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

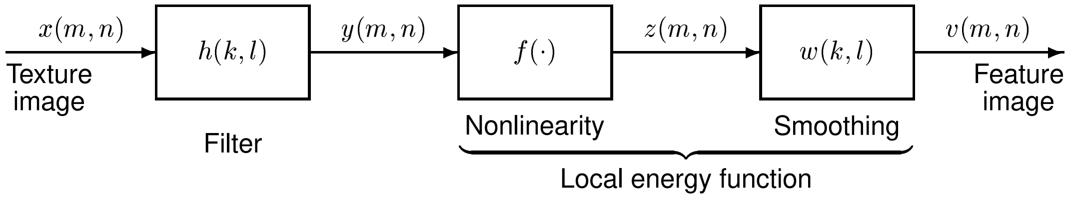


Figure 5.1: The experimental setup for the feature extractionexperiments with optimized filters.

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

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],

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

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

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

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

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

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

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

$\mu_{x_i} = E\left\{x_i(m,n)\right\}$, 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.

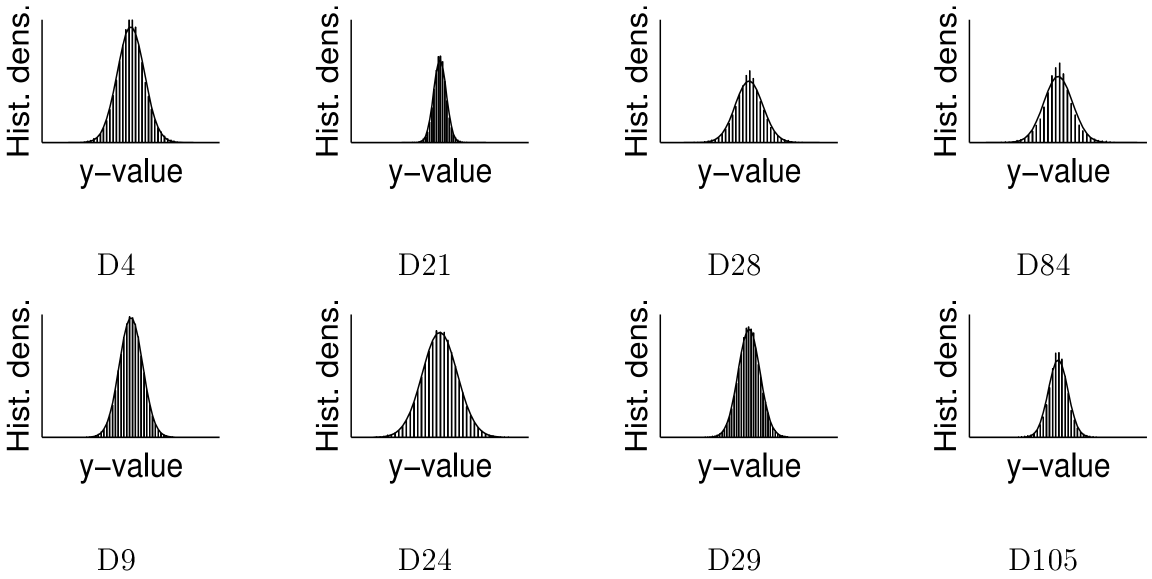


Figure 5.2: Filter response histograms from filters designed in the experiments illustrated in Figure 5.8, i.e., closed form optimization with respect to the Fisher criterion. The continuous curves illustrate the corresponding Gaussian distributions with the same means and variances.

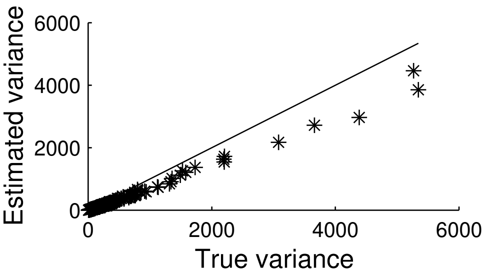


Figure 5.3: Estimated feature variance (vertical axis) versus true feature variance for a number of random drawn filters $h(m, n)$ and the Brodatz texture D4.

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

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

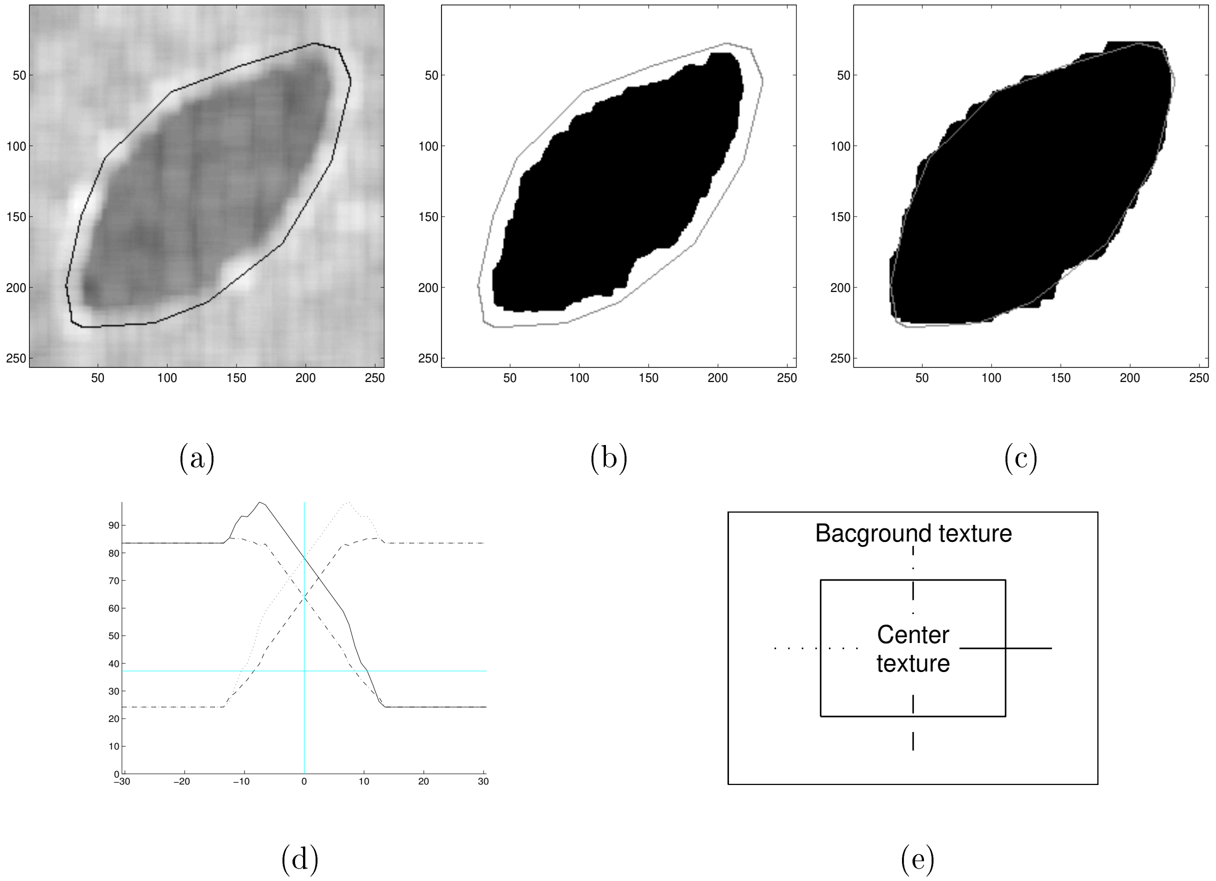


Figure 5.4: Illustration of edge modeling and compensation. (a) Feature image with biased edge response and (b) corresponding minimum error class map. (c) Region map images after edge effect compensation by morphological dilation. (d) Edge modeling of the feature values across a texture edge. The horizontal 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

$$\lambda = \frac{(\mathbf{h}^T \mathbf{R}_{x_1x_1} \mathbf{h})}{(\mathbf{h}^T \mathbf{R}_{x_2x_2} \mathbf{h})}. \tag{5.24}$$

The optimal filter coefficient vector is found as the eigenvector of $(\mathbf{R}_{x_2x_2}{}^{-1}\mathbf{R}_{x_1x_1})$ 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,

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

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.

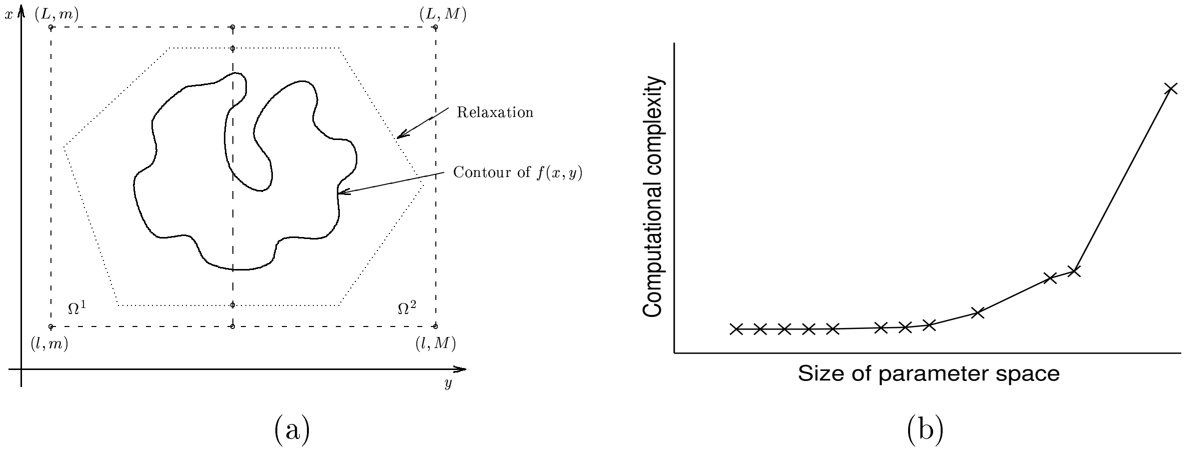


Figure 5.5: Exact iterative optimization: (a) Illustration of the principle of linear relaxation. (b) Computation time versus number of parameters in the system.

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.

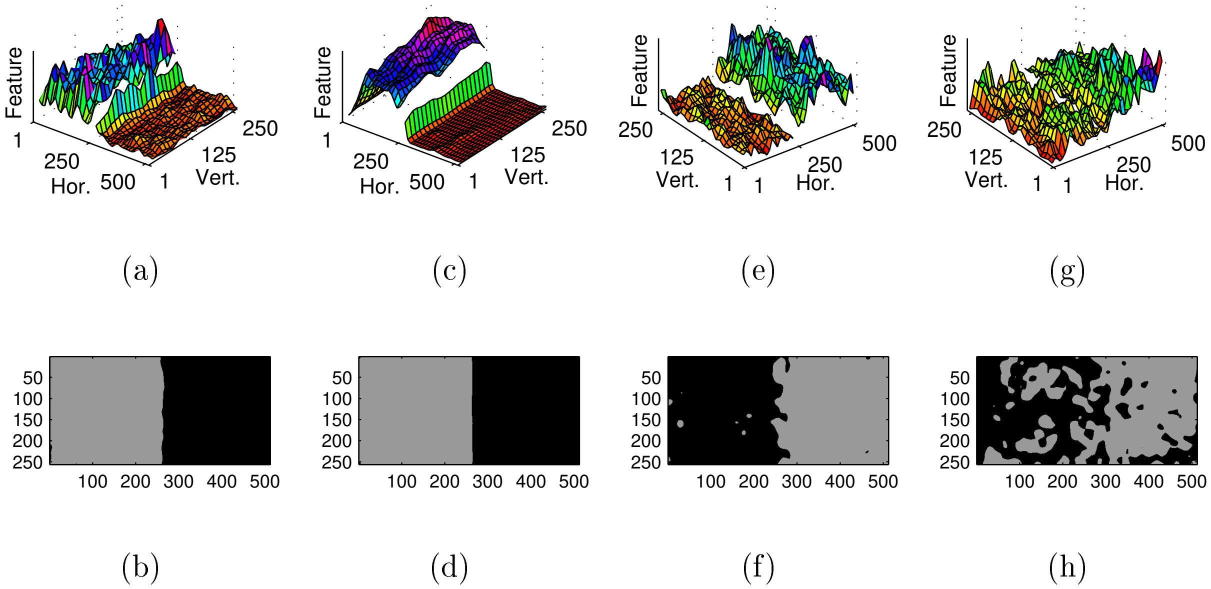


Figure 5.6: Features optimal with respect to J_U, Equation (5.4) for the textures (a) D4-D84, (c) D21-D105, (e) D28-D29, and (g) D9-D24 and their corresponding class maps (b), (d), (f), and (h). The true texture boundaries are indicated by discontinuities on the feature energy surfaces and the horizontal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

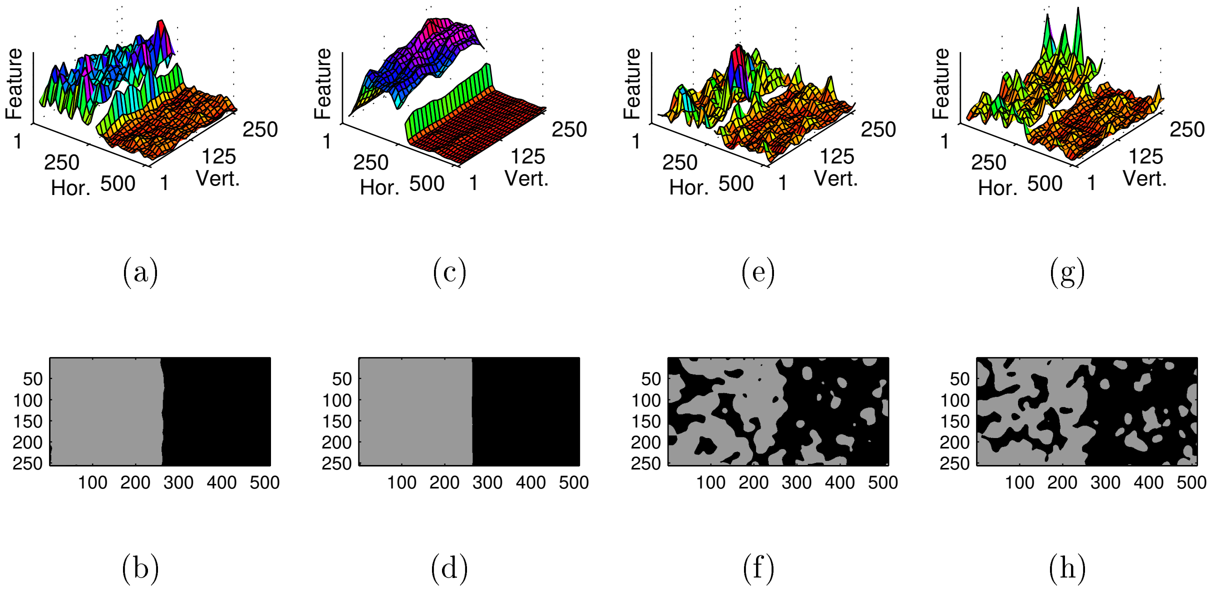


Figure 5.7: Features optimal with respect to J_{MS}, Equation (5.5) [66] for the textures (a) D4-D84, (c) D21-D105, (e) D28-D29, and (g) D9-D24 and their corresponding class maps (b), (d), (f), and (h).

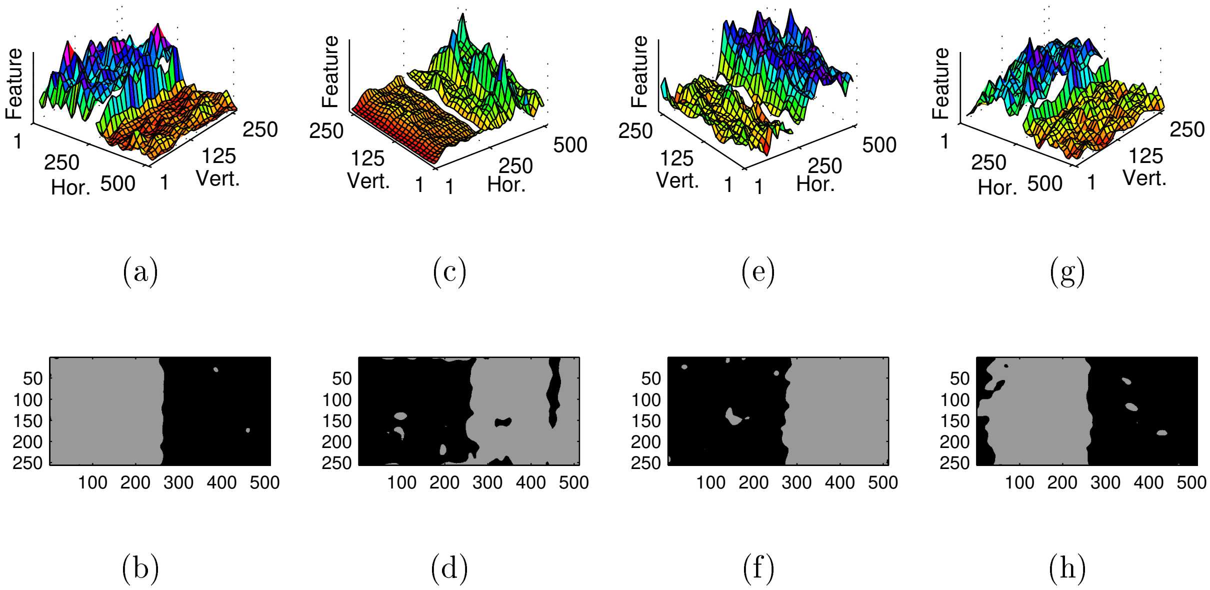


Figure 5.8: Features optimized with closed form solution for J_F, Equation (5.25), for the textures (a) D4-D84, (c) D21-D105, (e) D28-D29, and (g) D9-D24 and their corresponding class maps (b), (d), (f), and (h).

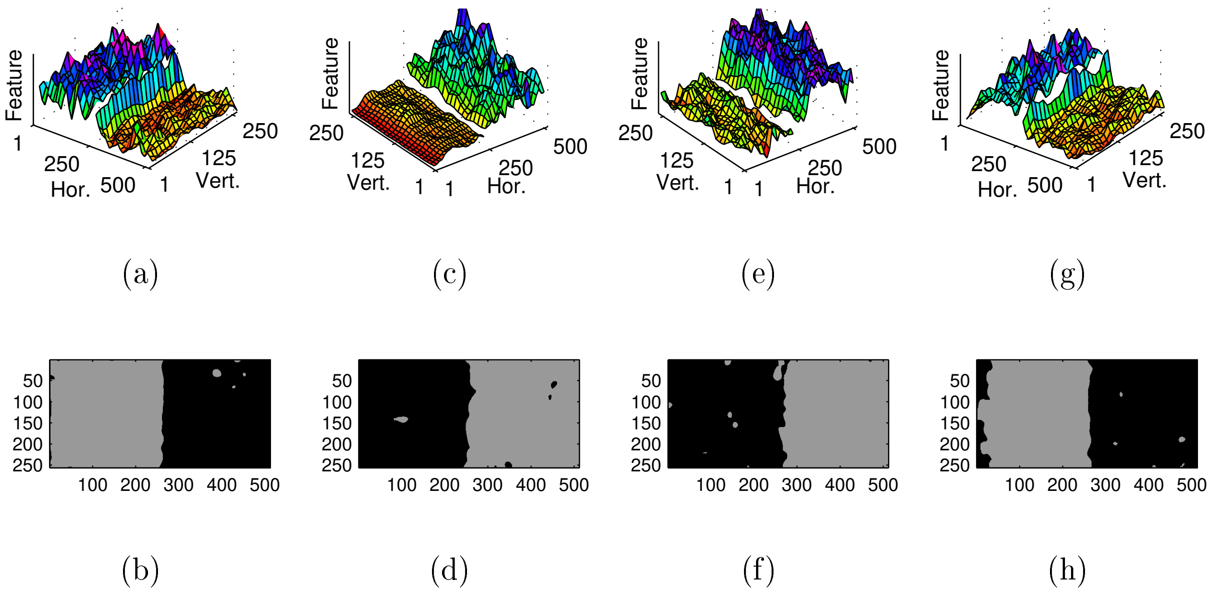


Figure 5.9: Features optimized with gradient search for J_F, Equation (5.25), using the closed form solution as the starting point, for the textures (a) D4-D84, (c) D21-D105, (e) D28-D29, and (g) D9-D24 and their corresponding class maps (b), (d), (f), and (h).

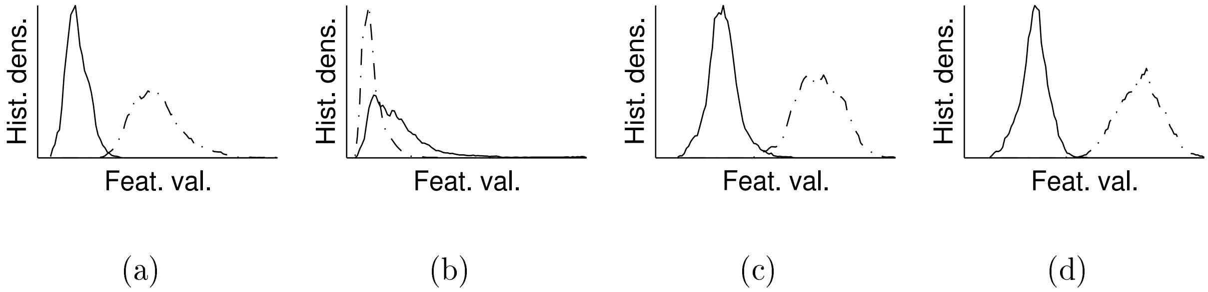


Figure 5.10: Feature histograms visualizing feature separation for texture pair D28-D29 for filters designed with respect to (a) J_U, (b) J_{MS}, (c) J_F closed form, and (d) J_F gradient search.

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.

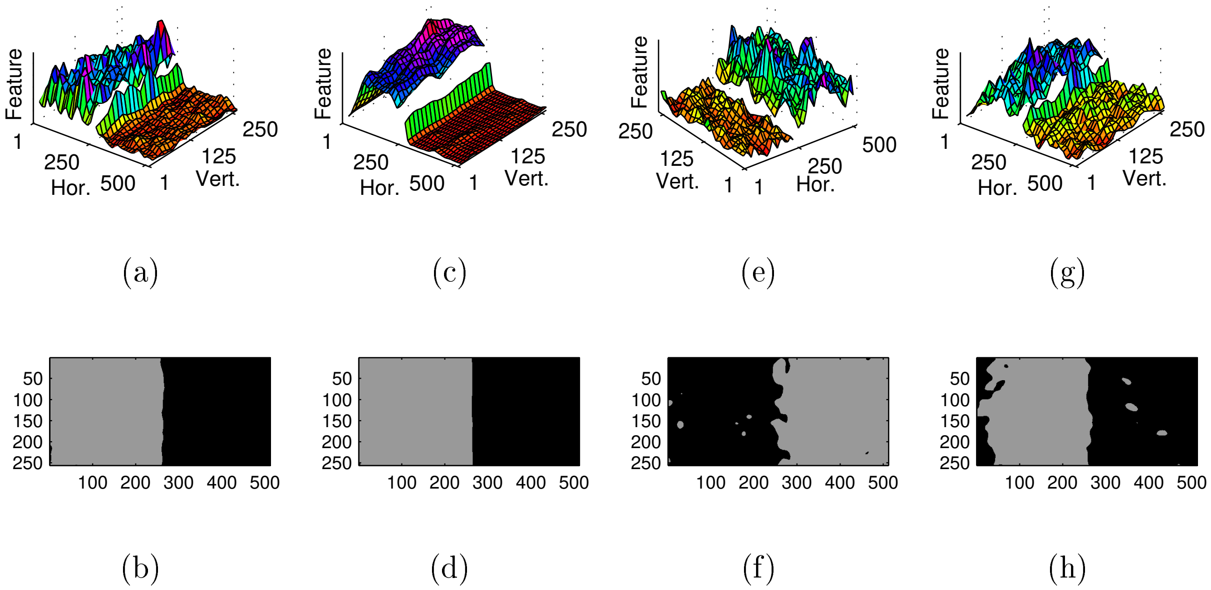


Figure 5.11: Features corresponding to optimal filters selected yielding minimum histogram overlap for the textures (a) D4-D84, (c) D21-D105, (e) D28-D29, and (g) D9-D24 and their corresponding class maps (b), (d), (f), and (h).

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

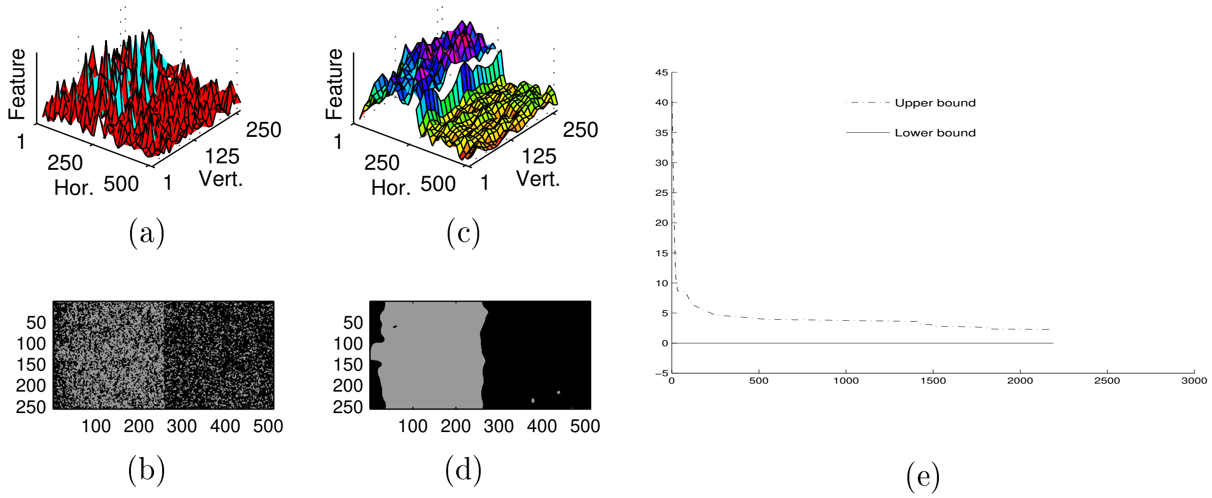
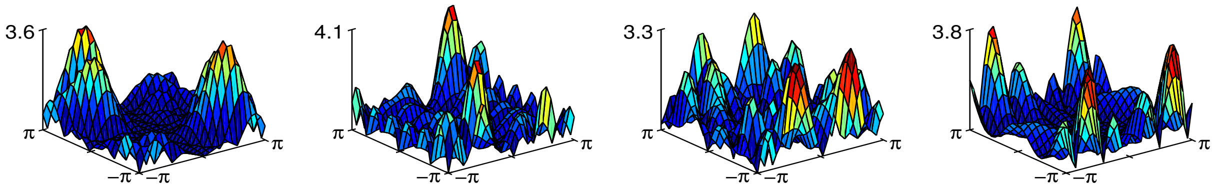


Figure 5.12: Features optimized with exact iterative optimization for J_F, Equation (5.25) for texture D9-D24. The optimal filter is of size $M_h \times N_h = 3 \times 3$, assuming a rectangular smoothing filter $w(m, n)$ of size only $M_w \times N_w = 3 \times 3$. (a) Feature surface and (b) class map 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).

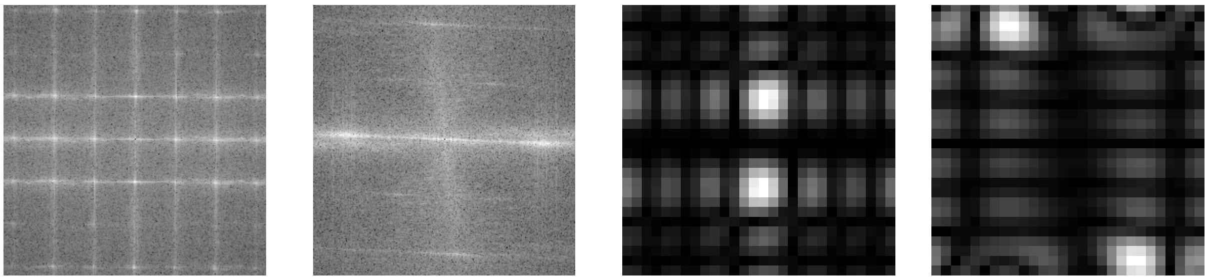
To illustrate what the optimal filters may look like, examples of amplitude responses are given in Figure 5.13. Two examples of amplitude spectra illustrated as gray-level images in conjunction with the amplitude spectra of the two corresponding textures are shown in Figure 5.14. This texture set, D21-D105, 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

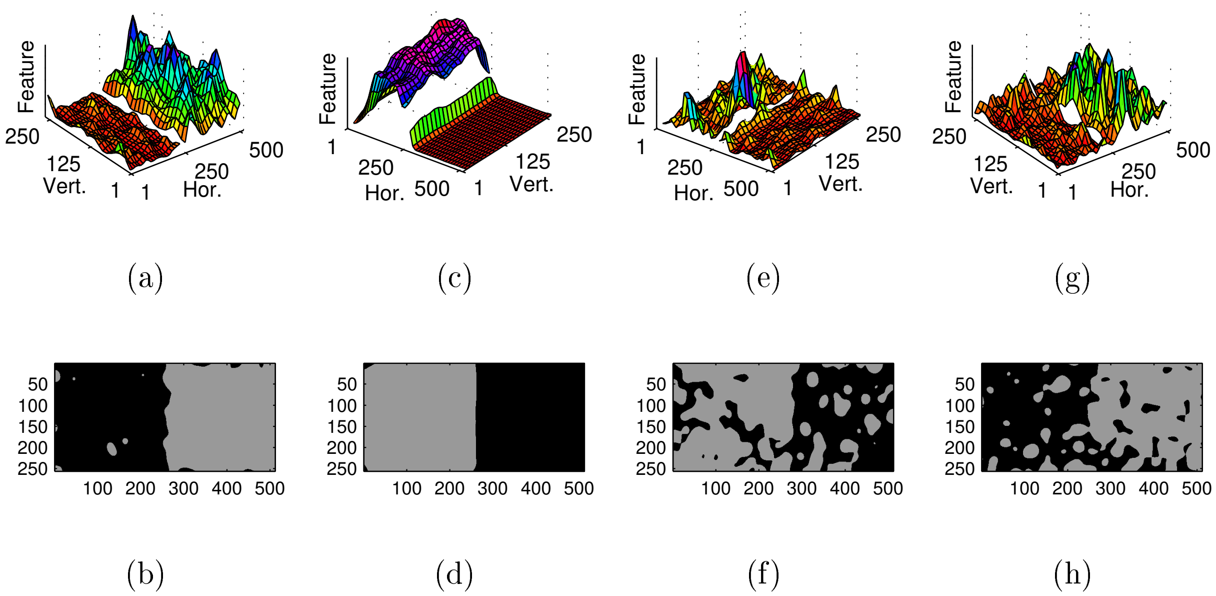


Figure 5.15: Features extracted with optimal Gabor filter [30] for the textures (a) D4-D84, (c) D21-D105, (e) D28-D29, and (g) D9-D24 and their corresponding class maps (b), (d), (f), and (h).

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 $\mu_{v_i} = (\mathbf{h}^T\mathbf{R}_{x_ix_i}\mathbf{h})$. 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

$$\begin{aligned}
\mu_v &= \frac{1}{I}\left(\mu_{v_1} + \cdots + \mu_{v_I}\right) \\
&= \frac{1}{I}\left((\mathbf{h}^T\mathbf{R}_{x_1x_1}\mathbf{h}) + \cdots + (\mathbf{h}^T\mathbf{R}_{x_Ix_I}\mathbf{h})\right) \\
&= \mathbf{h}^T\frac{1}{I}\left(\mathbf{R}_{x_1x_1} + \cdots + \mathbf{R}_{x_Ix_I}\right)\mathbf{h} \\
&= \mathbf{h}^T\mathbf{R}_{x_ax_a}\mathbf{h},
\end{aligned} \tag{5.32}$$

where $\mathbf{R}_{x_ax_a}$ is the average $\mathbf{R}_{x_ix_i}$. This expression is identical to the expression for the mean of one texture if we replace $\mathbf{R}_{x_ix_i}$ with the average $\mathbf{R}_{x_ix_i}$'s, $\mathbf{R}_{x_ax_a}$, of all the textures in the group. Hence, by replacing $\mathbf{R}_{x_1x_1}$ and $\mathbf{R}_{x_2x_2}$ with the $\mathbf{R}_{x_ax_a}$'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, $\mathbf{R}_{x_ix_i}$ for the entire texture is approximately equal to the mean $\mathbf{R}_{x_ix_i}$'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

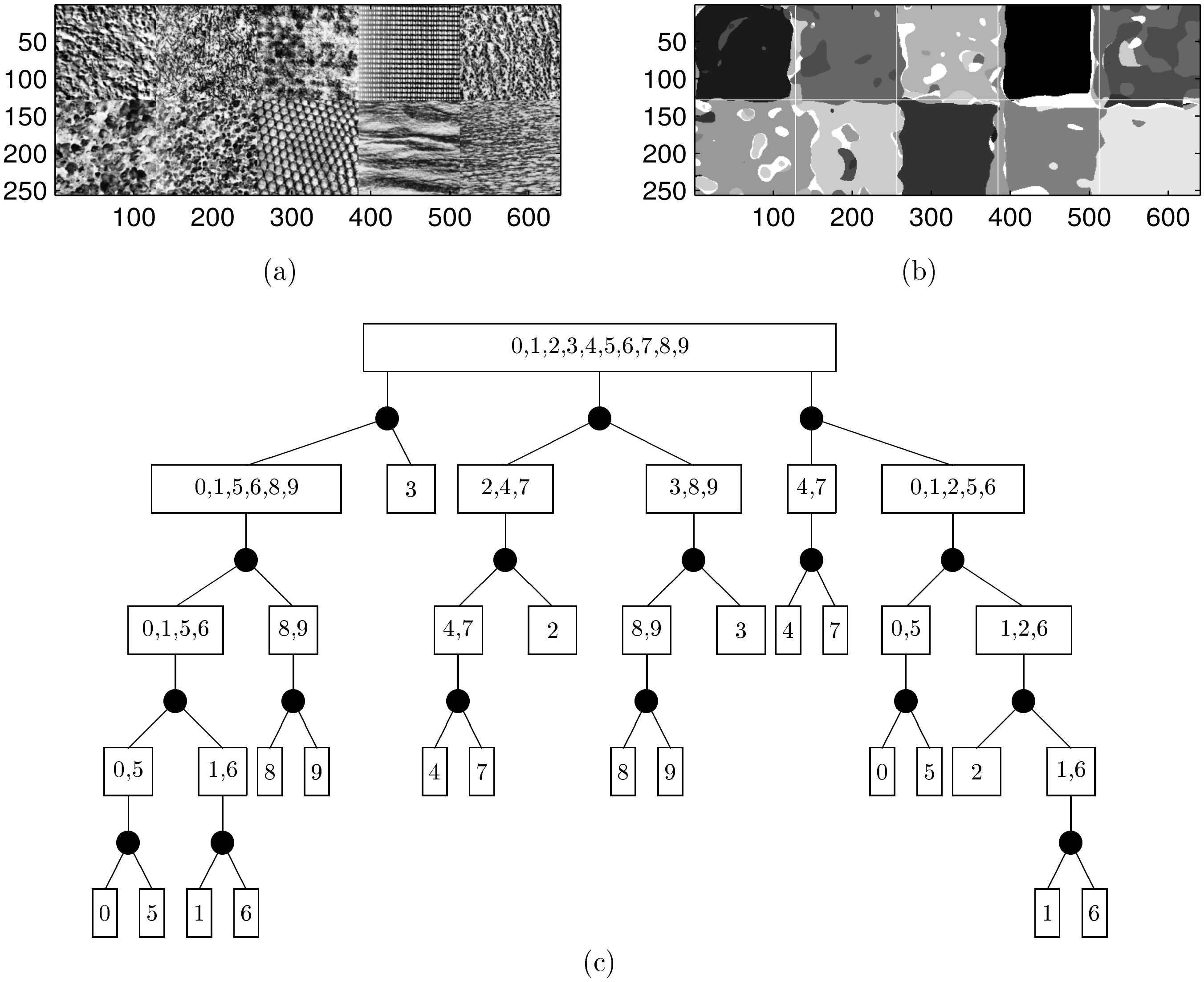


Figure 5.16: Multi-texture filter optimization experiment. (a) Ten-texture image, (b) segmentation by the multi-texture approach, and (c) texture grouping hierarchy where each black disc corresponds to one filter. The numbers in (c) refer to texture number, where the textures in (a) are numbered 0 to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

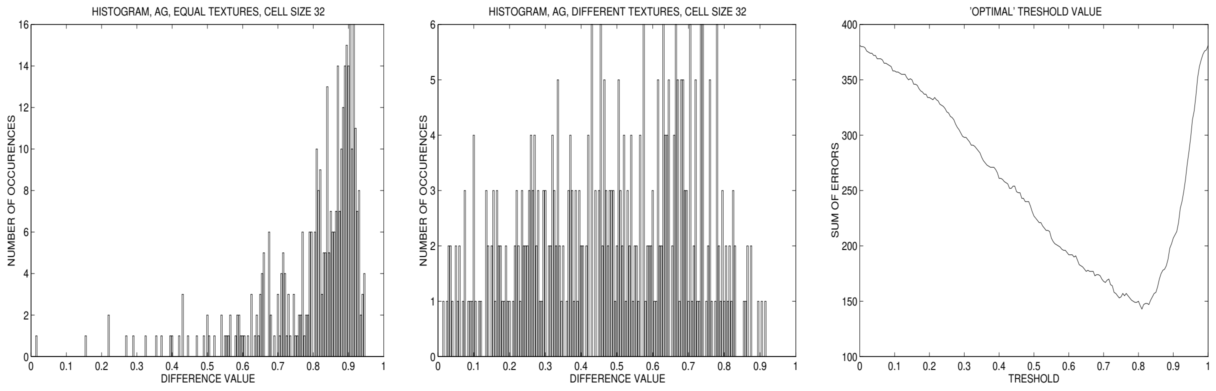
$$d_{i,j} = \frac{\frac{1}{N}\sum_{n=1}^{N}\lambda_n}{\left(\prod_{n=1}^{N}\lambda_n\right)^{\frac{1}{N}}}. \tag{5.35}$$

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.

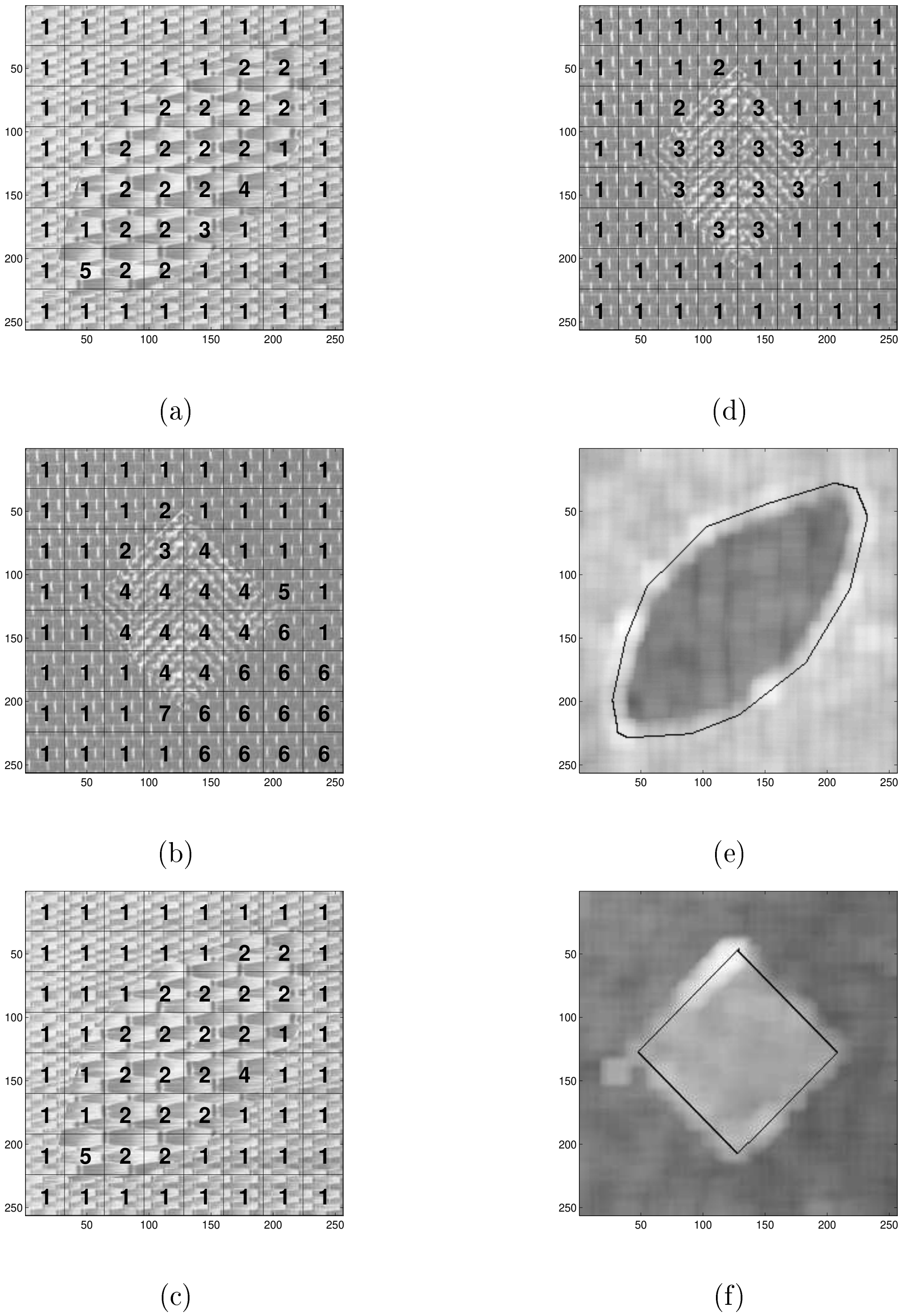


Figure 5.18: Unsupervised experiments. Texture images used in the experiments; (a) D55 and D56 [14] and (b) D6 and D176 [14]. The clustering results for the cell clustering are shown overlayed. (c-d) The clustering after merging similar cell clusters. (e-f) The feature images corresponding to Fisher-optimal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

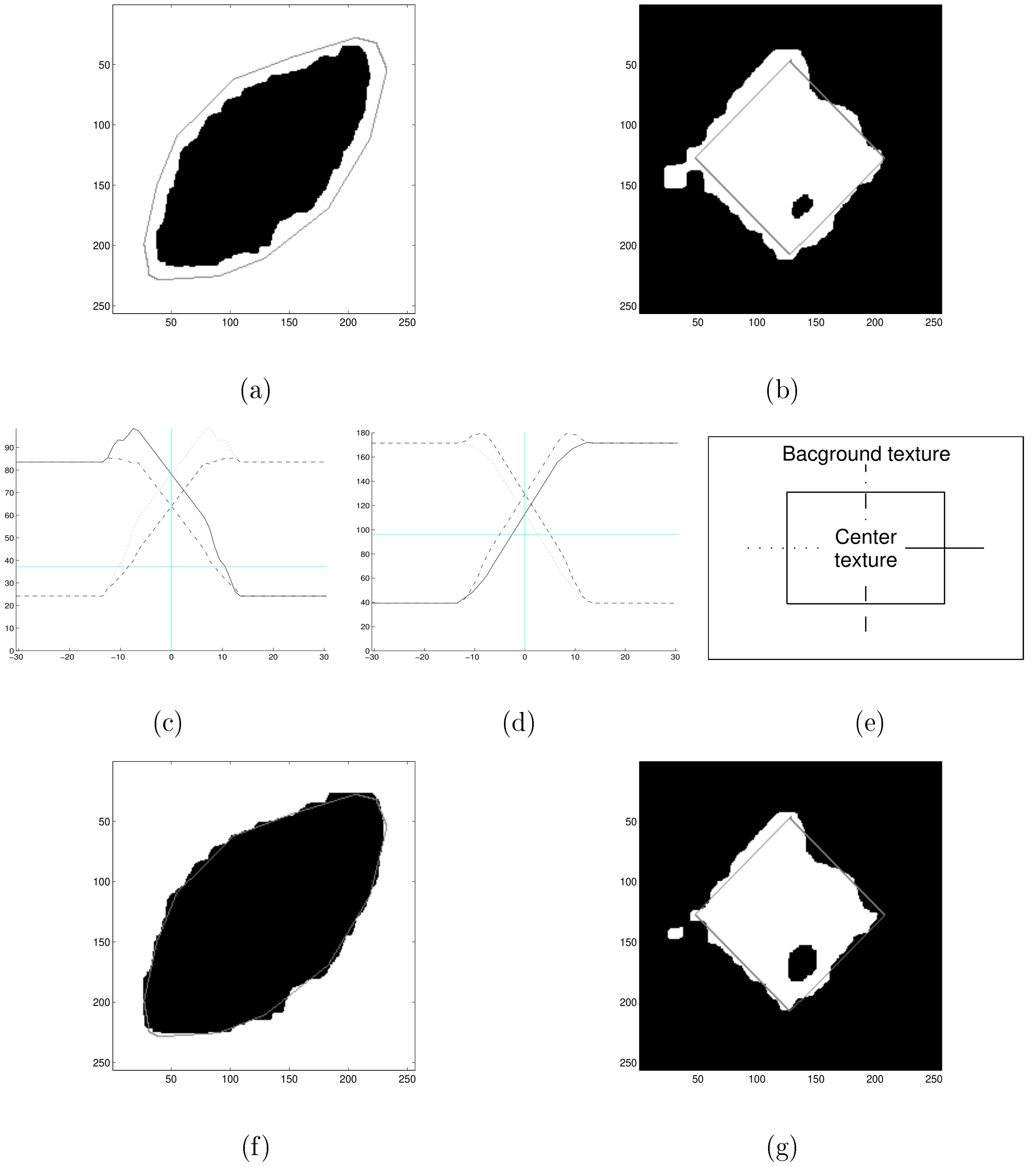


Figure 5.19: (a-b) Class map images resulting from minimum error thresholding of the feature images of Figures 5.18(e-f). The lines in the images indicate the true texture boundaries. (c-d) Edge modeling of the features of Figures 5.18(e-f), representing expected feature values across a texture edge. 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.

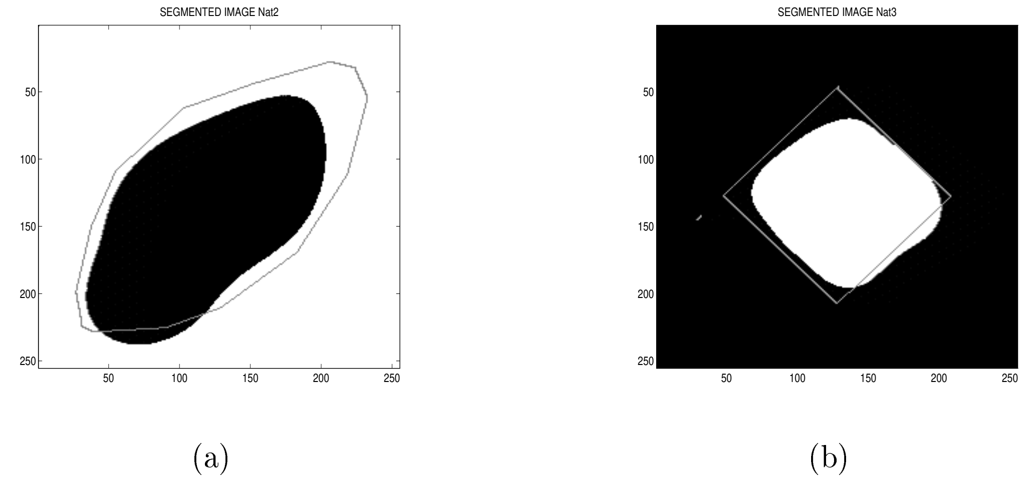


Figure 5.20: Unsupervised segmentation results for the images in Figures 5.18(a-b) using optimal Gabor filters as proposed by Teuner et al.

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.