution, etc. has been assumed. This example shows the applicability of the method. To further emphasize the flexibility of the approach, Figures 7.1(d) and 7.1(e) show the segmentation results from the same page rotated by $55°$ and at half the resolution, respectively. These figures show one important result: the segmentation quality is practically invariant with respect to scaling and rotation.

How does the method compare with other methods? Figure 7.2 show the same images as used in [49, 128]. The segmentation results from these images are practically the same. On these images, the sub-sampling did not yield any particular degradation in the segmentation result. These results, in combination with the results of Chapter 6, indicate that there is no practical need for using the significantly more complex Gabor filter bank for this application.

Unfortunately, not all documents are as easy to segment as the ones presented thus far. The text segmentation method is based on the assumption that the text regions may



(a) (b) (c)

(d) (e)

Figure 7.1: (a) The 1024×1024 pixel scanned image of parts of page 8 from SA, (b) the segments from unsupervised segmentation, and (c) with regions classified as picture excluded. (d) Unsupervised segmentation of the same page rotated by 55°, and (e) scanned at half the resolution, 512×512 pixels.

be considered as constituting a significantly different texture than the graphics regions. Therefore it is natural that the method performs worse when the graphics contain textures similar to the texture in the text region. The texture of parts of the graphics region in the document in Figure 7.3(a) is quite similar to the texture of text. As a result, we see that the segmentation, Figure 7.3(b), has included a considerable part of the graphics into the text cluster. No unsupervised experimental results on similarly difficult data have been found in other publications.

It is important to be aware that all experiments in this section were done with no assumptions on the input image and with no post-processing. Several kinds of post-processing techniques are possible and should improve the results, especially if we have information about the format of the input, e.g. that the text is organized in rectangular columns.

(a) (b) (c) (d)

Figure 7.2: (a), (c): Test images from [49] and (b), (d) the unsupervised segmentations of the images.

(a) (b)

Figure 7.3: (a) Page 2 from SA and (b) the unsupervised segmentation of the page.

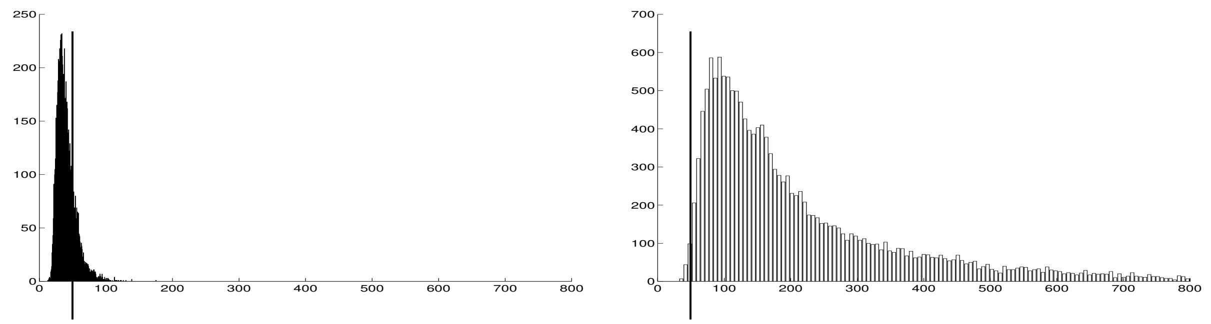
7.1.2 Supervised segmentation

Motivated by the problem with the unsupervised segmentation approach in the previous section, a somewhat different approach is proposed, utilizing supervised segmentation. A supervised approach will of course mean a limitation when compared to the flexibility of the unsupervised approach. However, in many applications it may be assumed that the font-size, scanning resolution, and page orientation are fixed. This will probably be the case if for instance a newspaper decides to digitize back issues.

A supervised approach was also presented by Jain and Bhattacharjee [49]. The specific application where the supervised approach was applied was segmentation of bar-code from text. Their supervised approach was based on training a classifier using exemplar features from bar-code and text. However, in a general text segmentation application it is not possible to assume that the non-text class is this well-defined. It will therefore not be possible to train a classifier on the non-text regions.

Training and classification

Although it is not possible to assume that the non-text class has fixed properties, it may be assumed that the text class has fixed and well-defined properties. To classify, we need



(a) (b)

Figure 7.4: Histograms of distance from codebook for feature vectors from (a) text and (b) image for page 8 of SA.

to calculate some kind of measure of similarity with the prototype text class. The training feature vectors were represented by a set of cluster center vectors obtained by clustering with the k-means clustering algorithm [116].

The means and variances of each cluster are computed from the features. The distance from the query pattern is computed by the squared weighted Euclidean distance to the closest cluster center. This measure is given by

$$d = (\mathbf{v} - \mathbf{m})^T \operatorname{diag}(\mathbf{C})^{-1} (\mathbf{v} - \mathbf{m}), \tag{7.1}$$

where $\mathbf{v}$ is the investigated feature vector, $\mathbf{m}$ is the cluster mean, and $\mathbf{C}$ is the cluster covariance matrix. The idea of clustering and distance computation is similar to the basic ideas of the Kohonen Self Organizing Map and the Learning Vector Quantization [58].

A feature vector, corresponding to one pixel or a small region of the image, is classified as belonging to the target pattern if the distance is below some threshold. An empirical distance threshold of 50 from the mean was selected. This choice was appropriate when the features were scaled to approximately the range $0 \cdots 255$. It is important to keep in mind that techniques like Bayes decision rule [29] may not be employed to determine an optimal threshold for classification as text because we do not know the non-text properties.

Experimental results

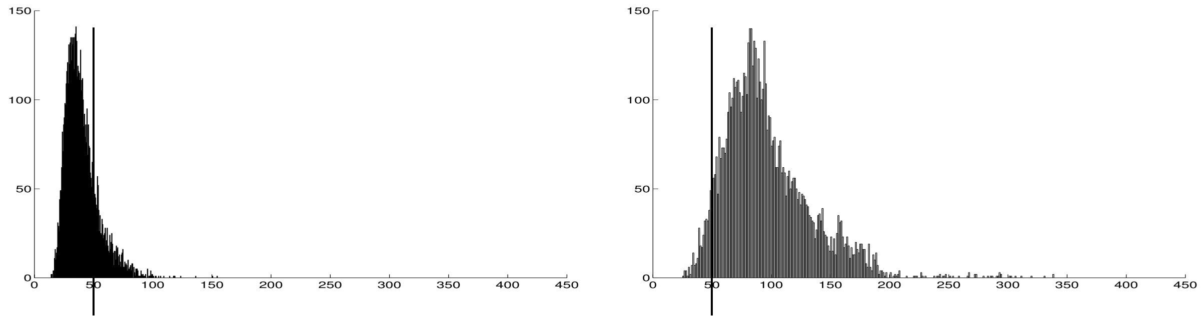
Two documents have been chosen for the reported supervised experiments. First, the results with page 8 from SA are shown, a page that was successfully segmented by the unsupervised approach. Then it is illustrated how the method performs for a more difficult page, page 2 of SA.

Page 8 of SA, Figure 7.1 (a), contains four different kinds of text; normal text, boldface text in the article introduction, and headings of two sizes. Attempts to classify the document according to each of these text types have been made, using codebooks for each of the four kinds of text. If a feature vector is close to several of the text classes, it is classified as the class with the closest prototype.

How good is the separation between text and non-text regions? Figures 7.4(a) and 7.4(b) show the distance histograms between the text codebook and the text and graphics

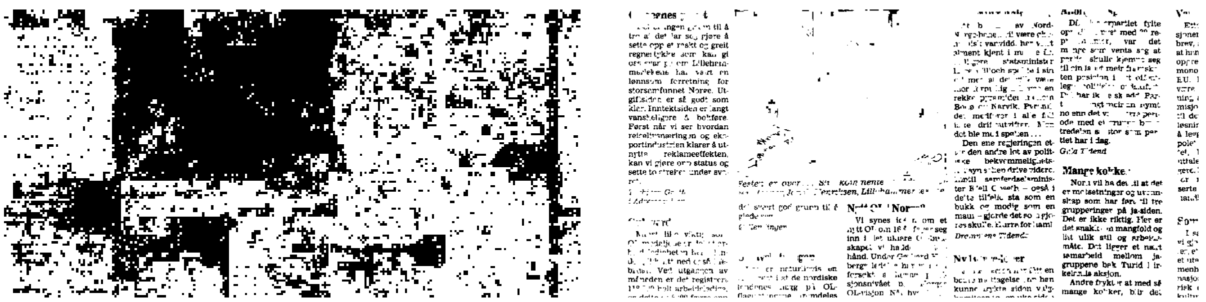
(a) (b)

Figure 7.5: Supervised segmentation of page 8 from SA, (a) shown as regions (black is non-text regions, dark gray is normal text, medium gray is boldface text and light gray and white are large and small headings, respectively) and (b) the input image masked through the segmentation result (non-text regions masked out).



(a) (b)

Figure 7.6: Histograms of distance from codebook for feature vectors from (a) text and (b) image for page 2 of SA.



(a) (b)

Figure 7.7: Supervised segmentation of page 2 from SA, (a) shown as regions and (b) the input image masked through the segmentation result (non-text regions masked out).

features, respectively. We see that the histograms are well separated. From the histograms we understand why the threshold of 50 is appropriate.

The segmentation results from page 8 are given in Figure 7.5. As we see, the segmentation is mainly correct, but with some small regions misclassified. These misclassified regions should be possible to detect and remove with various post-processing techniques, which are not addressed here. The promising results from the supervised approach were expected, since the results with the unsupervised approach were good.

The main motivation for developing a supervised approach was improvement of the results for the more difficult documents, like page 2 of SA. The feature distance histograms for this page are given in Figure 7.6. As we could expect, we see that the histograms have more overlap, but a text classification distance threshold of 50 still seems reasonable. The segmentation results are given in Figure 7.7. We see that only small parts of the graphics region are classified as text, while large parts were classified as text in the unsupervised approach. Hence, the results are much better with the supervised approach than with the unsupervised approach.

7.2 Query by image content

One of the major contributions of computers to everyday life is the ability to conduct fast and reliable text pattern searches. Text pattern searches allow us to find names in databases, books based on their keywords, etc. Searching non-textual data such as images based on contents (e.g., color and texture), on the other hand, is far more complex. However, the increased utility of computer vision and pattern recognition algorithms, along with the availability of more powerful computers at low cost, make such an application increasingly feasible. In an image database system, features may be computed in advance and the difficult issues are feature storage and comparison. In searches in images not stored in an integrated database, such as searches over a network, pre-computed features will typically not be available and the feature extraction complexity is of paramount importance.

Query by image content is a relatively new research area and an increasing number of approaches to querying by color and texture are reported. Picard et al. [88, 90, 91] focus their work on user interaction and adaptive feature selection. Claiming that no single feature extraction scheme is suitable for all image types, they apply a "society of models". Some works on retrieving a set of most similar sub-images based on matched filters [89], Markov random field parameters [37], and Gabor features [70] have been presented. Taking a more global perspective, Pentland et al. [86] and Flickner et al. [34] present complete systems, integrating many aspects of query systems. Except for the work by Liu and Picard [64] on modeling the human visual system, not much work on feature tuning for image queries has been reported.

In this section, a feature extraction scheme for color and texture content search is suggested and tested on a large number of real world images. In Subsection 7.2.1 the feature extraction and classification scheme is proposed and in Subsection 7.2.2 experiments are given.

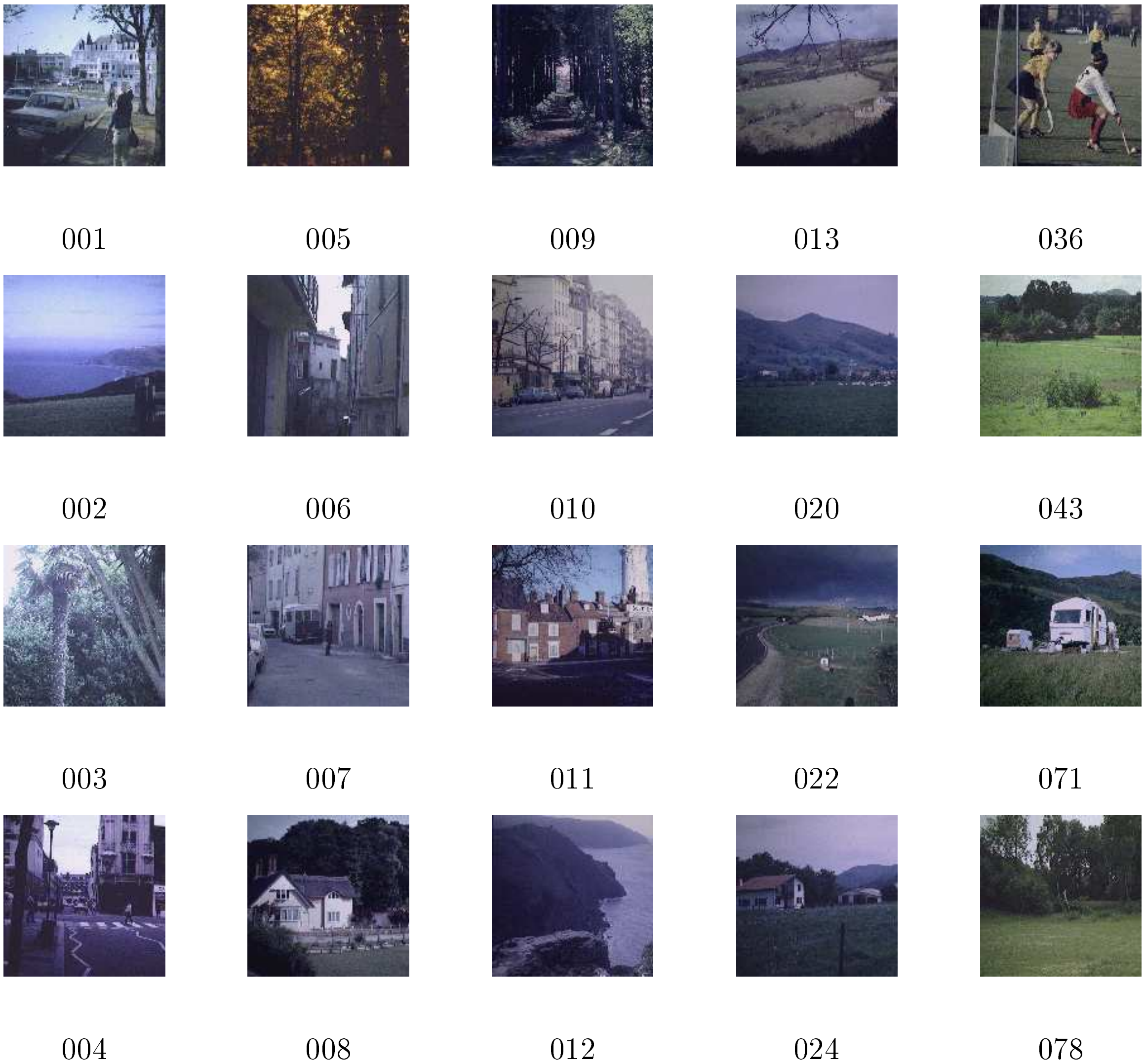


Figure 7.8: Subset of the test images used for image content search/labeling, downloaded from [73].

7.2.1 Methodology

In this work, a method for color and texture search is developed. The texture features will be extracted by the computationally attractive critically sampled IIR QMF filter bank scheme presented in Section 3.2. Color is very important for many query patterns. In this case, the color is represented in the feature space by the two chrominance components of the NTSC color representation [48]. The texture features are extracted from the luminance component.

How are the color and texture features combined in the classification? The straightforward approach is to combine them into one vector by appending the individual feature vectors. However, for filter banks with many sub-bands, this would imply high dimensionality in the *texture* feature sub-space while the *color* sub-space is limited to dimensionality two. Hence, large distance variations in color would only yield a low relative variation in the overall distance measure. Another scheme, yielding identical weight on color and texture, would be to classify using the color and texture feature vectors separately, yielding one color-based class map and one texture-based class map. An overall class map could then be formed by logical "and" of these two class maps. Both of these approaches were tested with relatively poor results.

Yet another approach is to append the color and texture features into one vector and balance the contributions from the color and the texture. The balancing may be done in the distance measure. A modification of the codebook design approach of Section 7.1.2, using a balanced squared weighted Euclidean distance measure, is applied. The distance measure is

$$\begin{aligned} d = & \frac{1}{2^2} \left(\mathbf{v_C} - \mathbf{m_C}\right)^T \operatorname{diag}\left(\mathbf{C_C}\right)^{-1} \left(\mathbf{v_C} - \mathbf{m_C}\right) \\ & + \frac{1}{N^2} \left(\mathbf{v_T} - \mathbf{m_T}\right)^T \operatorname{diag}\left(\mathbf{C_T}\right)^{-1} \left(\mathbf{v_T} - \mathbf{m_T}\right), \end{aligned} \tag{7.2}$$

where the C and T subscripts denote the sub-vectors and sub-matrices corresponding to the color and the texture features, respectively and N is the dimensionality of the texture feature vector. By this normalization, it may easily be shown that linear scaling of the distance in the color and texture domains affect the total distance measure identically.

In an image database system, query times of a few seconds, even if the database consists of thousands of images, are desirable and have been reported [90]. However, although the complexity of the proposed feature extraction scheme is very low, see Section 3.2.3, even loading all those images will violate the complexity constraints. A possible solution to this problem is to store a small amount of feature information for each image and apply a fast similarity match [9, 90, 141]. Another alternative is to classify the regions of the images off-line and store class labels. This will be illustrated by experiments in the next subsection. Hence, if a user requests images with hair, the class labels are examined and images with hair are returned. The problem of integration with higher-level information, e.g. context information, will not be treated here.

7.2.2 Experiments

An example of indexing regions with grass using the 98 vacation images in the MIT VisionTexture database [73] will be given. Some of the MIT images are shown in Figure 7.8. One of the most challenging tasks in the experiments was the selection of good training image regions. Eventually the images 008, 013, 020, 022, 024, 036, 043, 071, 078, and 095 were used as training images. When testing on any of the training images, the leave-one-out scheme was used.

How do we determine which sub-band decomposition is most appropriate for the problem? The selection of sub-band decomposition is clearly a tradeoff between the "curse of dimensionality" [29] and the need for adequate feature separation. Furthermore, classical feature analysis and selection schemes, like the scheme of Whitney [142], are not adequate for this problem due to the lack of a definition of the "everything else" class.

In the experiments, several sub-band decompositions were tested along with the dyadic Gabor features [50, 70], Section 2.2.3. The distance distributions along with the corre-

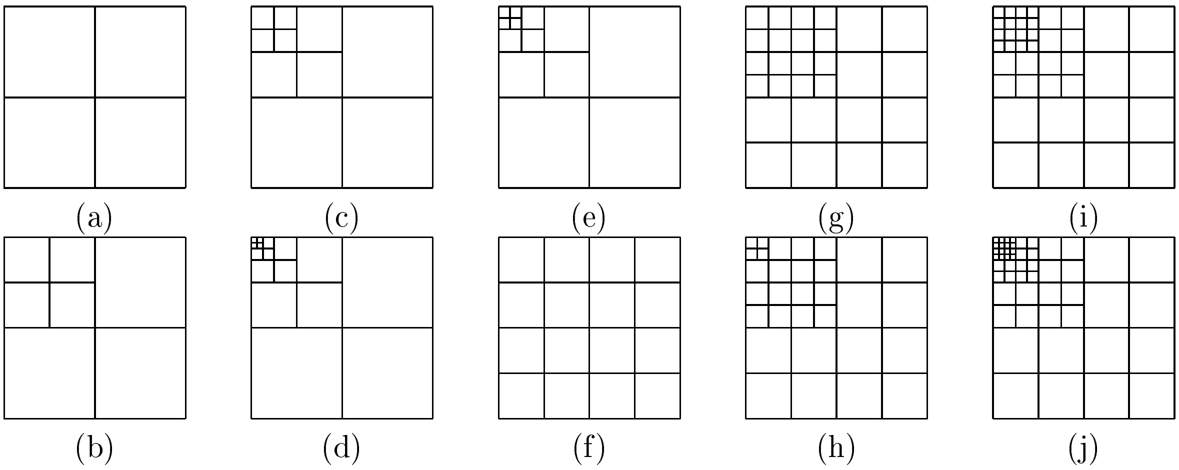


Figure 7.9: Sub-band decompositions of the filter banks used in the image content search experiments.

sponding overlaps were computed and are shown in Figure 7.10. From the overlap percentages, we can see that the filter bank structure of Figure 7.9(h) yields the most powerful features. We also see that it compares favorably with the computationally *much* more complex Gabor [50] features, Figure 7.10(l).

It is particularly interesting to see that some of the texture features hardly add any discrimination at all, and some even makes the results worse! From Figure 7.10(k) we see that the overlap without any texture measure applied at all is 20.2%. Many texture measures give results that are only slightly better. The filter structures of Figures 7.9(e) and 7.9(j) even makes the results worse, 29.5% and 33.8% overlaps.

In Figure 7.11 some classification results with this feature extractor are shown. Different intensities indicate different thresholds for the distance measure. As can be seen, if the results are studied in detail, the system detects regions with grass well. Some false detections of walls and leaves can be seen, but from a pure texture and color point of view, the falsely detected areas are very similar to certain types of grass. The experiments were restricted to grass as an illustrational example.

The results also tell us that as the dimensionality of the texture features becomes too large (e.g. Figures 7.9(j) and 7.10(j)), the performance degrades. This indicates that it becomes increasingly difficult to design a good codebook with the increased number of feature variables – the "curse of dimensionality" [29].

7.3 Summary

In this chapter, applications to content based image search and document segmentation of the critically sampled QMF filter bank scheme of Section 3.2 have been presented. The first application was an approach to document segmentation using an unsupervised classifier. This approach uses no a priori knowledge of the input, as opposed to most traditional text-graphics segmentation schemes. Still, satisfying results are obtained in many cases. The second approach was a supervised classification approach, yielding near

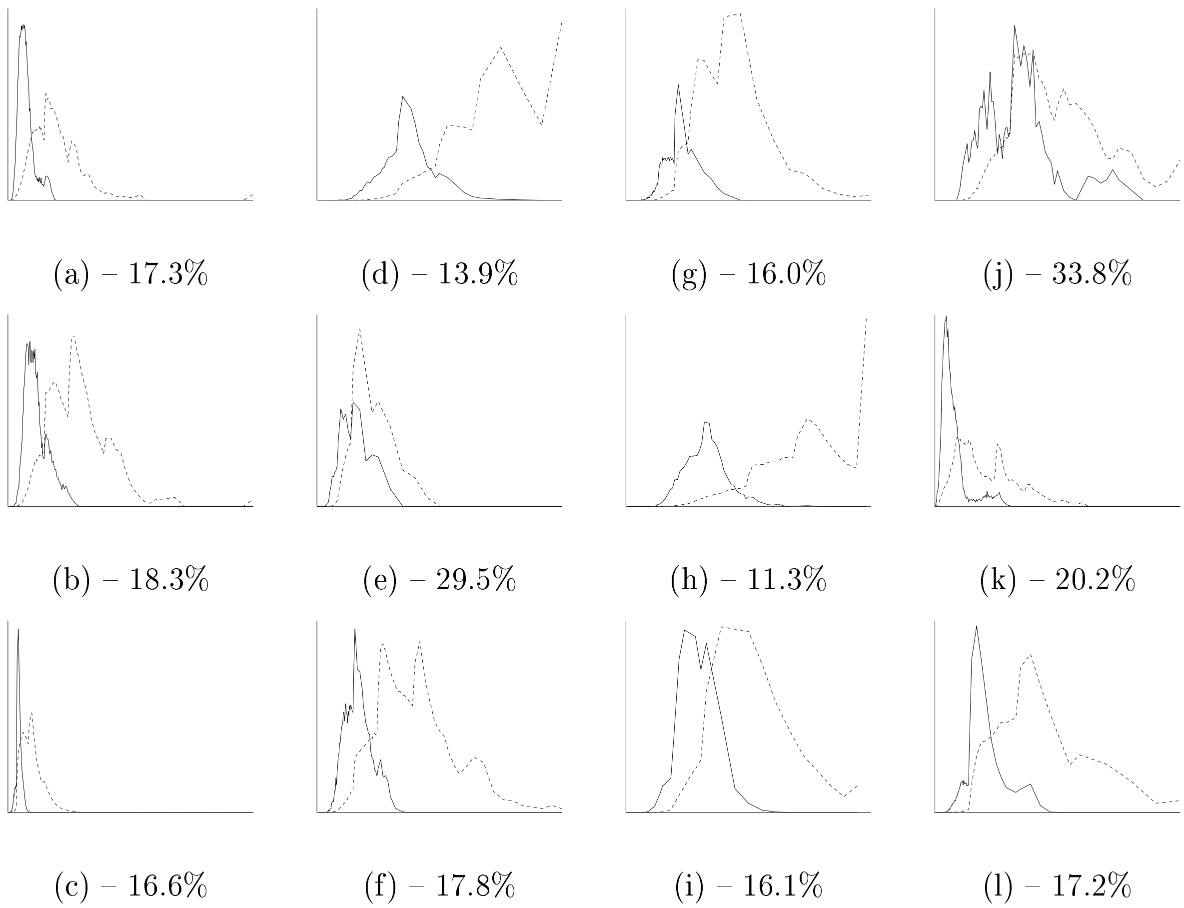


Figure 7.10: Histograms of feature distance to closest grass codebook vector for grass (solid) and non-grass (dashed) for different feature sets. The horizontal axes correspond to distance measure and the vertical to relative densities. Minimal overlap is optimal. Histograms (a) to (j) correspond to the filter structures in Figure 7.9, (k) corresponds to no texture filter bank at all (using only the luminance in addition to the color), and (l) to the dyadic Gabor filter bank [70]. The percentages of histogram overlaps are indicated.

perfect segmentation even for difficult documents where the unsupervised approach had poor segmentation.

Furthermore, a feature extraction and classification scheme for color-texture pattern search in digital image databases has been proposed. The color was represented by the two chrominance components in the NTSC scheme, while texture was represented by QMF filter bank features. The color and texture features were combined and their contributions balanced using a modified weighted Euclidean distance measure. The scheme was tested on the problem of labeling or finding regions in images with grass.

The focus of the applications has been on proving the applicability of the critically sampled QMF filter bank.[1] The issues of pre- and post-processing and integration of other types of information have been treated more superficial. This is not to indicate that these problems are less important, but rather that they are beyond the scope of this work.

[1] Both this and the previous application, document segmentation, are problems where only a subset of the possible texture classes are known a priori. Hence, the optimization approaches of Chapters 4 and 5 are not applicable.

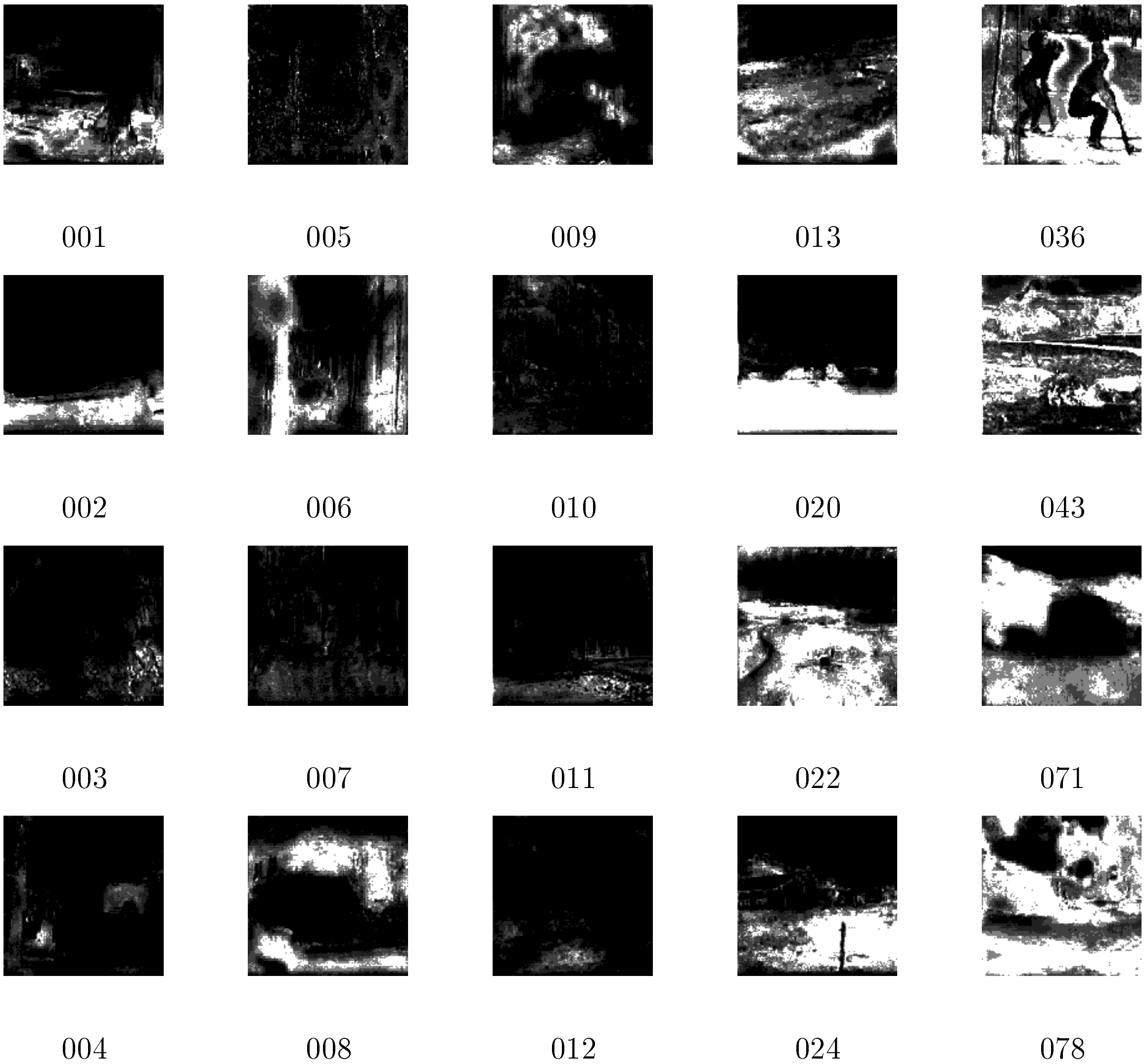


Figure 7.11: Segmentation results for the images of Figure 7.8.

Chapter 8

Automated seismic interpretation

The trade with petroleum is of considerable importance for the world economy and exploration and production of petroleum is a large business. Most petroleum resources are buried under the surface of the earth. Imaging properties of the subsurface is consequently a major task in petroleum exploration. Seismic analysis is one of the principal tools for mapping the subsurface properties.

In this chapter, solutions to two major problems in petroleum exploration, fault and stratigraphic boundary pattern recognition, will be proposed. In contrast to the previous chapters of this dissertation, the application is no longer texture analysis. However, parts of the approaches are related to texture approaches.

In Section 8.1, an introduction to petroleum exploration with reflection seismics will be given. In Sections 8.2 and 8.3, solutions to the problems of fault and stratigraphic boundary interpretation are proposed and discussed. Finally, in Section 8.4, the chapter is summarized.

8.1 Introduction to petroleum exploration with seismic

Denham [27] gives the following introduction to seismic reflection exploration: "Seismic reflection exploration – the study of the earth using sound waves reflected from beneath the surface – is the dominant method in exploration geophysics. [...] Exploration geophysics is merely a tool we use to look at the geology of a part of the earth we can neither sample nor visit in person. The geological understanding which we build using geophysics and other geological tools is itself only a tool used to detect a probable oil or gas field. Seismic interpretation is the interface between the exact [...] and inexact [...]. The earth is so complex that we never have a complete picture." In this section, a brief introduction to geology, petroleum exploration, seismic signals, and seismic signal processing will be given. For more extensive coverage of geology and seismics, several books and articles are available, e.g., [27, 54, 57, 119].

8.1.1 Geophysical background

Practically all occurrences of known oil and gas are in sedimentary rock. A sedimentary rock is formed by the accumulation, diagenesis, and solidification of organic material, mud, silt, sand, and other clastic materials. As new material is sedimented atop, pressure increase forms sedimentary rock bodies.

Provided adequate temperature, organic sediments starts forming oil and gas. These sedimentary layers are denoted *source rock*. The oil and gas may flow through permeable media. Since the density of petroleum is less than the density of water, it will be replaced by water if it is allowed to escape. If no sealing cap is present, it will escape to the surface of the earth. However, if there is a combination of a sealing cap and a porous medium, the petroleum may be trapped and accumulate in a *reservoir*. The quest for oil and gas is the search for such reservoirs in the ground. Typical porous media are sandstone and limestone while the most common caps are non-permeable sedimentary layers, e.g., shale, and structural seals, stemming from faults generated by tectonic or deformation forces.

Permeability is also a necessity for successful production, since reservoirs are produced by a few wells draining a very large volume (a petroleum reservoir may be several kilometers in extent). Some sediments are in nature permeable, but permeability may also stem from breaks in a sealing cap.

The focus of this chapter will be on automatic identification of *faults* and *stratigraphic* properties important for the flow (permeability) and sealing of petroleum. However, before the attention is focused on these problems, an introduction to seismic signals, signal gathering, preprocessing, and inversion will be given.

8.1.2 Seismic surveying

Seismic surveying is one of the major techniques used for constructing an image of the subsurface for the detection of oil and gas reservoirs. In a seismic survey, a source is emitting sound waves through the (sedimentary) layers of the earth. Any interface between layers of rock with different speeds of sound and densities will reflect some of the incoming sound pulse back to the surface. The sound source may be a dynamite explosion, a pneumatic air gun, a massive vibrating object, or similar. The recorders are typically geophones, measuring vibration in the earth, or hydrophones, measuring sound pulses in water. An illustration of a sea survey configuration with boat, sound source, and receivers is shown in Figure 8.1. Reflections with long reflection times (typically up to a few seconds) basically correspond to deep interfaces (up to a few thousand meters) or large reflection angles. The maximum resolution of the recorded seismic signal is in the range of beteween a meter and a tenth of a meter.

Traditionally, seismic surveys gave two-dimensional cross sections of the earth. However, most seismic surveys today provide three-dimensional mappings of the earth, e.g. by closely spaced two-dimensional cross sections. This provides for improved noise removal and a better understanding of the sediments. Another recent trend is 4D seismic, where multiple seismic surveys are conducted over time (fourth dimension) in order to monitor and map the drainage of the reservoirs.

Analysis tools and techniques with considerably better resolution are available for

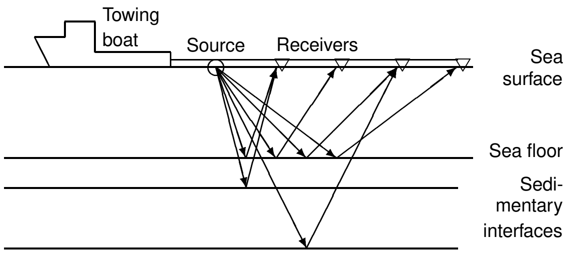


Figure 8.1: Principle of offshore seismic surveying.

wells. Some examples are sonic measurements, radioactive measurements, and core samples. However, wells are very costly to produce, typically in the range of ten million US dollars per well. Consequently, the number, and therefore the density, of wells is very low, especially during exploration. In practice, seismic and well measurements are the only applied techniques for imaging the subsurface in petroleum exploration.

8.1.3 Seismic processing

The processing steps leading up to an image of the sub-surface may be divided in two parts; 1) seismic processing for data enhancement and 2) seismic inversion. In this subsection, a few of the most important enhancement techniques are briefly reviewed. Inversion is treated briefly in the next subsection.

The ideal signal source would be a source emitting a Diracs delta pulse into the earth, i.e., a pulse with infinitesimal duration and infinite amplitude. However, this is not practically attainable and the submitted waveform is only approximating the Diracs delta pulse. Furthermore, the source is not 100% repeatable. To compensate for these problems, the recorded seismic reflection signals are deconvolved with the recorded source waveform.

The transmission medium, the earth, is a similar distortion source. Of particular interest is the fact that high frequencies are damped, resulting in blur and reduced resolution. This is particularly a problem for reflections with long travel times. To compensate for this problem, the seismic signal may be deconvolved with a modeled unit pulse response of the transmission medium.

Yet another problem with the data quality is the low signal to noise ratio of the signal due to several noise sources. If the noise is random and data independent, averaging multiple measurements of the same property will yield reduced noise. This is utilized in a technique denoted stacking, where multiple measurements from the same reflection point are averaged.

A multitude of other data quality improvement techniques with various tradeoffs also exist. A more complete presentation is given by Kearey [57].

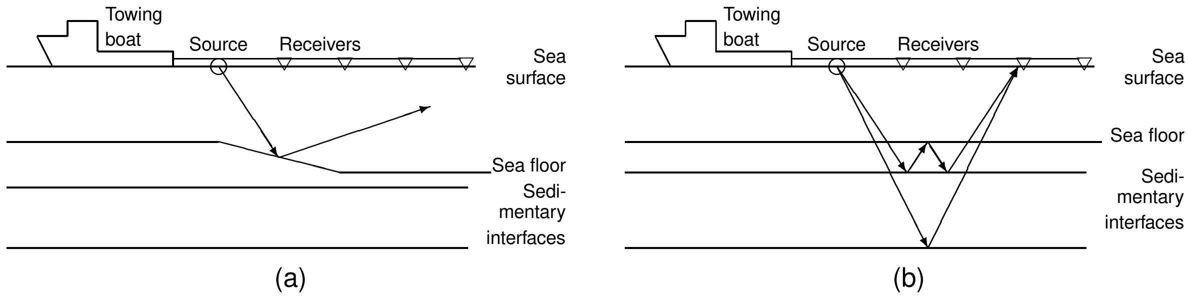


Figure 8.2: (a) Illustration of problem with dipping layers. (b) Multiple reflection paths yielding the same arrival time.

8.1.4 Seismic inversion

If the internal structure and physical properties of the earth were known, the magnitude of the recorded seismic measurement could be predicted accurately. However, in seismic data analysis, the problem is the converse, namely to deduce some aspects of the earth's internal structure on the basis of the measurements. This is a problem with a non-unique solution since many different geological configurations could produce the observed measurements.

The basic survey model, presented in Section 8.1.2, was developed with an assumption of horizontal reflectors. Since the angle of incidence is equal to the reflection angle, the reflection point is midway between the source and the receiver if the reflectors are horizontal. For dipping reflectors, however, this simple relation is no longer valid, see Figure 8.2(a), and without knowledge of the dip, the reflection point is unknown. The solution is to carefully examine the data to determine the most probable dip angles of the major reflectors and compensate for the dip by a process called *migration*.

Travel times and amplitudes are the only measured properties of the seismic signal. The travel time is a function of the sound velocity in the transmission medium and the length of the signal path. However, there are many possible paths from source to receiver having the same travel time. Of particular interest is multiples – reflections "bouncing" between multiple reflectors. An example sketch of multiple reflections is illustrated in Figure 8.2(b). The effect of multiple reflectors may be that the multiples from strong reflectors completely drown the information from weaker reflectors. In order to remove multiple reflections, the first arrivals are analyzed. Then a probable multiple reflection path is modeled and the modeled multiple response is subtracted from the measured seismic response.

The discussion given here is not a complete analysis of all inversion techniques. Compensation techniques for problems as refractions, diffractions, etc. also exist, see [27, 54, 57, 119].

8.2 Fault identification

At a large scale, the initial sedimentation pattern was horizontal, but deformation by folding and faults, as illustrated in Figure 8.3, might make structural traps. These fold-

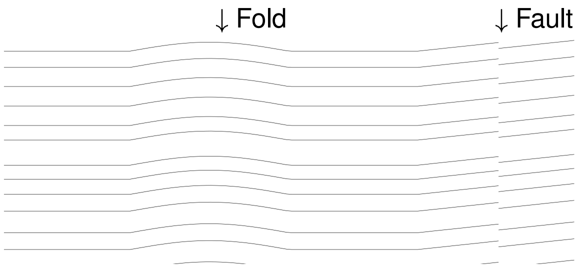


Figure 8.3: Illustration of principle behind two types of structural traps, fold and fault.

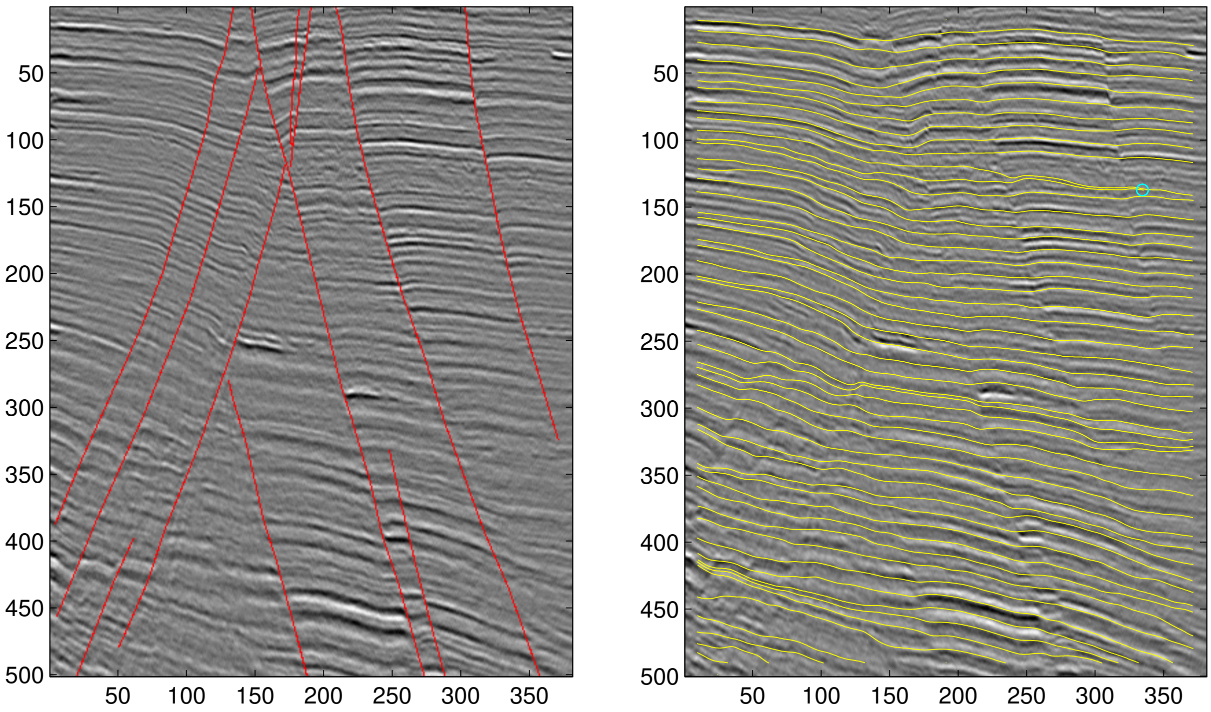
ings and faults are formed by tectonic forces, i.e., by movements from deep within the earth's crust (e.g. earthquakes), flow of salt or shale[1] from deeper layers, or differential compaction. A fault may generate a fluid channel through a sealing layer or seal a permeable layer and is therefore of significant importance in the quest for oil and gas. It will typically be represented by an abrupt change in the seismic response and may therefore be detected by methods for detection of abrupt changes. The final goal of fault analysis is an interpretation as shown in Figure 8.4(a).

A technique for fault analysis was presented by Alam and Caragounis [3], where the seismic horizons were auto-traced and faults identified by discontinuities in the extracted horizons. However, by experience, auto-traced horizons are not very reliable. Consequently, the fault detection may be biased by errors in the horizon identification. Bahorich and Farmer [7] recently presented a patented [8] technique for fault feature extraction (highlighting) by local coherence analysis. The coherence method is based on the idea that two adjacent traces[2] separated by a fault are poorly correlated, using attributes from the cross correlation as fault features. However, the windowing in the cross correlation causes a smeared response. A third approach to fault enhancement has been presented by Schwab et al. [117], in where an optimized horizontal linear prediction error filter is applied to the seismic. This filter will yield high amplitude at discontinuities, e.g. faults. Techniques for adapting the predictor to the underlying orientation of the seismic reflectors have not been presented, thus making it very sensitive to varying dip of the reflectors.

A fault feature extraction technique based on the orientation estimate is developed and tested in Subsection 8.2.1. A technique for transforming the fault features into a fault interpretation is required and a discussion of methods is given in Section 8.2.2.

[1]Salt is heavier but less compressible than most other sedimentary materials. As pressure increases, other materials are compressed more than the salt and their densities increase. This make them heavier than salt and the salt starts to prograde towards the surface. A similar situation may occur with water filled porous shale or clay where the water is not allowed to escape during the pressure increase.

[2]A trace is the term used for one column of seismic samples in the seismic volume, i.e., a vertical one-dimensional signal.



(a) (b)

Figure 8.4: (a) Example seismic cross section with manual fault interpretation. (b) Illustration of the local orientation the cross section.

8.2.1 Fault feature extraction

If a fault intersects a strong seismic reflector, the reflector typically becomes discontinuous and is characterized by an abrupt change in the seismic signal. A convolution mask (filter) like

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \tag{8.1}$$

will intuitively enhance abrupt changes in seismic sections, while suppress smooth regions (the average is zero) horizontally. This filter, known as the Prewitt [48] filter, is often used for enhancement of vertical edges in image processing. A similar version exists for enhancement of horizontal edges. An alternative filter is the Sobel [48] filter,

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}. \tag{8.2}$$

Both masks perform approximate partial derivative estimation.

An edge in image processing is the transition from a bright region to a dark region, or vice versa. For any such vertical edge, any of these masks will have either a large positive or a large negative value. For a fault, on the other hand, the arrangement of bright and dark samples is more complex and the response may change from large positive to large negative in a small neighborhood. However, compared to non-faulted regions, the *magnitude* of the response is overall large.

As we can see from the seismic section of Figure 8.4(a), we do not always have vertical fault planes intersecting horizontal seismic reflection patterns. Since the reflectors have similar characteristics as edges, any edge enhancement mask having some component perpendicular to the reflectors will also enhance the *reflectors*. Hence, to enhance faults and not reflectors, the edge enhancement mask must be tuned to be parallel to the reflection layers. The layer orientation may be estimated using the same technique as proposed by Kass and Wittkin [56] and improved by Rao and Schunck [113]. This approach basically used vertical and horizontal partial derivative of Gaussian filters for orientation estimation, followed by directional smoothing. An illustration of the estimated directionality of the seismic section in Figure 8.4(a) is shown in Figure 8.4(b).

Rotating a discrete filter mask by an arbitrary angle is not straightforward. To accomplish the rotation, a linearly interpolated continuous representation may be modeled, rotated, and re-sampled. One local orientation value is estimated for each pixel and the edge mask should be tuned accordingly. That is, the edge mask should adapt to the local orientation. However, the generation of a new mask for each pixel is computationally complex. A faster approach is to generate a set of masks for a range of orientations off-line, store them in a look-up table, and apply the mask tuned to the closest orientation. However, an even more computationally efficient approach is to use a separable edge filter. In a separable implementation, the edge mask tuned to the angle θ may be given by $\cos(\theta)h_0 + \sin(\theta)h_{\frac{\pi}{2}}$, where h_α is the edge mask tuned to the angle α. By this approach, only two separable filters are necessary in the feature extraction stage.

All of these approaches were evaluated, with only insignificant differences in feature quality. A result from the separable approach is illustrated in Figure 8.5(a). The perceived quality or clarity of the extracted features can be improved by some kind of smoothing of the features. Figure 8.5(b) shows the image of Figure 8.5(a) smoothed by a two-dimensional Gaussian low-pass filter with $\sigma_s = 3$ along the fault direction and $\sigma_s = 0.5$ perpendicular.

Any of the images of Figure 8.5 are highlighting the faults significantly better than the original seismic image. Hence, they may aid manual fault identification, which is a major and difficult task. Furthermore, as we will see in the next subsection, the features may also be used for automatic or semi-automatic fault identification.

Some other approaches to fault feature extraction have also been attempted, with less success. Approaches using appropriately tuned Gabor filters, Section 2.2.3, other band-pass filters derived from the QMF banks of Section 3.2, the optimal texture filters[3] of Section 5.3, and the phase difference highlighting technique by Bovik et al. [13] have been attempted. However, all the results were worse than the results with the edge enhancement filters. Some of the experiments are reported in [98]. Application of edge

[3]With optimal texture filters, the fault was modeled as an instantaneous texture.

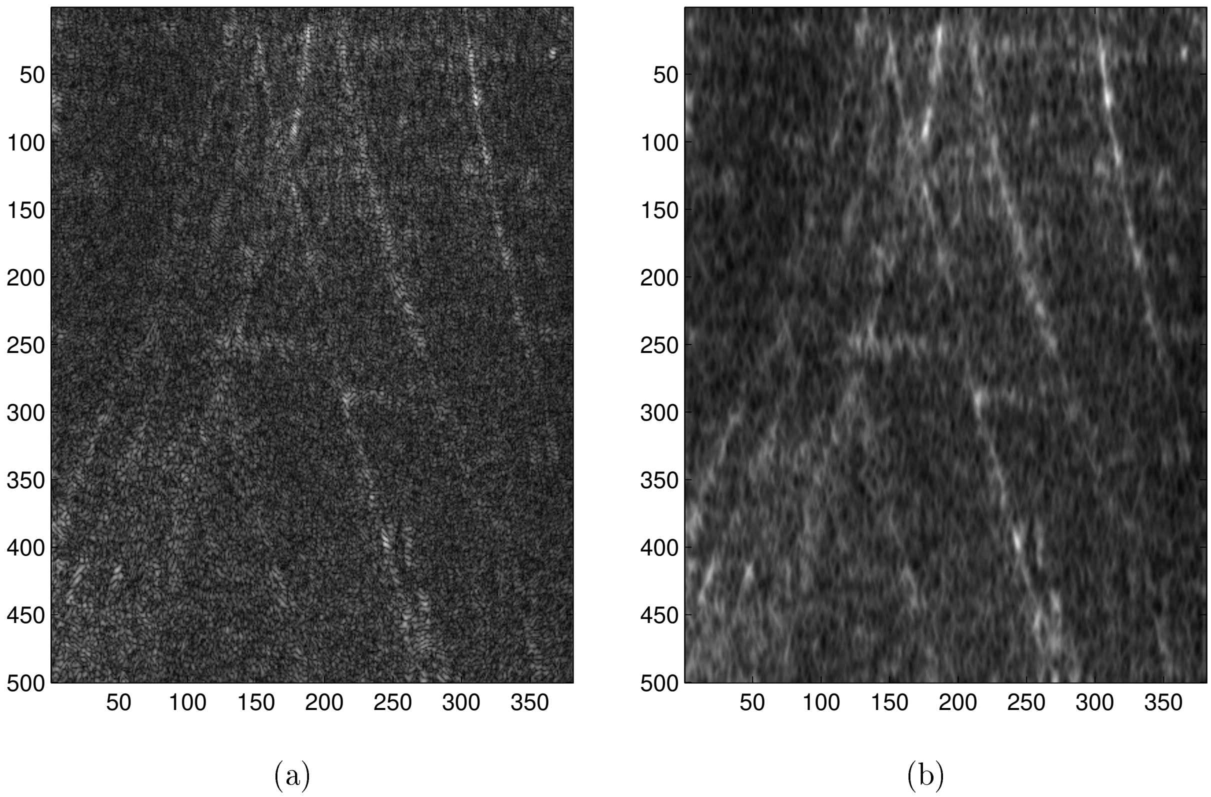
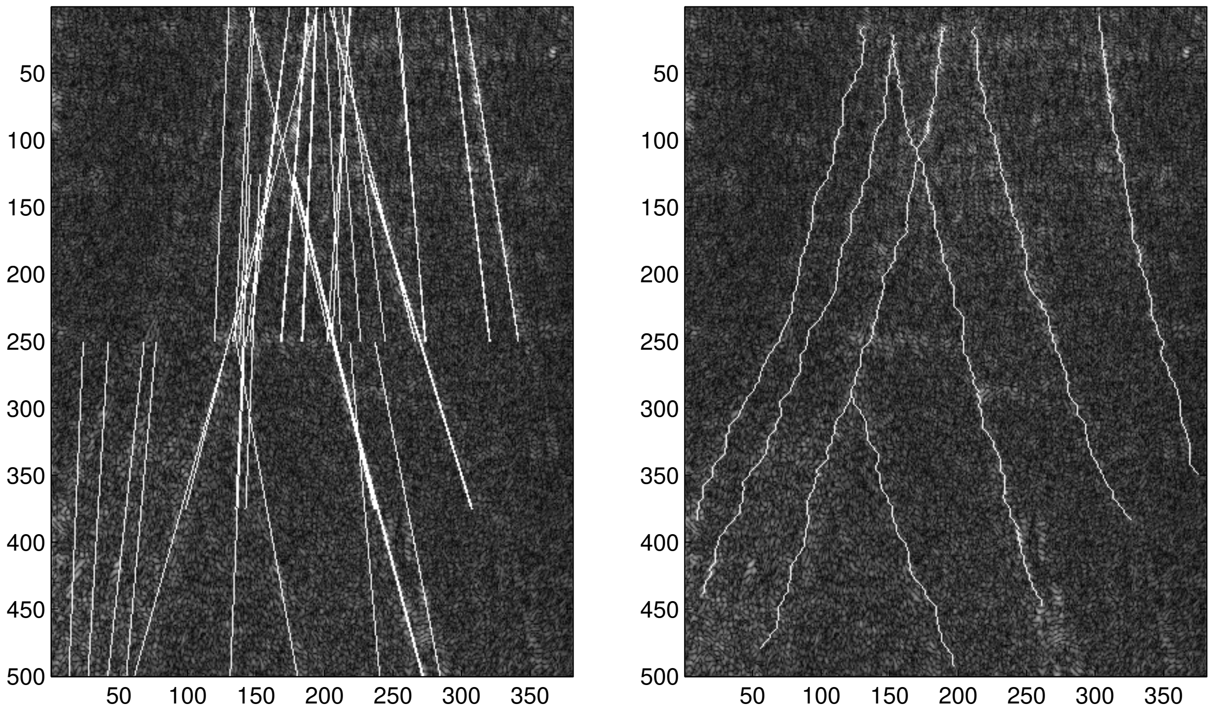


Figure 8.5: (a) Orientation adaptive *separable* Sobel mask response magnitude for the seismic cross section of Figure 8.4 and (b) a directionally smoothed version.

enhancement techniques without strong orientation properties, like the Laplacian [48] and SUSAN [121], have also been evaluated. However, due to strong edge characteristics of the seismic *reflectors*, all applied non-oriented techniques tended to emphasize the reflectors much more than the faults.

8.2.2 Fault surface identification from the extracted features

In order to have an automatic fault identification system, some approach for converting the fault features into fault surfaces in 3D or fault lines in 2D is needed. The ideal situation is to have a fully automatic system, transforming the fault features into fault surfaces with no user interaction. If it is assumed that the faults may be represented as lines of linked straight segments in 2D cross-sections, the Hough transform [39, 45] may be used to detect the segments. A significant effort was done to utilize the Hough transform technique for this approach, using the technique from [10]. The best result was as shown in Figure 8.6(a). For implementational reasons, the image was divided into overlapping blocks, and one line segment detected per block. This causes some false detections that may easily be removed. However, we still see that multiple faults or fault segments are missed by the approach.

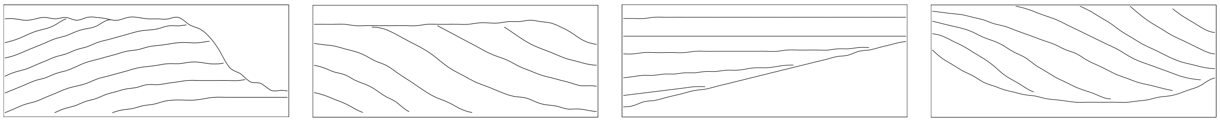


(a) (b)

Figure 8.6: Interpreted faults from the image of Figure 8.5(a). (a) Fully automatic identification using the Hough transform. (b) Semi-automatic identification using minimum cost path.

A more successful alternative may be to do semi-automatic detection of the faults, i.e., detection requiring some user interaction. One way of converting the fault features into fault lines or fault surfaces is to let the user select seed points that the system uses for tracing the faults. Verwer et al. [134] suggest a technique for using linear programming to find a minimum cost path between two seed points. Results using this approach are shown in Figure 8.6(b). As we can see, all the major faults were traced successfully, although the lines are still somewhat jagged.

The major effort in manual fault identification is not the interpretation of a single seismic cross section, but the interpretation of hundreds of cross sections. Furthermore, it is reasonable to assume that the fault lines in neighboring sections are fairly close in position, shape, and orientation. Hence, another approach to semi-automatic identification is to use deformable template matching. This approach was taken by Østebø [82], but some further adjustments are required to make this approach robust enough.



(a) (b) (c) (d)

Figure 8.7: Some important stratigraphic boundaries: (a) Erosional truncation, upper boundary formed by erosion. (b) Toplap, upper boundary formed mainly as a result of non-deposition. (c) Onlap and (d) downlap, lower boundaries formed by sedimentation on initially inclined surfaces.

8.3 Stratigraphic boundary identification

Most oil and gas discovered has been found in structural or mainly structural traps [27] and the undiscovered potential in such traps is probably decreasing. Another kind of traps, stratigraphic traps, is now receiving increased attention. A stratum is "a tabular or sheet-like mass, or a single and distinct layer, or [...] sedimentary material [...], visually separable from other layers above and below by a discrete change in character [...] or by a sharp physical break in deposition, or by both." [72, 74]. Stratigraphic analysis [85] is the analysis of strata and their boundaries. The analysis of stratigraphic traps is generally more difficult than structural trap analysis. Hence, it may be expected that the proportion of undiscovered stratigraphically trapped reservoirs is larger than for structural trapped reservoirs.

A stratigraphic trap occurs when the reservoir rock is surrounded by an impermeable seal and the depository layers did not have a horizontal structure. A river delta is a typical example of a non-horizontal deposition region. Crucial in the detection of stratigraphic traps is the identification of boundaries between different strata. An important class of boundaries are boundaries where one stratum *terminates* into another stratum [85, 75], see Figure 8.7. Seismic stratigraphic termination analysis is the analysis of termination patterns in the seismic signal. The straightforward approach to termination analysis is to do manual or automatic interpretation of all layers and analyze their terminations.

No explicit works on automated termination detection have been found, but implicit stratigraphic feature detection is reported with the coherence cube approach [7]. In this section, an approach to automated termination detection is proposed.

As with the fault feature extraction, the approach by Rao and Schunck [113] is utilized for orientation estimation. Given the local orientation estimate, *flow lines* following the local orientation are constructed. Starting at any position, the flow line is the curve through that position having the orientation of the local angle estimates in its path. The technique was applied to the image in Figure 8.8(a). In an illustrative experiment, flow lines sparsely covering the image were generated, Figure 8.8(b), and we can clearly see how the flow lines follow the orientation and indicate regions with terminations. A slightly different way of utilizing the method was applied in the experiments leading to Figure 8.8(c). In this experiment, the density of flow lines was increased significantly. When two flow lines (in this experiment; when two flow lines with seeds in the same column in the image)

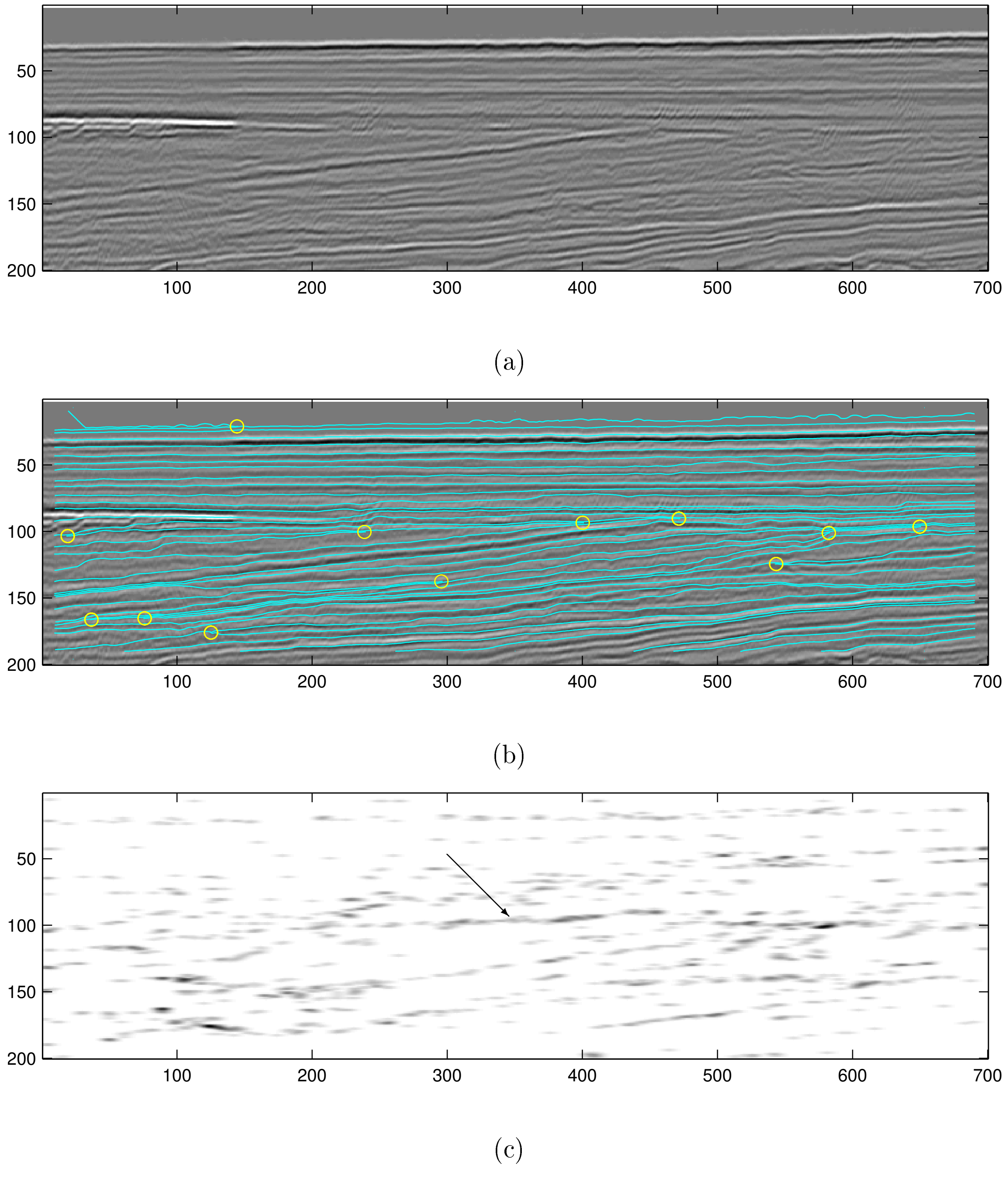


Figure 8.8: (a) Example seismic cross-section and (b) flow lines overlaid the seismic section, indicating the dominating local orientations. Terminating flow lines are marked by small circles. (c) Termination density, the arrow pinpoints an interface which is successfully highlighted by the technique.

converges to the same pixel position, that position is marked as a termination. For each termination, the value in an accumulator array with each cell corresponding to one pixel was increased by one and this accumulator array was smoothed and scaled appropriately for display. In this illustration, we can see the sequence boundaries of the different layers highlighted by black, e.g., around the pinpointed interface (about 100 samples from the top).

In the implementation used in the experiments, the step size in the flow line tracking was always one and the flow lines were represented as connections of line segments. The orientation of each line segment was taken to be the orientation in the sample position closest to the start of the line segment.

In summary, with the flow line technique proposed in this subsection, a solution to the problem of seismic stratigraphic termination detection is proposed. As we see, the technique is suited to the detection of terminating strata in the seismic images.

8.4 Summary

In this chapter, a review of the basic principles of geology, seismic data acquisition, and seismic data processing was given. Then the problem of fault identification was introduced. An approach to fault feature extraction using the edge enhancement filter masks tuned by the orientation of the seismic layers was proposed. It was shown that these features were powerful for fault highlighting. The fault analysis part was concluded with discussion and experiments on making a meaningful geometric description of the fault surface based on the extracted features.

Next, the problem of detection of terminating strata from seismic signals was introduced. A technique for termination detection based on local orientation estimates was suggested. In this approach, the local orientation was traced by flow lines. Flow line intersections correspond to terminating strata.

All the techniques for seismic analysis presented herein are founded on proven signal processing theory. No explicit use of horizon interpretation and other higher level seismic interpretation is made. This is expected to yield a system that degrades gracefully with the quality of the data and which is not biased by the interpretation.

For implementational reasons, all experiments, and therefore all discussion given in this chapter, have been made with the assumption that the input is a two-dimensional seismic cross section. Since the seismic data is inherently three-dimensional, this is a limitation to the power of the techniques. However, all techniques presented may readily be extended to three dimensions.

Chapter 9

Summary and conclusions

9.1 Major contributions of this work

The major contributions of this work are listed below.

1. In Section 3.2 a new class of filter banks, QMF filter banks, is proposed for texture feature extraction. Furthermore, a framework for feature extraction with multi rate (sub-sampled) filter banks is proposed. It is seen in several experiments that the features derived from the QMF filters yield good segmentation quality.

2. A new approach to texture filter design based on linear prediction error filters is proposed in Chapter 4. The approach is optimal with respect to representation. Hence, it is not optimal with respect to discrimination. Most texture images are successfully discriminated by the approach, but the method is unsuccessful for some images. The results are more or less of the same quality as with several non-optimized filter bank approaches, but with a considerably lower computational complexity.

3. In Section 5.2 and Appendix A, a new model for estimating the mean and variance of texture features extracted with linear FIR filters and local energy functions is developed. The model is applicable for analyzing the performance of a given filter or for filter optimization.

4. Multiple new two-texture optimal one-filter design approaches are presented in Section 5.3. Exact closed form optimization to mean feature value criteria and approximate closed form solutions to mean and variance criteria, in particular the Fisher criterion, are developed. Experiments in Section 5.3 and in Chapter 6 indicate that these new techniques are very good.

5. The mean value optimization approaches are extended to multiple-texture cases in Section 5.4. Experiments show very good results compared to other optimal and non-optimal filtering approaches. Advantages with the two- and multiple-texture optimization approaches are that they yield features that may be extracted with relatively low computational complexity. Furthermore, these features may be classified with low complexity.

6. The new methods proposed here along with methods broadly covering most categories of filters and filter banks used for texture feature extraction are evaluated in Chapter 6. In the experiments, the focus is put on making the results as comparable as possible. Model based and statistical features are also included in the test for reference.

7. Experiments on document image segmentation and content based image queries are reported in Chapter 7. In these experiments, it is shown that the decimated version of the QMF filter bank of Section 3.2 is a powerful texture feature extractor for real world images.

8. In Chapter 8, a method for estimating the local orientation in seismic data is discussed. A method for utilizing this estimate for detecting patterns corresponding to terminating sedimentary layers in seismic data is proposed. Furthermore, an approach to fault feature extraction in seismic signals based on the orientation estimate is proposed and tested. Finally, methods for interpreting fault planes from the fault features are discussed.

9.2 Major conclusions from this work

Several conclusions may be drawn from the experiments reported in this dissertation. Below is a summary of the major conclusions:

1. It may from the experiments in Chapters 3 and 6 be concluded that the dyadic decomposition in most cases is not optimal for texture feature extraction with filter banks. This supports similar findings reported by Chang and Kuo [16].

2. Numerous works have concentrated on tuning Gabor filters or filters derived from the wavelet transform for texture recognition. In this work, other filter bases have been shown to yield comparable or better results. Hence, restricting the filter basis to the Gabor or wavelet derived filters will limit the potential of the filtering.

3. By applying sub-sampled perfect reconstruction filter banks, good segmentation performance can be achieved with significant savings in computational complexity. However, with sub-sampling, some increase in edge inaccuracy is observed.

4. Several filter optimization approaches are evaluated and optimal filter designs do in many cases yield features that are superior compared to non-optimized filters. Furthermore considerable reductions in computational complexity are obtained by many optimal filter approaches.

9.3 Suggestions for further research

1. Due to mathematical constraints, a few assumptions and approximations had to be made to obtain the closed form optimal solutions of Chapter 5. These were Gaussian assumption, in some cases restrictions to criterion functions incorporating

only the feature mean, etc. Further effort should be put on working around these assumptions and approximations.

2. A few attempts to filter optimization with respect to the Fisher criterion have been made in Sections 5.3.2 and 5.3.3. Approximate closed form, gradient search, simulated annealing, annealing evolution, and exact iterative solutions have been proposed. Other iterative or closed form solutions should also be targeted. Furthermore, approaches based on the alternate expressions and derivations of Appendix A.4 should be examined further.

3. The ultimate criterion in texture segmentation is the classification error rate. All optimization approaches in this dissertation have been with respect to criteria that are only indirectly related to the error rate. The optimal solutions are consequently not optimal with respect to the error rate. The only reported approaches on optimization with respect to the error rate are some Gabor filter design approaches [31, 30, 138, 139] and a neural net approach [51]. The first approach is restricted in the class of filters, while the second is limited by the lack of a global optimum and closed form solution. Further effort should be put on minimum error optimization.

4. The multi-texture optimization strategy developed in Section 5.4 is based on forming pairs of textures and designing filters for these pairs. Alternate optimal filter bank design schemes should be developed, to see if even better results may be obtained.

5. The section on unsupervised optimal filter design, Section 5.5, did show good results. However, more effort should be put on testing the approach and developing the block clustering scheme further.

6. The experiments on document segmentation and content based image search were focused on the texture feature extraction. Several issues of pre- and post processing remain unresolved. The features proposed here are very advantageous with respect to computational complexity. More effort should be made in combining these features with proper pre- and post-processing systems.

7. The seismic interpretation approaches presented in Sections 8.2 and 8.3 were limited to two dimensions. For most of the techniques, extensions to multiple dimensions are readily available with only a moderate increase in complexity (e.g. most filtering approaches were separable). Extensions to multiple dimensions should be developed and tested.

Appendix A

Mathematical details

In this appendix, the mathematical details for some of the derivations presented earlier in this dissertation are given. In Appendix A.1 a validation of the existence of the eigensolutions found earlier is given, in Appendix A.2 the details of the feature variance model are developed, and in Appendix A.3 an approximation of the variance expression useful in the closed form solutions is developed. Finally in Appendix A.4 a compact representation of the variance model is discussed along with an approach to find an upper bound of the Fisher criterion and some ideas for future optimization research.

A.1 Validation of existence of optimal solutions

Are we guaranteed that all solutions to the eigenproblems, Equations (5.23), (5.28), and (5.31), are valid solutions? That is, are we guaranteed that the solution to

$$\left(\mathbf{R}_{x_2x_2}{}^{-1}\mathbf{R}_{x_1x_1}\right)\mathbf{h} = \lambda\mathbf{h} \tag{A.1}$$

will yield a solution vector $\mathbf{h}$ such that

$$\frac{(\mathbf{h}^T\mathbf{R}_{x_1x_1}\mathbf{h})}{(\mathbf{h}^T\mathbf{R}_{x_2x_2}\mathbf{h})} = \lambda, \tag{A.2}$$

where λ is the corresponding eigenvalue? Since $(\mathbf{h}^T\mathbf{R}_{x_2x_2}\mathbf{h})$ is a scalar, λ is unmodified by dividing and multiplying by $(\mathbf{h}^T\mathbf{R}_{x_2x_2}\mathbf{h})$, i.e.,

$$\lambda = \frac{(\mathbf{h}^T\mathbf{R}_{x_2x_2}\mathbf{h})\lambda}{(\mathbf{h}^T\mathbf{R}_{x_2x_2}\mathbf{h})} = \frac{\mathbf{h}^T\mathbf{R}_{x_2x_2}\lambda\mathbf{h}}{(\mathbf{h}^T\mathbf{R}_{x_2x_2}\mathbf{h})}. \tag{A.3}$$

Combining this with Equation (A.1), we get

$$\lambda = \frac{\mathbf{h}^T\mathbf{R}_{x_2x_2}\left(\mathbf{R}_{x_2x_2}{}^{-1}\mathbf{R}_{x_1x_1}\right)\mathbf{h}}{(\mathbf{h}^T\mathbf{R}_{x_2x_2}\mathbf{h})} = \frac{(\mathbf{h}^T\mathbf{R}_{x_1x_1}\mathbf{h})}{(\mathbf{h}^T\mathbf{R}_{x_2x_2}\mathbf{h})}, \tag{A.4}$$

which is identical to the requirement of Equation (A.2). Hence, all eigensolutions are valid solutions, provided that $\mathbf{R}_{x_2x_2}$ is non-singular and $(\mathbf{h}^T\mathbf{R}_{x_2x_2}\mathbf{h}) \neq 0$.

A.2 Variance expression

For the notations of this appendix, refer to Figure 5.1. The variance of the feature image, $v_i(m,n)$, is

$$\begin{aligned}\sigma_{v_i}^2 &= E\left\{(v_i(m,n)-\mu_{v_i})^2\right\} \\ &= E\left\{v_i^2(m,n)\right\} - 2E\left\{v_i(m,n)\right\}\mu_{v_i} + {\mu_{v_i}}^2 \\ &= E\left\{\mathbf{w}^T\mathbf{z}_i(m,n)\mathbf{z}_i^T(m,n)\mathbf{w}\right\} - {\mu_{v_i}}^2 \\ &= (\mathbf{w}^T\mathbf{R}_{z_iz_i}\mathbf{w}) - {\mu_{v_i}}^2,\end{aligned} \tag{A.5}$$

where

$$\mathbf{R}_{z_iz_i} = E\left\{\mathbf{z}_i(m,n)\mathbf{z}_i^T(m,n)\right\}. \tag{A.6}$$

The matrix $\mathbf{R}_{z_iz_i}$ is readily constructed from the autocorrelation function of the image $z_i(m,n)$, given by

$$\begin{aligned}r_{z_iz_i}(k,l) &= E\left\{z_i(m,n)z_i(m+k,n+l)\right\} \\ &= E\left\{y_i^2(m,n)y_i^2(m+k,n+l)\right\}.\end{aligned} \tag{A.7}$$

With this fourth order moment, no closed form solution for criteria incorporating the variance has yet been found. However, if we assume that $y_i(m,n)$ is Gaussian, we may write this as [83]

$$r_{z_iz_i}(k,l) = 2{c_{y_iy_i}}^2(k,l) + 4c_{y_iy_i}(k,l){\mu_{y_i}}^2 + {r_{y_iy_i}}^2(0,0), \tag{A.8}$$

where $c_{y_iy_i}(k,l)$ is the covariance function [83] of $y_i(m,n)$. Further

$$\begin{aligned}c_{y_iy_i}(k,l) &= E\left\{(y_i(m,n)-\mu_{y_i})\left(y_i(m+k,n+l)-\mu_{y_i}\right)\right\} \\ &= r_{y_iy_i}(k,l) - {\mu_{y_i}}^2,\end{aligned} \tag{A.9}$$

where $r_{y_iy_i}(k,l)$ is the autocorrelation function of $y_i(m,n)$. Hence, we have

$$\begin{aligned}r_{z_iz_i}(k,l) &= 2\left(r_{y_iy_i}(k,l) - {\mu_{y_i}}^2\right)^2 + 4\left(r_{y_iy_i}(k,l) - {\mu_{y_i}}^2\right){\mu_{y_i}}^2 + {r_{y_iy_i}}^2(0,0) \\ &= 2{r_{y_iy_i}}^2(k,l) - 2{\mu_{y_i}}^4 + {\mu_{v_i}}^2.\end{aligned} \tag{A.10}$$

Furthermore, we have

$$\begin{aligned}r_{y_iy_i}(k,l) &= E\left\{y_i(m,n)y_i(m+k,n+l)\right\} \\ &= E\left\{\mathbf{h}^T\mathbf{x}_i(m,n)\mathbf{x}_i^T(m+k,n+l)\mathbf{h}\right\} \\ &= (\mathbf{h}^T\mathbf{R}_{x_ix_i}(k,l)\mathbf{h}),\end{aligned} \tag{A.11}$$

where

$$\mathbf{R}_{x_ix_i}(k,l) = E\left\{\mathbf{x}_i(m,n)\mathbf{x}_i^T(m+k,n+l)\right\}. \tag{A.12}$$

Note that from Equation (5.13), we have $\mathbf{R}_{x_ix_i} = \mathbf{R}_{x_ix_i}(0,0)$. Combining Equations (A.5), (A.6), (A.10), and (A.11) we are able to estimate the variance of the extracted features, $\sigma^2_{v_i}$. This estimate is based on the autocorrelation function of the texture, $r_{x_ix_i}(m,n)$, the filter, $h(m,n)$, and the smoothing filter $w(m,n)$. Summarizing the derivations of this section, we may write the expression for $\sigma^2_{v_i}$ explicitly as

$$\begin{aligned}\sigma^2_{v_i} &= (\mathbf{w}^T\mathbf{R}_{z_iz_i}\mathbf{w}) - {\mu_{v_i}}^2 \\ &= \sum_{k_1=0}^{M_w-1}\sum_{l_1=0}^{N_w-1} w(k_1,l_1)\sum_{k_2=0}^{M_w-1}\sum_{l_2=0}^{N_w-1} w(k_2,l_2) r_{z_iz_i}(k_1-k_2,l_1-l_2) - {\mu_{v_i}}^2\end{aligned} \tag{A.13}$$

and

$$\sigma^2_{v_i} = \sum_{k_1=0}^{M_w-1}\sum_{l_1=0}^{N_w-1} w(k_1,l_1)\sum_{k_2=0}^{M_w-1}\sum_{l_2=0}^{N_w-1} w(k_2,l_2) 2{r_{y_iy_i}}^2(k_1-k_2,l_1-l_2) - 2{\mu_{y_i}}^4 + {\mu_{v_i}}^2 - {\mu_{v_i}}^2 \tag{A.14}$$

yielding

$$\sigma^2_{v_i} = 2\sum_{k_1=0}^{M_w-1}\sum_{l_1=0}^{N_w-1} w(k_1,l_1)\sum_{k_2=0}^{M_w-1}\sum_{l_2=0}^{N_w-1} w(k_2,l_2) (\mathbf{h}^T\mathbf{R}_{x_ix_i}(k_1-k_2,l_1-l_2)\mathbf{h})^2 - 2\left(\mu_{x_i}\mathbf{1}^T\mathbf{h}\right)^4, \tag{A.15}$$

where $\mathbf{1} = [1\ 1 \cdots 1]^T$ and $w(m,n)$ is the smoothing filter, which is of size $M_w \times N_w$. A discussion of compact representations for this variance expression is found in Appendix A.4. Using the matrix/vector differentiation rules of Searle [118] we get the partial derivative

$$\begin{aligned}\frac{\partial\sigma^2_{v_i}}{\partial\mathbf{h}} = 8&\sum_{k_1=0}^{M_w-1}\sum_{l_1=0}^{N_w-1} w(k_1,l_1)\sum_{k_2=0}^{M_w-1}\sum_{l_2=0}^{N_w-1} w(k_2,l_2) (\mathbf{h}^T\mathbf{R}_{x_ix_i}(k,l)\mathbf{h})\left(\mathbf{R}_{x_ix_i}(k,l)\mathbf{h} + \mathbf{R}_{x_ix_i}(k,l)^T\mathbf{h}\right) \\ &- 8\left(\mu_{x_i}\mathbf{1}^T\mathbf{h}\right)^3\mu_{x_i}\mathbf{1}.\end{aligned} \tag{A.16}$$

A.3 Approximate variance expression

From Appendix A.2, Equation (A.13), we know that it is possible to write the variance as

$$\sigma^2_{v_i} = \sum_{k_1=0}^{M_w-1}\sum_{l_1=0}^{N_w-1} w(k_1,l_1)\sum_{k_2=0}^{M_w-1}\sum_{l_2=0}^{N_w-1} w(k_2,l_2) r_{z_iz_i}(k_1-k_2,l_1-l_2) - {\mu_{v_i}}^2, \tag{A.17}$$

but as described in Section 5.3.2, this is not found to yield a closed form solution. However, if we approximate $r_{z_iz_i}(k,l)$ by $\hat{r}_{z_iz_i}(k,l)$, where

$$\hat{r}_{z_iz_i}(k,l) = \rho_{i1}\hat{r}_{z_iz_i}(k-1,l) \quad \forall\ k > 0 \tag{A.18}$$

and

$$\hat{r}_{z_iz_i}(k,l) = \rho_{i2}\hat{r}_{z_iz_i}(k,l-1) \;\; \forall\; l > 0, \tag{A.19}$$

where ρ_{i1} and ρ_{i2} are scalar parameters and $\hat{r}_{z_iz_i}(0,0) = r_{z_iz_i}(0,0)$, we may write

$$r_{z_iz_i}(k,l) \approx \hat{r}_{z_iz_i}(k,l) = \rho_{i1}^{|k|}\rho_{i2}^{|l|} r_{z_iz_i}(0,0). \tag{A.20}$$

Note that the approximations in Equation (A.18) and (A.19) correspond to modeling the image as separable and autoregressive (AR) [83] of order one.[1] With this approximation, we may write

$$\begin{aligned}
\sigma_{v_i}^2 &\approx \sum_{k_1=0}^{M_w-1}\sum_{l_1=0}^{N_w-1} w(k_1,l_1) \sum_{k_2=0}^{M_w-1}\sum_{l_2=0}^{N_w-1} w(k_2,l_2)\rho_{i1}^{|k_1-k_2|}\rho_{i2}^{|l_1-l_2|} r_{z_iz_i}(0,0) - {\mu_{v_i}}^2 \\
&= r_{z_iz_i}(0,0) \sum_{k_1=0}^{M_w-1}\sum_{l_1=0}^{N_w-1} w(k_1,l_1) \sum_{k_2=0}^{M_w-1}\sum_{l_2=0}^{N_w-1} w(k_2,l_2)\rho_{i1}^{|k_1-k_2|}\rho_{i2}^{|l_1-l_2|} - {\mu_{v_i}}^2.
\end{aligned} \tag{A.21}$$

To keep expressions manageable the following is defined

$$\rho_i = \sum_{k_1=0}^{M_w-1}\sum_{l_1=0}^{N_w-1} w(k_1,l_1) \sum_{k_2=0}^{M_w-1}\sum_{l_2=0}^{N_w-1} w(k_2,l_2)\rho_{i1}^{|k_1-k_2|}\rho_{i2}^{|l_1-l_2|}, \tag{A.22}$$

thus $\sigma_{v_i}^2 \approx \rho_i r_{z_iz_i}(0,0) - {\mu_{v_i}}^2$. Applying (A.10) and assuming zero mean input, we get

$$\begin{aligned}
\sigma_{v_i}^2 &\approx 2\rho_i {r_{y_iy_i}}^2(0,0) + {r_{y_iy_i}}^2(0,0) - {\mu_{v_i}}^2 \\
&= 2\rho_i(\mathbf{h}^T\mathbf{R}_{x_ix_i}\mathbf{h})^2 + (\mathbf{h}^T\mathbf{R}_{x_ix_i}\mathbf{h})^2 - (\mathbf{h}^T\mathbf{R}_{x_ix_i}\mathbf{h})^2 \\
&= 2\rho_i(\mathbf{h}^T\mathbf{R}_{x_ix_i}\mathbf{h})^2.
\end{aligned} \tag{A.23}$$

To find an optimal solution, we need the partial derivative of this approximation of $\sigma_{v_i}^2$ with respect to $\mathbf{h}$. As we see, the approximation of $\sigma_{v_i}^2$ consists of two elements, $(\mathbf{h}^T\mathbf{R}_{x_ix_i}\mathbf{h})^2$ and ρ_i, that are both functions of $\mathbf{h}$. The relative perturbations of $(\mathbf{h}^T\mathbf{R}_{x_ix_i}\mathbf{h})$ and ρ_i as functions of small perturbations of $\mathbf{h}$ were tested. The ρ_i's were determined to yield minimum deviation between Equations (A.15) and (A.23). These tests indicated that the relative perturbations of $(\mathbf{h}^T\mathbf{R}_{x_ix_i}\mathbf{h})$ were generally much larger than the relative perturbations of ρ_i. Hence, approximating ρ_i as constant will only perturbate the derivative of $\sigma_{v_i}^2$ minimally and we get the approximation

$$\frac{\partial\sigma_{v_i}^2}{\partial\mathbf{h}} \approx 8\rho_i(\mathbf{h}^T\mathbf{R}_{x_ix_i}\mathbf{h})\mathbf{R}_{x_ix_i}\mathbf{h}. \tag{A.24}$$

When substituted into $\partial J_F/\partial\mathbf{h} = 0$, these approximate expressions yield the approximate optimal closed form solution given in Section 5.3.2.

[1] Generally, making the approximations in Equation (A.20) makes the estimate of $r_{z_iz_i}(k,l)$ less accurate. The approximations are, however, justified by the useful results they give.

A.4 Reformulation of the feature extraction model

For realistically sized filters, the expression for $\sigma_{v_i}^2$, Equation (A.15), contains four large summations. In Appendix A.4.1 an alternate and more compact expression for the feature variance is developed. The development yields a technique for finding an upper bound on the Fisher criterion, detailed in Appendix A.4.2. In Appendix A.4.3, a discussion on the application of this approach for one-dimensional optimal filter design is given.

A.4.1 Alternate expressions for feature mean and variance

In the development of the feature variance estimator, Section A.2, it was assumed that

$$y_i(m,n) = h(m,n) * x_i(m,n), \tag{A.25}$$

where $*$ represents two-dimensional convolution. Furthermore

$$\begin{aligned} r_{y_iy_i}(k,l) &= E\left\{y_i(m,n)y_i(m+k,n+l)\right\} \\ &\approx \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} y_i(m,n)y_i(m+k,n+l) \\ &= y_i(k,l) * y_i(-k,-l) \\ &= h(k,l) * x_i(k,l) * h(-k,-l) * x(-k,-l) \\ &= \left(h\left(k,l\right) * h(-k,-l)\right) * \left(x_i(k,l) * x(-k,-l)\right) \\ &= r_{hh}(k,l) * r_{x_ix_i}(k,l), \end{aligned} \tag{A.26}$$

known as the Wiener-Lee theorem [65, page 85]. As has been illustrated many times, it is possible to write a 2D convolution as a vector multiplication. Consequently, the above expression may be rewritten

$$r_{y_iy_i}(k,l) = \mathbf{r}_{hh}{}^T\mathbf{r}_{x_ix_i}(k,l). \tag{A.27}$$

Combining this with Equations (5.12) and (A.11), we get

$$\mu_{v_i} = r_{y_iy_i}(0,0) = \mathbf{r}_{hh}{}^T\mathbf{r}_{x_ix_i}(0,0). \tag{A.28}$$

Using Equation (A.10) we get

$$\begin{aligned} r_{z_iz_i}(k,l) &= 2r_{y_iy_i}{}^2(k,l) - 2\mu_{y_i}{}^4 + \mu_{v_i}{}^2 \\ &= 2\left(\mathbf{r}_{hh}{}^T\mathbf{r}_{x_ix_i}(k,l)\right)^2 - 2\mu_{y_i}{}^4 + \mu_{v_i}{}^2 \\ &= 2\mathbf{r}_{hh}{}^T\mathbf{r}_{x_ix_i}(k,l)\mathbf{r}_{x_ix_i}{}^T(k,l)\mathbf{r}_{hh} - 2\mu_{y_i}{}^4 + \mu_{v_i}{}^2 \\ &= 2\mathbf{r}_{hh}{}^T\mathbf{R}_{r_ir_i}(k,l)\mathbf{r}_{hh} - 2\mu_{y_i}{}^4 + \mu_{v_i}{}^2. \end{aligned} \tag{A.29}$$

Hence, using Equation (A.17) and remembering that it is assumed that $\sum w(k,l) = 1$, we get

$$\begin{aligned}\sigma_{v_i}^2 &= \sum_{k_1=0}^{M_w-1}\sum_{l_1=0}^{N_w-1} w(k_1,l_1)\sum_{k_2=0}^{M_w-1}\sum_{l_2=0}^{N_w-1} w(k_2,l_2) r_{z_i z_i}(k_1-k_2,l_1-l_2) - {\mu_{v_i}}^2 \\ &= \sum_{k_1=0}^{M_w-1}\sum_{l_1=0}^{N_w-1} w(k_1,l_1)\sum_{k_2=0}^{M_w-1}\sum_{l_2=0}^{N_w-1} w(k_2,l_2) 2{\mathbf{r}_{hh}}^T \mathbf{R}_{r_i r_i}(k_1-k_2,l_1-l_2)\mathbf{r}_{hh} - 2{\mu_{y_i}}^4 + {\mu_{v_i}}^2 - {\mu_{v_i}}^2 \\ &= {\mathbf{r}_{hh}}^T \left(\sum_{k_1=0}^{M_w-1}\sum_{l_1=0}^{N_w-1} w(k_1,l_1)\sum_{k_2=0}^{M_w-1}\sum_{l_2=0}^{N_w-1} w(k_2,l_2) 2\mathbf{R}_{r_i r_i}(k_1-k_2,l_1-l_2)\right)\mathbf{r}_{hh} - 2{\mu_{y_i}}^4.\end{aligned} \tag{A.30}$$

The huge matrix summation in the expression above is purely input signal dependent. Hence, if we denote the matrix sum by $\mathbf{A}_{x_i}$, we may write

$$\sigma_{v_i}^2 = {\mathbf{r}_{hh}}^T \mathbf{A}_{x_i}\mathbf{r}_{hh} - 2{\mu_{y_i}}^4 \tag{A.31}$$

and if we assume zero mean input, we get

$$\sigma_{v_i}^2 = {\mathbf{r}_{hh}}^T \mathbf{A}_{x_i}\mathbf{r}_{hh}. \tag{A.32}$$

A.4.2 Upper bound on the Fisher criterion

The derivatives of the mean and the variance with respect to $\mathbf{r}_{hh}$ are

$$\frac{\partial \mu_{v_i}}{\partial \mathbf{r}_{hh}} = \mathbf{r}_{x_i x_i}(0,0) \tag{A.33}$$

and

$$\frac{\partial \sigma_{v_i}^2}{\partial \mathbf{r}_{hh}} = {\mathbf{A}_{x_i}}^T \mathbf{r}_{hh} + \mathbf{A}_{x_i}\mathbf{r}_{hh} = 2\mathbf{A}_{x_i}\mathbf{r}_{hh}, \tag{A.34}$$

respectively. Then the derivative of the Fisher criterion becomes

$$\begin{aligned}\frac{\partial J(\mathbf{r}_{hh})}{\partial \mathbf{r}_{hh}} &= \frac{2\left(\mu_{v_1}-\mu_{v_2}\right)\left(\frac{\partial \mu_{v_1}}{\partial \mathbf{r}_{hh}} - \frac{\partial \mu_{v_2}}{\partial \mathbf{r}_{hh}}\right)\left(\sigma_{v_1}^2+\sigma_{v_2}^2\right)}{\left(\sigma_{v_1}^2+\sigma_{v_2}^2\right)^2} - \frac{\left(\mu_{v_1}-\mu_{v_2}\right)^2\left(\frac{\partial \sigma_{v_1}^2}{\partial \mathbf{r}_{hh}} + \frac{\partial \sigma_{v_2}^2}{\partial \mathbf{r}_{hh}}\right)}{\left(\sigma_{v_1}^2+\sigma_{v_2}^2\right)^2} \\ &= \frac{2\left(\mu_{v_1}-\mu_{v_2}\right)\left(\mathbf{r}_{x_1x_1}(0,0)-\mathbf{r}_{x_2x_2}(0,0)\right)\left(\sigma_{v_1}^2+\sigma_{v_2}^2\right)}{\left(\sigma_{v_1}^2+\sigma_{v_2}^2\right)^2} \\ &\quad - \frac{\left(\mu_{v_1}-\mu_{v_2}\right)^2\left(2\mathbf{A}_{x_1}\mathbf{r}_{hh}+2\mathbf{A}_{x_2}\mathbf{r}_{hh}\right)}{\left(\sigma_{v_1}^2+\sigma_{v_2}^2\right)^2}.\end{aligned} \tag{A.35}$$

Equating to zero yields

$$\left(\mathbf{r}_{x_1x_1}(0,0)-\mathbf{r}_{x_2x_2}(0,0)\right)\left(\sigma_{v_1}^2+\sigma_{v_2}^2\right) = \left(\mu_{v_1}-\mu_{v_2}\right)\left(\mathbf{A}_{x_1}+\mathbf{A}_{x_2}\right)\mathbf{r}_{hh} \tag{A.36}$$

and consequently the optimal $\mathbf{r}_{hh}$ is

$$\mathbf{r}_{hh} = \frac{\sigma_{v_1}^2 + \sigma_{v_2}^2}{\mu_{v_1} - \mu_{v_2}} \left(\mathbf{A}_{x_1} + \mathbf{A}_{x_2}\right)^{-1} \left(\mathbf{r}_{x_1x_1}(0,0) - \mathbf{r}_{x_2x_2}(0,0)\right). \tag{A.37}$$

The factor $\left(\sigma_{v_1}^2 + \sigma_{v_2}^2\right) / \left(\mu_{v_1} - \mu_{v_2}\right)$ is a scalar and multiplying $\mathbf{r}_{hh}$ by a scalar is equivalent to multiplying $h(m,n)$ by a scalar. Since it may easily be shown that scalar multiplication with $h(m,n)$ do not change the Fisher criterion value, the scalar may be neglected in the optimization.

Note that given a *valid* $\mathbf{r}_{hh}$, finding the corresponding filter $h(m,n)$ is a question of solving an equation set with the coefficients of $h(m,n)$ as the unknowns. However, not every $\mathbf{r}_{hh}$ corresponds to a filter $h(m,n)$. The computation of $\mathbf{r}_{hh}$, which for filters of quadratic size has almost four times the number of coefficients of $h(m,n)$, imposes complex symmetry properties. These properties imply that only particular $\mathbf{r}_{hh}$'s do correspond to any filter $h(m,n)$.

Although an optimal filter design approach using these derivations has not been found, it must still be noted that the criterion value corresponding to $\mathbf{r}_{hh}$ in Equation (A.37) is the maximally attainable criterion value for any $\mathbf{r}_{hh}$. Hence it is an *upper bound* for the Fisher criterion for any filter $h(m,n)$.

A.4.3 One-dimensional or separable optimal filter design

In the case of a one-dimensional filter $h(m,n)$, i.e., a filter where the spatial extent either vertically or horizontally is one, the symmetry relation on $\mathbf{r}_{hh}$ is much simpler. In this case, an optimal $\mathbf{r}_{hh}$ corresponding to to a filter $h(m,n)$ may be found. This *may* lead to an optimal solution for one-dimensional filters (applicable for one-dimensional textures or as separable filters for two-dimensional textures). However, this approach involves the non-trivial problem of solving a large quadratic equation set. This problem is not solved in this dissertation, but still the formulation is given for completeness and as an introduction for further research.

For a real valued, one-dimensional filter $h(m)$, $m = 0 \cdots M-1$, the correlation function is defined to be

$$r_{hh}(k) = E\left\{h(m)h(m+k)\right\}. \tag{A.38}$$

Then it follows that $r_{hh}(k) = r_{hh}(-k)$, hence $r_{hh}(k)$ is symmetric around zero. Since, in the one-dimensional case, $\mathbf{r}_{hh}$ is defined by

$$\mathbf{r}_{hh} = \left[r_{hh}\left(-(M-1)\right) \cdots r_{hh}(M-1)\right]^T, \tag{A.39}$$

it then follows that $\mathbf{r}_{hh}$ is symmetric. If we define

$$\mathbf{r}_{hh}' = \left[r_{hh}\left(-(M-1)\right) \cdots r_{hh}(0)\right]^T, \tag{A.40}$$

we may rewrite the expressions for the mean and variance as

$$\mu_{v_i} = r_{y_iy_i}(0,0) = \mathbf{r}_{hh}'^T \mathbf{r}_{x_ix_i}'(0,0) \tag{A.41}$$

and

$$\sigma_{v_i}^2 = \mathbf{r}_{hh}'^T \mathbf{A}_{x_i}' \mathbf{r}_{hh}, \tag{A.42}$$

The transformations from $\mathbf{r}_{x_i x_i}(0,0)$ to $\mathbf{r}_{x_i x_i}'(0,0)$ and from $\mathbf{A}_{x_i}$ to $\mathbf{A}_{x_i}'$ are trivial. Note that we now have a vector $\mathbf{r}_{hh}'$ with the same number of elements as there are coefficients, M, in the filter we want to optimize.

Inserting the new expressions for the mean and variance into the Fisher criterion expression, differentiating with respect to $\mathbf{r}_{hh}'$ and equating to zero, yields

$$\mathbf{r}_{hh}' = \frac{\sigma_{v_1}^2 + \sigma_{v_2}^2}{\mu_{v_1} - \mu_{v_2}} \left(\mathbf{A}_{x_1}' + \mathbf{A}_{x_2}'\right)^{-1} \left(\mathbf{r}_{x_1 x_1}'(0,0) - \mathbf{r}_{x_2 x_2}'(0,0)\right). \tag{A.43}$$

This, as we see, is on the same form as the previous solution, Equation (A.37). We now have a scheme for optimization of one-dimensional texture filters according to the Fisher criterion. The only assumptions made in this solution are that the filter response, y, is Gaussian and that the input textures are wide sense stationary. The issue of finding $h(m,n)$ from $\mathbf{r}_{hh}'$ still remains to be solved, a problem involving the solution of the large quadratic equation set given by $r_{hh}(k,l) = h(k,l) * h(-k,-l)$.

Bibliography

[1] F. Ade. Characterization of texture by 'eigenfilter'. *Signal Processing*, 5(5):451–457, 1983.

[2] A. N. Akansu and R. A. Haddad. *Multiresolution Signal Decomposition.* Academic Press, San Diego, 1992.

[3] A. Alam and P. Caragounis. Advances in 3D seismic fault interpretation. In *56th Mtg. Eur. Assoc. Expl Geophys., Extended Abstracts*, page B052, 1994.

[4] F. Albregtsen, H. Schulerud, and L. Yang. Texture classification of mouse liver cell nuclei using invariant moments of consistent regions. In V. Hlavac and R. Sara, editors, *CAIP'95 Proceedings, Lecture Notes in Computer Science*, pages 496–502, Check Republic, Sept. 1995.

[5] V. Alvestad. Unsupervised image texture segmentation with optimized filters. Master's thesis, Høgskolen i Stavanger, Stavanger, Norway, 1995.

[6] V. Alvestad and T. Randen. Unsupervised image texture segmentation with optimized filters. In *Proc. NORSIG*, pages 181–186, Stavanger, Norway, Sept. 1995.

[7] M. Bahorich and S. Farmer. 3-D seismic for faults and stratigraphic features: The coherence cube. *The leading edge*, 14(10):1053–1058, Oct. 1995.

[8] M. S. Bahorich. Method of seismic signal processing and exploration. Int. patent application PCT/US95/13644, Amoco Corporation, Chicago, IL, June 1996.

[9] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R*-tree: An efficient and robust access method for points and rectangles. In *Proc. ACM SIGMOD International Conference on Management of Data*, pages 322–331, Atlantic City, NJ, May 1990.

[10] N. N. Bennett, M. Philip, and N. Saito. A method to detect and characterize fractures, beddings, and other geometric features from borehole images using the Hough transform. In *Proc. Schlumberger Signal Processing Wkshp.*, Austin, TX, Feb. 1997. Published electronically.

[11] A. C. Bovik. Analysis of multichannel narrow-band filters for image texture segmentation. *IEEE Trans. Signal Processing*, 39(9):2025–2043, Sept. 1991.

[12] A. C. Bovik, M. Clark, and W. S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Trans. Pattern Anal. Machine Intell.*, 12(1):55–73, Jan. 1990.

[13] A. C. Bovik, M. Clark, and W. S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Trans. Acoust., Speech, Signal Processing*, 12(1):55–73, 1990.

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

[15] F. W. Campbell and J. G. Robson. Application of fourier analysis to the visibility of gratings. *J. Physiology*, 197:551–566, 1968.

[16] T. Chang and C.-C. J. Kuo. Texture analysis and classification with tree-structured wavelet transform. *IEEE Trans. Image Proc.*, 2(4):429–441, 1993.

[17] P. Chauvet, J. Lopez-Krahe, E. Taflin, and H. Maître. System for an intelligent office document analysis, recognition and description. *Signal Processing*, 32(1-2):161–190, 1993.

[18] C.-T. Chen. *System and Signal Analysis*. Saunders College Publishing, Orlando, Forida, USA, 2nd edition, 1994.

[19] D. Clausi and M. Jernigan. Towards a novel approach for segmentation of SAR sea ice imagery. In *26th Symposium on Remote Sensing of Environment and 18th Annual Symposium of the Canadian Remote Sensing Society*, Mar. 1995.

[20] J. M. Coggins. *A Framework for Texture Analysis Based on Spatial Filtering*. PhD thesis, Michigan State University, Dept. of Computer Science, East Lansing, MI 48824–1027, 1982.

[21] J. M. Coggins and A. K. Jain. A spatial filtering approach to texture analysis. *Pattern Recognition Letters*, 3(3):195–203, 1985.

[22] H. A. Cohen and J. You. Texture statistic selective masks. In *Proc. 9th Scandinavian Conf. Image Proc.*, pages 930–935, Oulu, Finland, June 1989.

[23] H. A. Cohen and J. You. A multi-scale texture classifier based on multiresolution 'tuned' mask. *Pattern Recognition Letters*, 13(8):599–604, Aug. 1992.

[24] R. W. Conners and C. A. Harlow. A theoretical comparison of texture algorithms. *IEEE Trans. Pattern Anal. and Machine Intell.*, PAMI-2(3):204–221, May 1980.

[25] I. Daubechies. *Wavelets*. S.I.A.M, Philadelphia, 1992.

[26] J. G. Daugman. Uncertainty relation for resolution in space, spatial-frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of Optical Society of America*, 2(7):1160–1169, 1985.

[27] L. R. Denham. Seismic interpretation. *Proc. IEEE*, 72(10):1255–1265, 1984.

[28] J. M. H. du Buf, M. Kardan, and M. Spann. Texture feature performance for image segmentation. *Pattern Recognition*, 23(3/4):291–309, 1990.

[29] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* Wiley, New York, NY, USA, 1973.

[30] D. F. Dunn and W. E. Higgins. Optimal Gabor filters for texture segmentation. *IEEE Trans. Image Proc.*, 4(7):947–964, July 1995.

[31] D. F. Dunn, W. E. Higgins, and J. Wakeley. Texture segmentation using 2-D Gabor elementary functions. *IEEE Trans. Pattern Anal. and Machine Intell.*, 22(2):130–149, Feb. 1994.

[32] F. Farrokhnia. *Multi-Channel Filtering Techniques for Texture Segmentation and Surface Quality Inspection.* PhD thesis, Michigan State University, 1990.

[33] L. A. Fletcher and R. Kasturi. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Trans. Pattern Anal. and Machine Intell.*, 10(6), Nov. 1988.

[34] M. Flickner, H. Sawhney, W. Niblack, J. Sahley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: The QBIC system. *IEEE Computer*, 28(9):23–32, Sept. 1995.

[35] K. Fukunaga. *Statistical Pattern Recognition.* Academic Press, San Diego, 2nd edition, 1990.

[36] D. Gabor. Theory of communication. *J. Inst Elect. Engr.*, 93:429–457, 1946.

[37] G. L. Gimel'farb and A. K. Jain. On retrieving textured images from an image database. *Pattern Recognition*, 29(9):1461–1483, 1996.

[38] G. H. Golub and C. F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, Maryland, USA, 2nd edition, 1989.

[39] R. C. Gonzales and R. Woods. *Digital Image Processing.* Addison Wesley, 1992.

[40] T. O. Gulsrud and J. H. Husøy. Optimal filter for detection of clustered microcalcifications. *IEEE Trans. Medical Imaging.* Submitted.

[41] T. O. Gulsrud and S. Kjøde. Optimal filter for detection of stellate lesions and circumscribed masses in mammograms. In *Proc. SPIE's Visual Communications and Image Processing 1996*, pages 430–440, Orlando, Florida, Mar. 1996.

[42] R. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Trans. Syst., Man, Cyb.*, 3:610–621, Nov. 1973.

[43] R. M. Haralick. Statistical and structural approaches to texture. *Proc. IEEE*, 67(5):786–804, May 1979.

[44] S. Haykin. *Adaptive Filter Theory*. Prentice Hall, Englewood Cliffs, NJ, USA, 2nd edition, 1991.

[45] P. V. C. Hough. Methods and means for recognizing complex patterns. US Patent 3,069,654, 1962.

[46] J. H. Husøy. Low complexity subband coding of still images and video. *Optical Engineering*, 30(7):904–911, July 1991.

[47] J. H. Husøy, T. Randen, and T. O. Gulsrud. Image texture classification with digital filter banks and transforms. In *Proc. SPIE Int. Symp. on Optical Inst. and Applied Science: Appl. of Digital Image Proc. XVI*, pages 260–271, San Diego, July 1993.

[48] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Englewood Cliffs, 1989.

[49] A. K. Jain and S. Bhattacharjee. Text segmentation using Gabor filters for automatic document processing. *Machine Vision and Applications*, 5(3):169–184, 1992.

[50] A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

[51] A. K. Jain and K. Karu. Learning texture discrimination masks. *IEEE Trans. Pattern Anal. and Machine Intell.*, 18(2):195–205, Feb. 1996.

[52] A. K. Jain and D. Zongker. Feature selection: Evaluation, application, and small sample performance. *IEEE Trans. Pattern Anal. and Machine Intell.*, 19(2):153–158, Feb. 1997.

[53] J. D. Johnston. A filter family designed for use in quadrature mirror filter banks. In *Proc. Int. Conf. Acoust. Speech, Signal Proc.*, pages 291–294, Denver, CO, 1980. IEEE.

[54] J. H. Justice, D. J. Hawkins, and G. Wong. Multidimensional attribute analysis and pattern recognition for seismic interpretation. *Pattern Recognition*, 18(6):391–407, 1985.

[55] K. Karu, A. K. Jain, and R. M. Bolle. Is there any texture in the image? *Pattern Recognition*, 29(9):1437–1446, 1996.

[56] M. Kass and A. Witkin. Analyzing oriented patterns. *Computer Vision, Graphics and Image Processing*, 37:362–385, 1987.

[57] P. Kearey and M. Brooks. *An introduction to Geophysical Exploration*. Blackwell Scientific Publ., Oxford, England, 2nd edition, 1991.

[58] T. Kohonen. The self-organizing map. *Proc. IEEE*, 78(9):1464–1480, Sept. 1990.

[59] M. Krishnamoorthy, G. Nagy, S. Seth, and M. Viswanathan. Syntactic segmentation and labeling of digitized pages from technical journals. *IEEE Trans. Pattern Anal. and Machine Intell.*, 15(7):737–747, 1993.

[60] A. Laine and J. Fan. Texture classification by wavelet packet signatures. *IEEE Trans. Pattern Anal. Machine Intell.*, 15(11):1186–1190, Nov. 1993.

[61] N. Lamei, K. D. Hutchinson, M. M. Crawford, and N. Khazenie. Cloud-type discrimination via multispectral textural analysis. *Optical Engineering*, 33(4):1303–1313, Apr. 1994.

[62] K. I. Laws. Rapid texture identification. In *Proc. of the SPIE Conference on Image Processing for Missile Guidance*, pages 376–380, 1980.

[63] L. Linnett, S. Clarke, C. Graham, and D. Langhorne. Remote sensing of the seabed using fractal techniques. *Electronics & Communication Engineering Journal*, 3(5):195–203, 1991.

[64] F. Liu and R. W. Picard. Peridicity, directionality, and randomness: Wold features for image modeling and retrieval. *IEEE Trans. Pattern Anal. and Machine Intell.*, 18(7):722–733, July 1996.

[65] H. D. Lüke. *Signal-Übertragung: Grundlagen der Digitalen und Analogen Nachrichtenübertragungssysteme.* Springer-Verlag, Berlin, Germany, 3rd edition, 1985.

[66] A. Mahalanobis and H. Singh. Application of correlation filters for texture recognition. *Applied Optics*, 33(11):2173–2179, 1994.

[67] S. G. Mallat. Multifrequency channel decomposition of images and wavelet models. *IEEE Trans. Acoust., Speech, Signal Processing*, 37(12):2091–2110, Dec. 1989.

[68] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Anal. and Machine Intell.*, 11(7):674–693, July 1989.

[69] U. Manber. *Introduction to Algorithms: A Creative Approach.* Addison-Wesley, Reading, MA, USA, 1989.

[70] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. and Machine Intell.*, 18(8):837–842, Aug. 1996.

[71] J. Mao and A. K. Jain. Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern Recognition*, 25(2):173–188, 1992.

[72] E. D. McKee and G. W. Weir. Terminology for stratification and cross-stratification in sedimentary rocks. *Geol. Soc. America Bull.*, 64:381–389, 1953.

[73] MIT Vision and Modelling Group. http://www.media.mit.edu/vismod/.

[74] R. M. Mitchum, Jr. Seismic stratigraphy and global changes of sea level, part 11: Glossary of terms used in seismic stratigraphy. *AAPG Memoir*, 26:205–212, 1977.

[75] R. M. Mitchum, Jr., P. R. Vail, and S. Thompson. Seismic stratigraphy and global changes of sea level, part 2: The depositional sequence as a basic unit for stratigraphic analysis. *AAPG Memoir*, 26:53–63, 1977.

[76] G. Nagy, S. Seth, and M. Viswanathan. A prototype document image analysis system for technical journals. *Computer*, 25(7):10–22, 1992.

[77] F. Nergård. Random search optimization of filters for texture identification in digital images. Master's thesis, Høgskolen i Stavanger, Stavanger, Norway, 1996.

[78] F. Nergård and T. Randen. Optimization of spatial filters for image texture feature separation by simulated annealing. In *Proc. NORSIG*, pages 251–254, Helsinki, Finland, Sept. 1996.

[79] I. Ng, T. Tan, and J. Kittler. On local linear transform and Gabor filter representation of texture. In *Proc. Int. Conf. on Patt. Rec.*, pages 627–631. Int. Assoc. for Patt. Rec., 1992.

[80] P. P. Ohanian and R. C. Dubes. Performance evaluation for four classes of textural features. *Pattern Recognition*, 25(8):819–833, 1992.

[81] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29(1):51–59, 1996.

[82] M. Østebø. Application of template matching for structural interpretation of seismic images. Master's thesis, Stavanger College (Høgskolen i Stavanger), Stavanger, Norway, 1997.

[83] A. Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, New York, 3rd edition, 1991.

[84] D. Patel. Page segmentation for document image analysis using a neural network. *Optical Engineering*, 35(7):1854–1861, July 1996.

[85] C. E. Payton, editor. *Seismic Stratigraphy – Applications to Hydrocarbon Exploration.* AAPG, Tulsa, OK, 1977.

[86] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: Tools for content-based manipulation of image databases. In *Proc. SPIE Conf. on Storage and Retrieval of Image and Video Databases II*, pages 34–47, San Jose, CA, Feb. 1994.

[87] A. Perry and D. G. Lowe. Segmentation of textured images. In *Proc. Computer Vis. and Pattern Rec.*, pages 319–325. IEEE Comp. Soc., 1989.

[88] R. W. Picard. A society of models for video and image libraries. Perceptual Computing Section Technical Report 360, MIT Media Lab, Cambridge, MA, 1996.

[89] R. W. Picard and T. Kabir. Finding similar patterns in large image databases. In *Proc. Int. Conf. Acoust. Speech, Signal Proc.*, volume V, pages 161–164, Minneapolis, MN, Apr. 1993.

[90] R. W. Picard and T. P. Minka. Vision texture for annotation. *J. Multimedia Systems*, 3:3–14, 1995.

[91] R. W. Picard, T. P. Minka, and M. Szummer. Modeling user subjectivity in image libraries. In *Proc. Int. Conf. Image Proc.*, Lausanne, Switzerland, Sept. 1996.

[92] M. Pietikäinen, A. Rosenfeld, and L. S. Davis. Experiments with texture classification using averages of local pattern matches. *IEEE Trans. Syst., Man, Cyb.*, 13(3):421–426, 1983.

[93] I. Pitas and C. Kotropoulos. A texture-based approach to the segmentation of seismic images. *Pattern Recognition*, 25(9):929–945, 1992.

[94] J. G. Proakis, C. M. Rader, F. Ling, and C. L. Nikias. *Advanced Digital Signal Processing*. Macmillan Publishing Company, New York, 1992.

[95] H. Ramampiaro. Exact algorithms for texture classification. Master's thesis, Stavanger College (Høgskolen i Stavanger), Stavanger, Norway, 1997.

[96] T. A. Ramstad, S. O. Aase, and J. H. Husøy. *Subband Compression of Images – Principles and Examples*. ELSEVIER Science Publishers BV, North Holland, 1995.

[97] T. Randen. Flerkanal filtrering for tekstursegmentering. Master's thesis, Rogaland University Center, Stavanger, Norway, 1993 (in Norwegian).

[98] T. Randen. Analysis of faults and stratigraphic terminations in seismic sections. In *Proc. Schlumberger Signal Processing Wkshp.*, Austin, TX, Feb. 1997. Published electronically.

[99] T. Randen, V. Alvestad, and J. H. Husøy. Optimal filtering for unsupervised texture feature extraction. In *Proc. SPIE's Visual Comm. and Image Proc.*, pages 441–452, Orlando, FL, Mar. 1996.

[100] T. Randen and J. H. Husøy. Texture segmentation using filters with optimized energy separation. *IEEE Trans. Image Proc.* Submitted, in second phase of review.

[101] T. Randen and J. H. Husøy. Filter banks for texture segmentation. *Piksel'n*, 10(3):22–29, Sept. 1993.

[102] T. Randen and J. H. Husøy. Novel approaches to multichannel filtering for image texture segmentation. In *Proc. SPIE's Visual Comm. and Image Proc.*, pages 626–636, Boston, Nov. 1993.

[103] T. Randen and J. H. Husøy. Least squares image texture analysis and synthesis. Working papers from Høgskolen i Stavanger 197, Høgskolen i Stavanger, Stavanger, Norway, 1994.

[104] T. Randen and J. H. Husøy. Multichannel filtering for image texture segmentation. *Optical Engineering*, 33(8):2617–2625, Aug. 1994.

[105] T. Randen and J. H. Husøy. Segmentation of text/image documents using texture approaches. In *Proc. NOBIM-konferansen-94*, pages 60–67, Asker, Norway, June 1994.

[106] T. Randen and J. H. Husøy. Texture segmentation with optimal linear prediction error filters. *Piksel'n*, 11(3):25–28, Sept. 1994.

[107] T. Randen and J. H. Husøy. Design of optimal filters for image texture feature separation. In *Proc. NORSIG*, pages 221–226, Stavanger, Norway, Sept. 1995.

[108] T. Randen and J. H. Husøy. Optimal texture filtering. In *Int. Conf. on Image Proc.*, pages 374–377, Washington, D.C., Oct. 1995.

[109] T. Randen and J. H. Husøy. Image content search by color and texture properties. In *Proc. Int. Conf. Image Proc.*, Santa Barbara, CA, Oct. 1997.

[110] T. Randen and J. H. Husøy. Optimal filter-bank design for multiple texture discrimination. In *Proc. Int. Conf. Image Proc.*, Santa Barbara, CA, Oct. 1997.

[111] T. Randen and J. H. Husøy. Optimal texture filter design using feature extraction modeling. Working papers from Høgskolen i Stavanger 27, Høgskolen i Stavanger, Stavanger, Norway, 1997.

[112] T. Randen and J. H. Husøy. Segmentation of text/image documents using texture approaches. Working papers from Høgskolen i Stavanger 28, Høgskolen i Stavanger, Stavanger, Norway, 1997.

[113] A. R. Rao and B. G. Schunck. Computing oriented texture fields. *CVGIP: Graphical Models and Image Processing*, 53(2):157–185, Mar. 1991.

[114] T. R. Reed and J. M. H. D. Buf. A review of recent texture segmentation and feature extraction techniques. *CVGIP: Image Understanding*, 57:359–372, May 1993.

[115] N. Saito and R. R. Coifman. Local discriminant bases and their applications. *J. Mathematical Imaging and Vision*, 5(4):337–358, 1995.

[116] R. Schalkoff. *Pattern Recognition: Statistical, Structural and Neural Approaches.* Wiley, New York, 1992.

[117] M. Schwab, J. F. Claerbout, and C. Holden. Revealing geological discontinuities by plane reflector suppression. In *Int. Mtg., Soc. Exploration Geophys., Expanded Abstr.*, pages 302–305, 1996.

[118] S. R. Searle. *Matrix Algebra Useful for Statistics.* John Wiley & Sons, NY, USA, 1982.

[119] R. C. Selley. *Petroleum Geology for Geophysicists and Engineers.* Int. Human Resources Developmennt Corp., Boston, MA, USA, 1983.

[120] H. Singh and A. Mahalanobis. Correlation filters for texture recognition and applications to terrain-delimination in wide-area surveillance. In *Proc. Int. Conf. Acoust. Speech, Signal Proc.*, pages V153–V156. IEEE, 1994.

[121] S. M. Smith and J. M. Brady. SUSAN – A new approach to low level image processing. Technical Report TR95SMS1c, Defence Research Agency, Chertsey, Surrey, UK, 1995.

[122] S. N. Srihari. Document image understanding. In *Proc. IEEE Computer Society Fall Joint Computer Conf.*, pages 87–96, Dallas, TX, Nov. 1986.

[123] J. Strand and T. Taxt. Local frequency features for texture classification. *Pattern Recognition*, 27(10):1397–1406, 1994.

[124] G. Strang and T. Nguyen. *Wavelets and Filter Banks.* Wellesley-Cambridge Press, 1996.

[125] B. J. Super and A. C. Bovik. Shape from texture using local spectral moments. *IEEE Trans. Pattern Anal. and Machine Intell.*, 17(4):333–343, Apr. 1995.

[126] T. N. Tan and A. G. Constantinides. Texture analysis based on a human visual model. In *Proc. Int. Conf. Acoust. Speech, Signal Proc.*, pages 2137–2140, Alberquerque, NM, 1990. IEEE.

[127] A. Teuner, O. Pichler, and B. J. Hosticka. Unsupervised texture segmentation of images using tuned matched Gabor filters. *IEEE Trans. Image Proc.*, 4(6):863–870, June 1995.

[128] M. Tuceryan and A. K. Jain. Texture analysis. In C. H. Chen, L. F. Pau, and P. S. P. Wang, editors, *Handbook of Pattern Recognition and Computer Vision*, chapter 2, pages 235–276. World Scientific, Singapore, 1993.

[129] M. R. Turner. Texture discrimination by Gabor functions. *Biological Cybernetics*, 55:71–82, 1986.

[130] M. Unser. Local linear transforms for texture measurements. *Signal Processing*, 11(1):61–79, 1986.

[131] M. Unser. Texture classification and segmentation using wavelet frames. *IEEE Trans. Image Proc.*, 4(11):1549–1560, Nov. 1995.

[132] M. Unser and M. Eden. Nonlinear operators for improving texture segmentation based on features extracted by spatial filtering. *IEEE Trans. Syst., Man, Cyb.*, 20:804–815, 1990.

[133] P. P. Vaidyanathan. Quadrature mirror filter banks, M-band extensions and perfect-reconstruction techniques. *IEEE ASSP Magazine*, 4(3):4–20, July 1987.

[134] B. J. H. Verwer, P. W. Verbeek, and S. T. Dekker. An efficient uniform cost algoritm applied to distance transforms. *IEEE Trans. Pattern Anal. and Machine Intell.*, 11:425–429, 1989.

[135] F. M. Wahl, K. Y. Wong, and R. G. Casey. Block segmentation and text extraction in mixed text/image documents. *Computer Graphics and Image Processing*, 20:375–390, 1982.

[136] D. Wang and S. N. Srihari. Classification of newspaper image blocks using texture analysis. *Computer Vision, Graphics and Image Processing*, (47):327–352, 1989.

[137] Webster's Encyclopedic Unabridged Dictionary of the English Language. Random House, New York, NY, USA, 1989.

[138] T. P. Weldon and W. E. Higgins. Design of multiple gabor filters for texture segmentation. In *Proc. Int. Conf. Acoust. Speech, Signal Proc.*, pages 2243–2246, Atlanta, GA, May 1996.

[139] T. P. Weldon and W. E. Higgins. Integrated approach to texture segmentation using multiple gabor filters. In *Proc. Int. Conf. Image Proc.*, pages 955–958, Lausanne, Switzerland, Sept. 1996.

[140] J. S. Weszka, C. R. Dyer, and A. Rosenfeld. A comparative study of texture measures for terrain classification. *IEEE Trans. Syst., Man, Cyb.*, 6(4):269–285, Apr. 1976.

[141] D. A. White and R. Jain. Similarity indexing with the SS-tree. In *Proc. 12th IEEE International Conference on Data Engineering*, New Orleans, Louisiana, Feb. 1996.

[142] A. W. Whitney. A direct method of nonparametric measurement selection. *IEEE Trans. Commun.*, pages 1100–1103, 1971.

[143] J. You and H. A. Cohen. Classification and segementation of rotated and scaled textured images using texture "tuned" masks. *Pattern Recognition*, 26(2):245–258, 1993.