# Optimal Texture Filter Design using Feature Extraction Modeling

Trygve Randen and Dr. John Håkon Husøy. *

## Abstract

This paper presents a new method for modeling the means and variances of texture features extracted by filtering and local energy computation. Having a model for the means and variances of the extracted features, we express feature goodness criteria, and develop these into optimal filter design approaches for both two- and multiple texture problems. Earlier the ratio of the feature means has been used as a criterion for two-texture filter design. Here we examine this criterion and develop closed form design techniques for another feature mean criterion. We furthermore develop new closed form design approaches for the Fisher criterion and a generalized criterion on both the feature means and variances, and the assumptions necessary for the solutions to be valid. For the Fisher criterion we also develop and test the necessary model for gradient search optimization. Finally, we develop a scheme for designing an optimal bank of filters for discriminating texture mosaics with more than two textures.

**Keywords:** Image processing, pattern recognition, texture segmentation, filter, optimal, feature extraction.

## 1 Introduction

Most natural surfaces exhibit texture. A texture recognition system will therefore be a natural part of many computer vision systems. Applications of texture analysis span from geophysical surveying through medical diagnosis to document image analysis. Recent discussions on what is and what is not texture may be found in [?, ?].

Several methods for texture discrimination have been proposed, and a recent overview of some of them is given by Tuceryan and Jain [?]. One important group of approaches to texture discrimination is the use of filters together with a filter response energy measure as feature extractors. A typical system setup is illustrated in Fig. 1. In multichannel approaches, one such system is typically used for each channel.

Most commonly the filter, $h(m,n)$, has been selected to be a Gabor filter or a wavelet filter, but other filters have also been employed. Randen and Husøy [?], for instance, suggest a class of computationally efficient critically sampled filter-banks.

*The authors are with Høgskolen i Stavanger, Stavanger, Norway, e-mail t.randen@ieee.org.

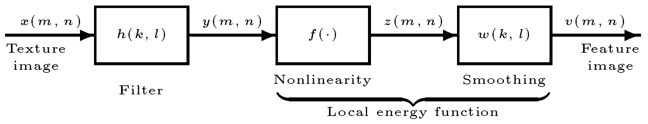


Figure 1

Most approaches to feature extraction with filters use a bank of filters (multichannel filters) which is more or less determined irrespective of the actual textures to be discriminated. Thus, there is no guarantee that the bank of filters will be able to work properly for a specific set of textures. Another unpleasant consequence of most filter-bank approaches is that they are computationally demanding. For these reasons, the design of *optimal* filters for texture discrimination is gaining increased interest.

### 1.1 A survey of optimal filter design approaches

In optimal filter design, the filter impulse response, $h(m,n)$, or some other description of the filter is optimized. In this paper, we start by several two-texture optimization approaches, and when a thorough understanding of that problem is developed, we extend the framework to also incorporate a multiple texture optimal filter-bank design.

Several approaches to the design of Gabor filters optimized with respect to one texture [?, ?], a pair of textures [?, ?], and a set of multiple textures [?, ?] have been presented. Optimization approaches for infinite impulse response (IIR) filters, based on first order all pass filter building blocks have also been presented [?]. However, both the Gabor and the IIR filters are filters with only a few free parameters, making it likely to believe that the true optimum is excluded in most cases.

A general form finite impulse response (FIR) [?] filter generally has many more free parameters than the above approaches. *One*-texture FIR filter optimization, or eigenfilter design, have been presented [?, ?], but design with respect to only one texture does not guarantee any discriminability. In the approaches by Cohen and You [?, ?] filters are optimized with respect to the discriminability using random search. The search is accomplished by random draw of a filter, feature extraction with that filter, computing a criterion function, gradient estimation, semi-random draw of a new filter etc.

Jain and Karu [?] proposes an approach to optimal filter design using multi layer perceptron neural networks [?] for the filtering and local energy operations, and the back propagation training rule for the determination of the optimal parameters for multiple textures. This scheme corresponds to filtering, local energy function, and smoothing [?].

In a broad sense, the approaches by Cohen and You [?, ?] and Jain and Karu [?] are the same, where the latter approach utilizes the back-propagation [?] rule for random search. The major drawback with these approaches is that they do not lead to closed form solutions. Hence, there is no guarantee that a global minimum will be attained. Furthermore, for large filters (which may be necessary for some textures), the complexities are likely to be very large.

Mahalanobis and Singh [?] uses yet another approach, determining the optimal filter by stochastic texture analysis with a closed form solution. A related approach was presented by M. Unser [?], for design of an optimal *transform* for the two texture classification problem. The ideas of Mahalanobis and Singh [?] and Unser [?] will be discussed further in this article.

## 1.2 The feature extraction model

We develop and evaluate several approaches to the design of optimal linear FIR filters, $h(m,n)$, for two-class and subsequently multi-class texture feature extraction, with segmentation and classification in mind. Fig. 1 illustrate the feature extraction model for one texture and one filter. Each texture is processed by that feature extraction system, and if multiple filters are employed, a similar system is applied for each filter. We assume that the nonlinearity and the smoothing operator of Fig. 1, which constitute the local energy function, are constant.

Basically, the purpose of the filter $h(m,n)$ is extraction of frequencies where one or 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 energy.

The energy of a signal, $y$, is defined as $E\left\{y^2\right\}$. However, for segmentation we need a local operator, not a global one as the expectation operator. This can be accomplished by *smoothing* $y^2$. Yet often not explained, this is the basic assumption underlying most texture filtering approaches.

The squaring operator, $y^2$ is not the only viable nonlinearity, and several have been suggested, but Unser [?] found the squaring in conjunction with $\log(\cdot)$ after the smoothing to be the best operator pair for unsupervised segmentation from a set of tested operator pairs. Therefore, and for the sake of ease of modeling, we will let the nonlinear operator be a squaring operator,

$$f(\cdot) = (\cdot)^2. \tag{1}$$

Our multi-texture discrimination approach is based on breaking the problem up into two-class subproblems, where each class may constitute several textures. In our setup, we will assume that the classifier to be applied to the extracted features is a thresholding operator, which will be equivalent to the Bayes classifier [?] if the one-dimensional two-class features are well behaved. Then, applying a second monotonic transformation to the extracted features will have no effect, and the second nonlinearity is consequently omitted.

The smoothing filter, $w(m,n)$, is a unity gain low-pass filter, thus the mean of the output of the filter is identical to the mean of the input. We assume that the textures being filtered are wide sense stationary [?], and that they are well described by their autocorrelation functions, $r_{xxi}(k,l)$.

Based on this model, in Section 2 we develop expressions for estimating the mean, $\mu_{vi}$, and variance, $\sigma_{vi}^2$, of the output from the feature extractor for a given texture, $x_i(m,n)$, and a given filter $h(m,n)$.

## 1.3 Optimization criteria

We will in the following discuss several criteria for describing the quality of an extracted feature variable for the purpose of segmentation and classification. We start with criteria for two-texture problems, and extend to multiple textures.

The first two-texture criterion we will examine was suggested by Singh and Mahalanobis [?]. Their idea was that if the ratio between the means of the extracted features is high, the features are far apart, and the segmentation will be good. Hence, they used the criterion function

$$J_{MS} = \frac{\mu_{v1}}{\mu_{v2}}$$

as feature quality measure, and optimized the feature extraction filter with respect to this. In lack of any other name for this criterion, we will denote it as the *Mahalanobis-Singh criterion.* Their approach will be covered in Section 3.1.

A somewhat similar criterion is

$$J_U(\mathbf{h}) = \frac{(\mu_{v1} - \mu_{v2})^2}{\mu_{v1}\mu_{v2}},$$

suggested by Unser [?] for designing optimal texture transforms. This criterion has not been applied for texture filter design earlier, and a scheme for designing an optimal filter with respect to this criterion is developed in Section 3.2. We will denote this criterion the *Unser criterion.*

The main drawback with both of these criteria 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 for the two textures in mind (see Fig. 7). A criterion well known in the pattern recognition literature which takes the feature variances into account is the *Fisher criterion* [?],

$$J_F(\mathbf{h}) = \frac{(\mu_{v1} - \mu_{v2})^2}{\sigma_{v1}^2 + \sigma_{v2}^2}.$$

We develop the framework for designing filters optimal with respect to this criterion in Section 4. Both a closed form optimization technique, based on simplifying assumptions, and the necessary mathematics for a gradient search optimization are developed.

In Section 5, common characteristics of the closed form two-texture solutions to the different criteria are pointed out, and based on these common characteristics, we formulate a generalized solution.

The two-texture problem is interesting, but the range of applications is rather limited. To extend the usefulness of our feature extraction model and the ideas of the optimization approaches for two-texture cases, we develop a multiple texture optimal filter-bank design scheme in Section 6. We manage to develop a closed form solution based on optimizing with respect to the mean values of the extracted features, and show an example of its application. We also give the necessary expressions for gradient search optimization including either only the mean scatter or even the feature variance in the criterion function.

In summary, the contributions of this paper are

1. A new model for estimating the mean and variance of features extracted using a setup as shown in Fig. 1. The approach by Mahalanobis and Singh [?] only included a model for the mean, and did not include the full system of Fig. 1.

2. Optimal closed form two-texture filter design with respect to the Unser, Fisher, and a generalized criterion, with the development of what conditions are required for these solutions to be valid.

3. Gradient search optimization for the Fisher criterion.

4. Extensions to optimal multiple texture filter-bank design.

# 2 Feature extraction model

In this section we develop estimates of the mean and variance of the extracted features using a feature extraction scheme as illustrated in Fig. 1. Such a model may be used for analyzing given filters, or as we will do later in this article, for designing optimum filters.

## 2.1 Vector formulation

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

$$\begin{aligned} y_i(m,n) &= h(m,n) * x_i(m,n) \\ &= \sum_{k=0}^{M_h-1} \sum_{l=0}^{N_h-1} h(k,l) x_i(m-k, n-l), \end{aligned} \quad (2)$$

where $*$ denotes two-dimensional convolution and $h(m,n)$ is an $M_h \times N_h$ filter. This operation corresponds to the first block of Fig. 1. We define

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

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

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

Then, the output at pixel $(m,n)$ for texture $i$, Eq. (2), may be rewritten as

$$y_i(m,n) = \mathbf{h}^T \mathbf{x}_i(m,n). \quad (3)$$

Similarly, the smoothing operation, which we have defined to be an averaging operation, may be defined as

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

This formulation may easily be extended to filter masks of any shape, but for simplicity, we only consider rectangular masks here.

## 2.2 Feature mean

Using this matrix formulation for the filtering operations, the expected feature value at the output of the filter is defined by (see Fig. 1)

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

Defining $\mathbf{R}_{\mathbf{xx}i}$ as

$$\mathbf{R}_{\mathbf{xx}i} = E\left\{\mathbf{x}_i(m,n)\mathbf{x}_i^T(m,n)\right\}, \quad (4)$$

we get

$$\mu_{vi} = (\mathbf{h}^T \mathbf{R}_{\mathbf{xx}i} \mathbf{h}). \quad (5)$$

By writing the elements of $\mathbf{R}_{\mathbf{xx}i}$ explicitly and assuming wide sense stationarity [?], it is readily seen that $\mathbf{R}_{\mathbf{xx}i}$ is constructed from the two-dimensional autocorrelation function for texture $x_i$.

Since $\mathbf{R}_{\mathbf{xx}i}$, Eq. (4) is symmetric, the derivative is [?]

$$\frac{\partial \mu_{vi}}{\partial \mathbf{h}} = \frac{\partial(\mathbf{h}^T \mathbf{R}_{\mathbf{xx}i}\mathbf{h})}{\partial \mathbf{h}} = 2\mathbf{R}_{\mathbf{xx}i}\mathbf{h}. \quad (6)$$

This derivative will be needed in the optimization approaches of the subsequent sections.

### 2.3 Feature variance

The variance of the feature image, $v_i(m,n)$, is

$$\begin{aligned}\sigma_{v_i}^2 &= E\left\{(v_i(m,n)-\mu_{v_i})^2\right\} \\ &= E\left\{v_i^2(m,n)\right\} - 2E\left\{v_i(m,n)\right\}\mu_{v_i} + {\mu_{v_i}}^2 \\ &= E\left\{\mathbf{w}^T\mathbf{z}_i(m,n)\mathbf{z}_i^T(m,n)\mathbf{w}\right\} - {\mu_{v_i}}^2 \\ &= (\mathbf{w}^T\mathbf{R}_{\mathbf{zz}i}\mathbf{w}) - {\mu_{v_i}}^2, \end{aligned} \tag{7}$$

where

$$\mathbf{R}_{\mathbf{zz}i} = E\left\{\mathbf{z}_i(m,n)\mathbf{z}_i^T(m,n)\right\}. \tag{8}$$

The matrix $\mathbf{R}_{\mathbf{zz}i}$ is readily constructed from the autocorrelation function of the image $z_i(m,n)$ (see Fig. 1), given by

$$\begin{aligned} r_{zzi}(k,l) &= E\left\{z_i(m,n)z_i(m+k,n+l)\right\} \\ &= E\left\{y_i^2(m,n)y_i^2(m+k,n+l)\right\}. \end{aligned} \tag{9}$$

Unfortunately, we have not been able to find any useful variance model including this fourth order moment. However, if we assume that $y_i(m,n)$ is Gaussian, we may write [?]

$$r_{zzi}(k,l) = 2c_{yyi}^2(k,l) + 4c_{yyi}(k,l){\mu_{y_i}}^2 + r_{yyi}^2(0,0), \tag{10}$$

where $c_{yyi}(k,l)$ is the covariance function [?] of $y_i(m,n)$, that is

$$c_{yyi}(k,l) = E\left\{\left(y_i(m,n)-\mu_{y_i}\right)\left(y_i(m+k,n+l)-\mu_{y_i}\right)\right\}$$

which yields

$$c_{yyi}(k,l) = r_{yyi}(k,l) - {\mu_{y_i}}^2,$$

with $r_{yyi}(k,l)$ being the autocorrelation function of $y_i(m,n)$. Hence,

$$\begin{aligned} r_{zzi}(k,l) &= 2\left(r_{yyi}(k,l)-{\mu_{y_i}}^2\right)^2 \\ &+ 4\left(r_{yyi}(k,l)-{\mu_{y_i}}^2\right){\mu_{y_i}}^2 + r_{yyi}^2(0,0) \end{aligned}$$

giving

$$r_{zzi}(k,l) = 2r_{yyi}^2(k,l) - 2{\mu_{y_i}}^4 + {\mu_{v_i}}^2. \tag{11}$$

Furthermore, we have

$$\begin{aligned} r_{yyi}(k,l) &= E\left\{y_i(m,n)y_i(m+k,n+l)\right\} \\ &= E\left\{\mathbf{h}^T\mathbf{x}_i(m,n)\mathbf{x}_i^T(m+k,n+l)\mathbf{h}\right\} \\ &= (\mathbf{h}^T\mathbf{R}_{\mathbf{xx}i}(k,l)\mathbf{h}), \end{aligned} \tag{12}$$

where

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

Note that from Eq. (4), $\mathbf{R}_{\mathbf{xx}i} = \mathbf{R}_{\mathbf{xx}i}(0,0)$. Combining Eqs. (7), (8), (11), and (12) we are able to estimate the variance of the extracted features, $\sigma_{v_i}^2$, given the autocorrelation function of the texture $x_i(m,n)$, the filter, $h(m,n)$, and the smoothing filter $w(m,n)$.

For later use, we note that we may write this more explicitly as

$$\begin{aligned}\sigma_{v_i}^2 &= (\mathbf{w}^T\mathbf{R}_{\mathbf{zz}i}\mathbf{w}) - {\mu_{v_i}}^2 \\ &= \sum_{k_1=0}^{M_w-1}\sum_{l_1=0}^{N_w-1} w(k_1,l_1)\sum_{k_2=0}^{M_w-1}\sum_{l_2=0}^{N_w-1} w(k_2,l_2) \\ &\quad r_{zzi}(k_1-k_2,l_1-l_2) - {\mu_{v_i}}^2 \\ &= 2\sum_{k_1=0}^{M_w-1}\sum_{l_1=0}^{N_w-1} w(k_1,l_1)\sum_{k_2=0}^{M_w-1}\sum_{l_2=0}^{N_w-1} w(k_2,l_2) \\ &\quad 2r_{yyi}^2(k_1-k_2,l_1-l_2) - 2{\mu_{y_i}}^4 + {\mu_{v_i}}^2 - {\mu_{v_i}}^2 \end{aligned}$$

yielding

$$\begin{aligned}\sigma_{v_i}^2 &= 2\sum_{k_1=0}^{M_w-1}\sum_{l_1=0}^{N_w-1} w(k_1,l_1)\sum_{k_2=0}^{M_w-1}\sum_{l_2=0}^{N_w-1} w(k_2,l_2), \\ &\quad 2(\mathbf{h}^T\mathbf{R}_{\mathbf{xx}i}(k_1-k_2,l_1-l_2)\mathbf{h})^2 - 2\left(\mu_{x_i}\mathbf{1}^\mathbf{T}\mathbf{h}\right)^4 \end{aligned} \tag{13}$$

where $\mathbf{1} = [\mathbf{1}\ \mathbf{1}\cdots\mathbf{1}]^\mathbf{T}$ and $w(m,n)$ is the smoothing filter which is of size $M_w \times N_w$.

We will also need the partial derivative of $\sigma_{v_i}^2$ with respect to $\mathbf{h}$ later. From Eq. (13) and the differentiation rules of [?] we get

$$\begin{aligned}\frac{\partial\sigma_{v_i}^2}{\partial\mathbf{h}} &= 8\sum_{k_1=0}^{M_w-1}\sum_{l_1=0}^{N_w-1} w(k_1,l_1)\sum_{k_2=0}^{M_w-1}\sum_{l_2=0}^{N_w-1} w(k_2,l_2) \\ &\quad (\mathbf{h}^T\mathbf{R}_{\mathbf{xx}i}(k,l)\mathbf{h})\left(\mathbf{R}_{\mathbf{xx}i}(k,l)\mathbf{h} + \mathbf{R}_{\mathbf{xx}i}(k,l)^T\mathbf{h}\right) \\ &- 8\left(\mu_{x_i}\mathbf{1}^\mathbf{T}\mathbf{h}\right)^3\mu_{x_i}\mathbf{1}. \end{aligned} \tag{14}$$

### 2.4 Model assessment

When we developed an estimator for the variance above, we made the assumption that the output of the filter, $y_i(m,n)$, was Gaussian. This is an approximation, which is generally not true. However, we tested distributions for several filters and several real world textures from the Brodatz album [?], both by plotting the histograms of $y_i(m,n)$ and by making Q-Q plots [?], and found that the approximation was fair.

How good is the total variance model we have developed? In an experiment, we computed the true and estimated variances for a number of textures with several randomly drawn filters. The experiments indicated a good relationship between estimated and true variance. However, the figures indicated that there might be an approximately linear relationship between the variance-estimation error and the true variance. An example where this is apparent is shown in Fig. 2. In this experiment, we applied several random drawn filters of size $5\times 5$ with $w(m,n)$ as a $9\times 9$ averaging filter to texture D5 [?]. As we see, the ratio between the true and modeled feature variance is approximately constant. We tested a large number of textures, and found apparent linear relations in all cases, but the ratio

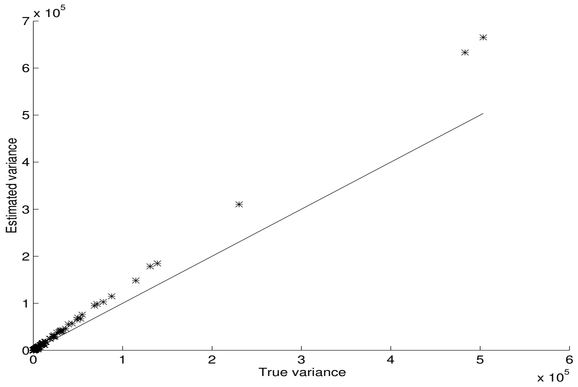


Figure 2

was different for different textures. We have no explanation for this observation.

As this experiment indicates, a possible improvement of the estimated variances may be to run a test as above and estimate the parameters of the linear relationship. The estimates for the variance may then be adjusted accordingly.

From other experiments it was clear that the major part of the estimation error stems from approximating Eq. (9) by Eq. (10).

We did a similar experiment to analyze the estimation of the feature mean, and found an excellent match between estimated and true feature mean value. The error was typically in the range 1-4% for real world textures.

The estimate for the mean value of $y_i(m,n)$, $\mu_{y_i}$ also had an error typically in the same range. However, since the expression for $r_{zzi}(m,n)$ in Eq. (11) contains ${\mu_{y_i}}^4$, i.e. to the fourth order, even a small deviation in the estimate for $\mu_{y_i}$ may yield a considerable error in the estimate of $r_{zzi}(m,n)$, and thus in the estimate of $\sigma_{v_i}^2$. Hence, if the feature variance is small compared to the mean of the input texture, the variance estimate is likely to be unreliable. We do not consider this to be a major limitation, because if the mean value of two homogeneous patterns differ significantly, they are easily discriminated by smoothing, and texture techniques are not necessary. Hence, we may subtract the mean prior to design and filtering in most cases where texture is of interest.

### 2.5 Texture boundary and non-stationarity modeling

The estimate for the mean value of the extracted features given by Eq. (5) was based on the assumption that $x_i(m,n)$ was wide sense stationary (WSS). However, when computing the expected feature value somewhere on the edge between two textures, the WSS assumption is not any more appropriate. By writing the elements of $\mathbf{R_{xx}}_i(k,l)$ explicitly for a given edge location relative to the region of support, it is readily seen that $\mathbf{R_{xx}}_i(k,l)$ may be created from the auto- and cross-correlation functions of the two textures making up the edge. Using this, we can model and compensate for the feature response along edges of any desired orientation and shape. An application of this was shown in [?].

## 3 Mean value optimized features

The obvious aim of an optimal filter for two-texture discrimination is to make the local energies of the filter outputs maximally separable. If it is possible to extract features with mean values far apart, these features are likely to be easily separated, and may be used in segmentation and classification of the texture image.

### 3.1 Mahalanobis-Singh criterion function

Mahalanobis and Singh [?] used the ratio between the expected energies of the filter responses for the two textures as the criterion function for filter optimization, i.e.

$$J_{MS} = \frac{\mu_{v1}}{\mu_{v2}} \tag{15}$$

where $\mu_{v1}$ and $\mu_{v2}$ represent the feature means. We remember that the feature means, $\mu_{vi}$, Eq. (5), are functions of the filter coefficient vector $\mathbf{h}$. Finding the coefficient vector, $\mathbf{h}$, optimal with respect to this criterion entails the solution of

$$\frac{\partial J_{MS}(\mathbf{h})}{\partial \mathbf{h}} = 0.$$

The chain rule for differentiation yields

$$\frac{\partial J_{MS}(\mathbf{h})}{\partial \mathbf{h}} = \frac{1}{\mu_{v2}}\frac{\partial \mu_{v1}}{\partial \mathbf{h}} - \frac{\mu_{v1}}{{\mu_{v2}}^2}\frac{\partial \mu_{v2}}{\partial \mathbf{h}} = 0. \tag{16}$$

Combining Eqs. (5), (6), and (16), we get the eigenvalue equation

$$({\mathbf{R_{xx}}_2}^{-1}\mathbf{R_{xx}}_1)\mathbf{h} = \lambda \mathbf{h}, \tag{17}$$

where

$$\lambda = \frac{(\mathbf{h}^T\mathbf{R_{xx}}_1\mathbf{h})}{(\mathbf{h}^T\mathbf{R_{xx}}_2\mathbf{h})} = \frac{\mu_{v1}}{\mu_{v2}}. \tag{18}$$

The optimal filter coefficient vector is found as the eigenvector of $({\mathbf{R_{xx}}_2}^{-1}\mathbf{R_{xx}}_1)$ with the maximum corresponding criterion, $J_{MS}(\mathbf{h})$. Since the eigenvalue is given by $\lambda = J_{MS}(\mathbf{h})$, this means that the optimal filter coefficient vector is the eigenvector corresponding to the maximum eigenvalue.

Are we guaranteed that all eigen-solutions will satisfy Eq. (18), i.e. that all eigen-solutions are valid solutions? Since $(\mathbf{h}^T\mathbf{R_{xx}}_2\mathbf{h})$ is a scalar, $\lambda$ is unmodified by dividing and multiplying by $(\mathbf{h}^T\mathbf{R_{xx}}_2\mathbf{h})$, i.e.

$$\lambda = \frac{(\mathbf{h}^T\mathbf{R_{xx}}_2\mathbf{h})\lambda}{(\mathbf{h}^T\mathbf{R_{xx}}_2\mathbf{h})} = \frac{\mathbf{h}\mathbf{R_{xx}}_2\lambda\mathbf{h}}{(\mathbf{h}^T\mathbf{R_{xx}}_2\mathbf{h})}.$$

Combining this with (17), we get

$$\lambda = \frac{\mathbf{h}^T\mathbf{R_{xx}}_2({\mathbf{R_{xx}}_2}^{-1}\mathbf{R_{xx}}_1)\mathbf{h}}{(\mathbf{h}^T\mathbf{R_{xx}}_2\mathbf{h})} = \frac{(\mathbf{h}^T\mathbf{R_{xx}}_1\mathbf{h})}{(\mathbf{h}^T\mathbf{R_{xx}}_2\mathbf{h})},$$

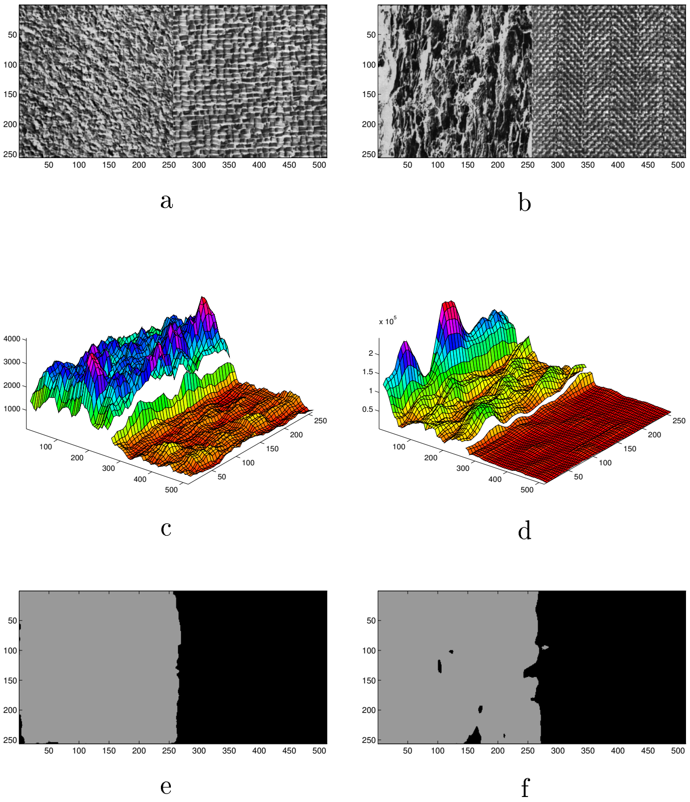

Figure 3

which proves that all eigenvectors, $\mathbf{h}$, satisfy (18), i.e. they are valid solutions.

We applied filters optimized by this technique to several textures, and the results for some of them are shown in Fig. 3. Note that in the figure, the feature surface was subsampled and the texture border was made discontinuous for display purposes. We classified the extracted features by thresholding, and the results of this is also shown in Fig. 3. In all experiments we designed an $M_h \times N_h = 7 \times 7$ filter, $h(m,n)$, letting the smoothing filter, $w(m,n)$, be an $M_w \times N_w = 31 \times 31$ averaging filter. Note that the size of the smoothing filter is a tradeoff between feature variations and edge accuracy. For the test images in this article, a smoothing filter of size $31 \times 31$ was found to be adequate.

The major conceptual drawback with this method is the fact that the variances are not included in the criterion. Thus the optimal filter may yield large feature mean separation, but with large feature variances, so that the density functions for the features for the two textures have a considerable overlap.

As we might expect, this is reflected in the performance of the filters for some texture pairs. Two examples of texture pairs where the energy separations are non-satisfactory are shown in Fig. 4. In these examples, this method yields features with too high variances and consequently a considerable feature overlap. We can clearly see that successful segmentation and classification was not achieved. A his-

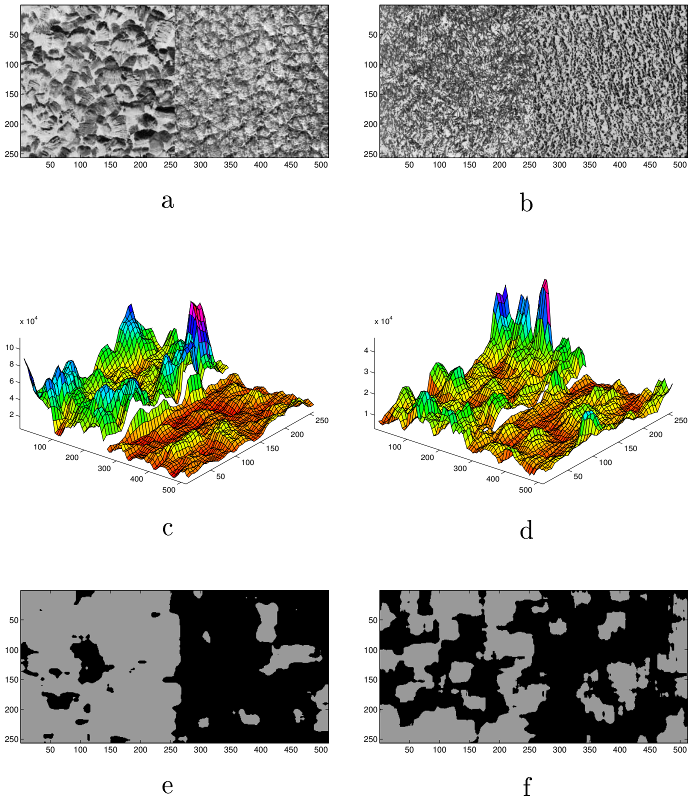

Figure 4

togram for one of these texture pairs, D5-D92, is shown in Fig. 7(a), depicting the overlap.

### 3.2 Unser criterion function

As we saw in the experiments in the previous subsection, the criterion suggested by Mahalanobis and Singh [?] is not suited for all texture pairs. It is therefore natural to search for other criteria, and hope that they overcome the noted difficulties.

Unser [?] suggested the squared difference of the feature means, normalized by the product of the mean values as criterion for optimal texture *transform* design,

$$J_U(\mathbf{h}) = \frac{(\mu_{v1} - \mu_{v2})^2}{\mu_{v1}\mu_{v2}}. \tag{19}$$

Since a texture transform corresponds to a sub-sampled filter-bank, we can not apply the design technique suggested by Unser [?] directly for *filter* design. However, filter optimization with respect to $J_U$ is easily developed. Equating the partial derivative of $J_U(\mathbf{h})$ with zero yields

$$(\mathbf{R_{xx}}_2^{-1}\mathbf{R_{xx}}_1)\mathbf{h} = \lambda\mathbf{h}, \tag{20}$$

where again

$$\lambda = \frac{\mu_{v1}}{\mu_{v2}}.$$

We identify Eq. (20) as the same eigen-problem as in the previous approach.

In the present approach as well, we choose the eigenvector yielding maximum criterion value as our filter coefficient vector. Since $J_U = \lambda + \lambda^{-1} - 2$, selection of the optimum eigenvector is done by analyzing the corresponding eigenvalues. Note that since in general $J_U(\mathbf{h}) \neq J_{MS}(\mathbf{h})$, this approach may lead to the selection of a *different eigenvector* than the previous approach.

Since the eigenproblem is identical to the one expressed in Eqs. (17) and (18), the proof of existence of all solutions we gave in the previous subsection is valid in this case also.

As with the previous criterion, also this criterion neglects the feature variances, with the potential risk of designing filters giving considerable feature overlap. However, in a large number of experiments, the filters were generally found to perform better than filters designed with the Mahalanobis-Singh criterion.

Some examples of extracted features and resulting classifications are shown in Fig. 5. We see that the result for texture D9-D24 is more satisfactory than it was in the previous approach. The only plausible explanation we have for this is obtained by rewriting the criterion as $J_U(\mathbf{h}) = \mu_{v1}/\mu_{v2} + \mu_{v2}/\mu_{v1} - 2$. Even if it is difficult designing a filter yielding high features for texture 1 and low for texture 2, e.g. if the general energy of texture 1 is generally low compared to texture 2, the Mahalanobis-Singh criterion will tend to select a filter emphasizing texture 1, while the Unser criterion, due to the second term of the rewritten criterion, will tend to select a filter emphasizing texture 2.

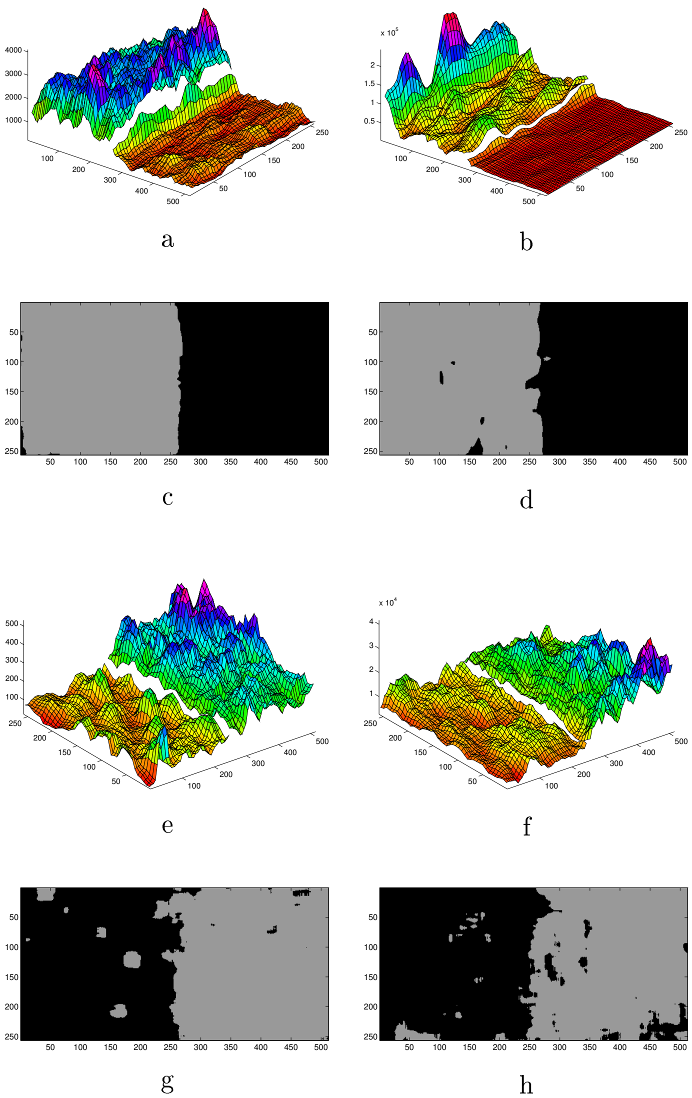


Figure 5

# 4 Fisher criterion optimized features

As we have already suggested, 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 [?],

$$J_F(\mathbf{h}) = \frac{(\mu_{v1} - \mu_{v2})^2}{\sigma_{v1}^2 + \sigma_{v2}^2}, \tag{21}$$

which is well known in pattern recognition.

As with the previous approaches, in order to find the optimal filter, we equate the partial derivative of the criterion with respect to the filter coefficient vector, $\partial J_F(\mathbf{h})/\partial\mathbf{h}$, to zero,

$$\begin{aligned} &\frac{2(\mu_{v1} - \mu_{v2})\left(\frac{\partial\mu_{v1}}{\partial\mathbf{h}} - \frac{\partial\mu_{v2}}{\partial\mathbf{h}}\right)(\sigma_{v1}^2 + \sigma_{v2}^2)}{(\sigma_{v1}^2 + \sigma_{v2}^2)^2} \\ &- \frac{(\mu_{v1} - \mu_{v2})^2\left(\frac{\partial\sigma_{v1}^2}{\partial\mathbf{h}} + \frac{\partial\sigma_{v2}^2}{\partial\mathbf{h}}\right)}{(\sigma_{v1}^2 + \sigma_{v2}^2)^2} = 0. \end{aligned} \tag{22}$$

However, by inserting the expressions for the mean and the variance and their derivatives, Eqs. (5), (6), (13), and (14), we have not been able to find a general closed form solution.

Of course, since we know the derivative, we may use gradient search for optimization. However, such an approach is both time-consuming and likely to converge to a local optimum. Tests with random drawn start vectors indicated that this is indeed the case. Another option is the use of a random search technique such as simulated annealing or combinations of simulated annealing and genetic algorithms, as done in [?]. However, in order to find the global minimum by such a technique, the computation time may be infinitely long. Hence, it is desirable to search for some kind of closed form solution for this criterion.

## 4.1 Approximate variance expression

From Section 2.3, Eq. (13) we know that it is possible to write the variance as

$$\begin{aligned} \sigma_{vi}^2 &= \sum_{k_1=0}^{M_w-1}\sum_{l_1=0}^{N_w-1} w(k_1,l_1) \sum_{k_2=0}^{M_w-1}\sum_{l_2=0}^{N_w-1} w(k_2,l_2) \\ &\quad r_{zzi}(k_1-k_2, l_1-l_2) - {\mu_{v}}_i^2. \end{aligned}$$

If we approximate $r_{zzi}(k,l)$ by $\hat{r}_{zzi}(k,l)$, where

$$\hat{r}_{zzi}(k,l) = \rho_{i1}\hat{r}_{zzi}(k-1,l) \quad \forall\ k > 0,$$

$$\hat{r}_{zzi}(k,l) = \rho_{i2}\hat{r}_{zzi}(k,l-1) \quad \forall\ l > 0,$$

and $\hat{r}_{zzi}(0,0) = r_{zzi}(0,0)$, we may write

$$r_{zzi}(k,l) \approx \hat{r}_{zzi}(k,l) = \rho_{i1}^{|k|}\rho_{i2}^{|l|} r_{zzi}(0,0). \tag{23}$$

Hence

$$\begin{aligned} \sigma_{vi}^2 &\approx \sum_{k_1=0}^{M_w-1}\sum_{l_1=0}^{N_w-1} w(k_1,l_1) \sum_{k_2=0}^{M_w-1}\sum_{l_2=0}^{N_w-1} w(k_2,l_2) \\ &\quad \rho_{i1}^{|k_1-k_2|}\rho_{i2}^{|l_1-l_2|} r_{zzi}(0,0) - {\mu_{v}}_i^2 \\ &= r_{zzi}(0,0) \sum_{k_1=0}^{M_w-1}\sum_{l_1=0}^{N_w-1} w(k_1,l_1) \sum_{k_2=0}^{M_w-1}\sum_{l_2=0}^{N_w-1} w(k_2,l_2) \\ &\quad \rho_{i1}^{|k_1-k_2|}\rho_{i2}^{|l_1-l_2|} - {\mu_{v}}_i^2. \end{aligned}$$

To keep expressions manageable we define

$$\rho_i = \sum_{k_1=0}^{M_w-1}\sum_{l_1=0}^{N_w-1} w(k_1,l_1) \sum_{k_2=0}^{M_w-1}\sum_{l_2=0}^{N_w-1} w(k_2,l_2)\rho_{i1}^{|k_1-k_2|}\rho_{i2}^{|l_1-l_2|}. \tag{24}$$

Hence $\sigma_{vi}^2 \approx \rho_i r_{zzi}(0,0) - {\mu_{v}}_i^2$. Applying (11) and assuming zero mean input, we get

$$\begin{aligned} \sigma_{vi}^2 &\approx 2\rho_i r_{yyi}^2(0,0) + r_{yyi}^2(0,0) - {\mu_{v}}_i^2 \\ &= 2\rho_i(\mathbf{h}^T\mathbf{R}_{\mathbf{xx}i}\mathbf{h})^2 + (\mathbf{h}^T\mathbf{R}_{\mathbf{xx}i}\mathbf{h})^2 - (\mathbf{h}^T\mathbf{R}_{\mathbf{xx}i}\mathbf{h})^2 \\ &= 2\rho_i(\mathbf{h}^T\mathbf{R}_{\mathbf{xx}i}\mathbf{h})^2. \end{aligned} \tag{25}$$

To be able to optimize, we need the partial derivative of this approximation of $\sigma_{vi}^2$ with respect to $\mathbf{h}$. As we see, the approximation of $\sigma_{vi}^2$ consists of two elements, $(\mathbf{h}^T\mathbf{R}_{\mathbf{xx}i}\mathbf{h})^2$ and $\rho_i$ which are both functions of $\mathbf{h}$. We tested the relative perturbations of $(\mathbf{h}^T\mathbf{R}_{\mathbf{xx}i}\mathbf{h})$ and $\rho_i$ as functions of small perturbations of $\mathbf{h}$. These tests indicated that the relative perturbations of $(\mathbf{h}^T\mathbf{R}_{\mathbf{xx}i}\mathbf{h})$ were generally much larger than the relative perturbations of $\rho_i$. Hence, approximating $\rho_{ij}$ by a constant will only perturbate the derivative of $\sigma_{vi}^2$ minimally. Hence, we get the approximation

$$\frac{\partial\sigma_{vi}^2}{\partial\mathbf{h}} \approx 8\rho_i(\mathbf{h}^T\mathbf{R}_{\mathbf{xx}i}\mathbf{h})\mathbf{R}_{\mathbf{xx}i}\mathbf{h}. \tag{26}$$

We note that if the image $z_i(m,n)$ is separable [?] and autoregressive [?] of order one the approximation in Eq. (23) becomes exact, i.e. $r_{zzi}(k,l) = \hat{r}_{zzi}(k,l),\ \forall k,l$. However, generally making these approximations (23) makes the estimate of $r_{zzi}(k,l)$, and in consequence, the estimate of $\sigma_{vi}^2$ less accurate.

## 4.2 Closed form solution

In order to find the optimal filter, we need to solve Eq. (22), yielding

$$\begin{aligned} &2(\sigma_{v1}^2 + \sigma_{v2}^2)\frac{\partial\mu_{v1}}{\partial\mathbf{h}} - (\mu_{v1} - \mu_{v2})\frac{\partial\sigma_{v1}^2}{\partial\mathbf{h}} \\ &= 2(\sigma_{v1}^2 + \sigma_{v2}^2)\frac{\partial\mu_{v2}}{\partial\mathbf{h}} + (\mu_{v1} - \mu_{v2})\frac{\partial\sigma_{v2}^2}{\partial\mathbf{h}}. \end{aligned} \tag{27}$$

Approximating the variances and their derivatives, Eqs. (25) and (26), inserting into (27), and simplifying yields

$$\left(\rho_2(\mathbf{h}^T\mathbf{R_{xx}}_2\mathbf{h})^2 + \rho_1(\mathbf{h}^T\mathbf{R_{xx}}_1\mathbf{h})(\mathbf{h}^T\mathbf{R_{xx}}_2\mathbf{h})\right)\mathbf{R_{xx}}_1\mathbf{h}$$
$$= \left(\rho_1(\mathbf{h}^T\mathbf{R_{xx}}_1\mathbf{h})^2 + \rho_2(\mathbf{h}^T\mathbf{R_{xx}}_1\mathbf{h})(\mathbf{h}^T\mathbf{R_{xx}}_2\mathbf{h})\right)\mathbf{R_{xx}}_2\mathbf{h},$$

which furthermore may be simplified to

$$\mathbf{R_{xx}}_2^{-1}\mathbf{R_{xx}}_1\mathbf{h} = \frac{(\mathbf{h}^T\mathbf{R_{xx}}_1\mathbf{h})}{(\mathbf{h}^T\mathbf{R_{xx}}_2\mathbf{h})}\mathbf{h}. \tag{28}$$

We note that this equation is identical to the eigenproblems, $\mathbf{R_{xx}}_2^{-1}\mathbf{R_{xx}}_1\mathbf{h} = \lambda\mathbf{h}$, in the previous approaches, (17) and (20). In Section 3.1 we proved that all eigenvalue/vector pairs satisfy $\lambda = (\mathbf{h}^T\mathbf{R_{xx}}_1\mathbf{h})/(\mathbf{h}^T\mathbf{R_{xx}}_2\mathbf{h})$, hence all eigenvectors are valid solutions.

In summary, the optimal filter coefficient vector is designed by getting the eigenvectors of $\mathbf{R_{xx}}_2^{-1}\mathbf{R_{xx}}_1$, and selecting the one maximizing the criterion $J_F(\mathbf{h})$.

With any of these optimization approaches, if for some reason, the optimal filter does not yield satisfying results, it is possible to add features by selecting the next optimal filters, yielding not a single filter, but a filter-bank. Due to the orthogonality of the eigenvectors, this is likely to increase feature separation.

## 4.3 Iterative solutions

Another way for improving the results, is to let the closed form solution form the initial filter for a gradient or random search, hence alleviating all the approximations we had to make to get a closed form solution. In this paper we give some results from gradient search. For results from random search, please refer to [?].

For the textures in this paper, the gradient search tended to increase the criterion value by a factor of about two, most of it in the first few iterations.

Gradient search is typically a time consuming operation, especially when estimating the gradient is as complex as here, Eq. (14). We note that the complexity of the variance estimation is much dependent on the size of the smoothing filter, $w(m, n)$. Hence, the gradient search is faster for smaller smoothing filters. If design time is critical, it can be useful to be aware that the performance difference between filters optimized with gradient search modeling small and large smoothing filters are typically much smaller than the performance difference between a gradient search optimized filter and the closed form solution.

## 4.4 Experimental results

We have performed tests on a large number of texture pairs, and show some of the results in this article. As we can see from Fig. 6, the extracted features for the texture pairs treated earlier in this article are generally sufficiently well separated, and that was also the conclusion of our test on the large number of textures. Also the texture pairs that caused problems for the method of Section 3.1 are well separated. In order to display the feature separation of the four

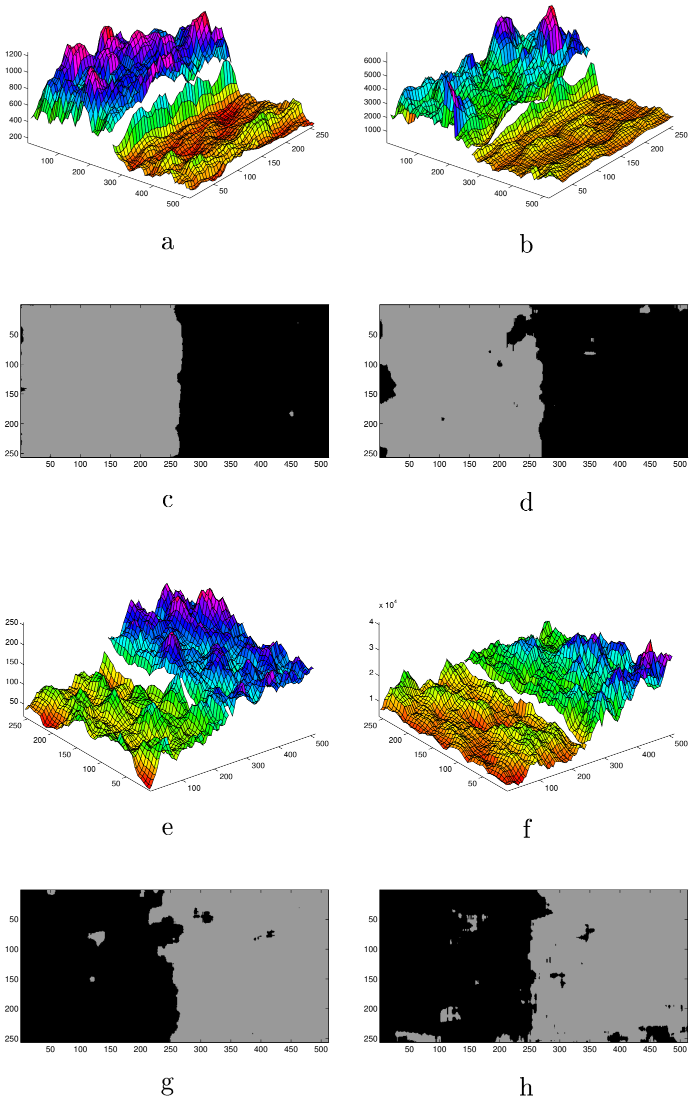

Figure 6

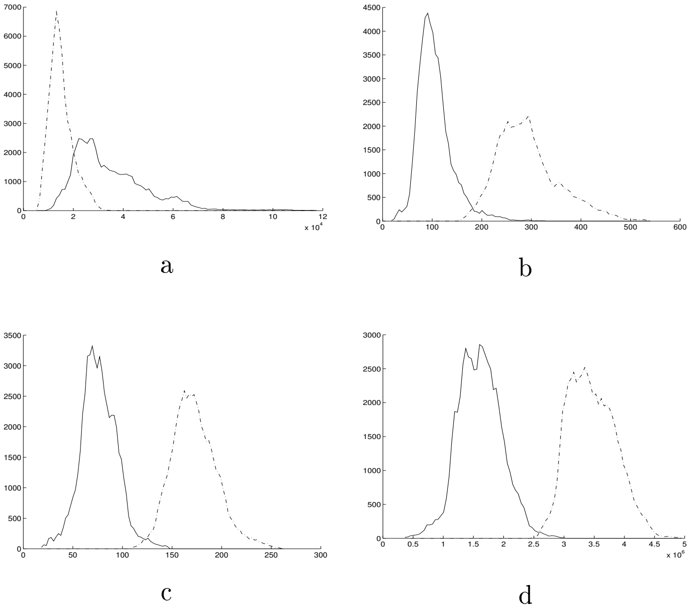


Figure 7

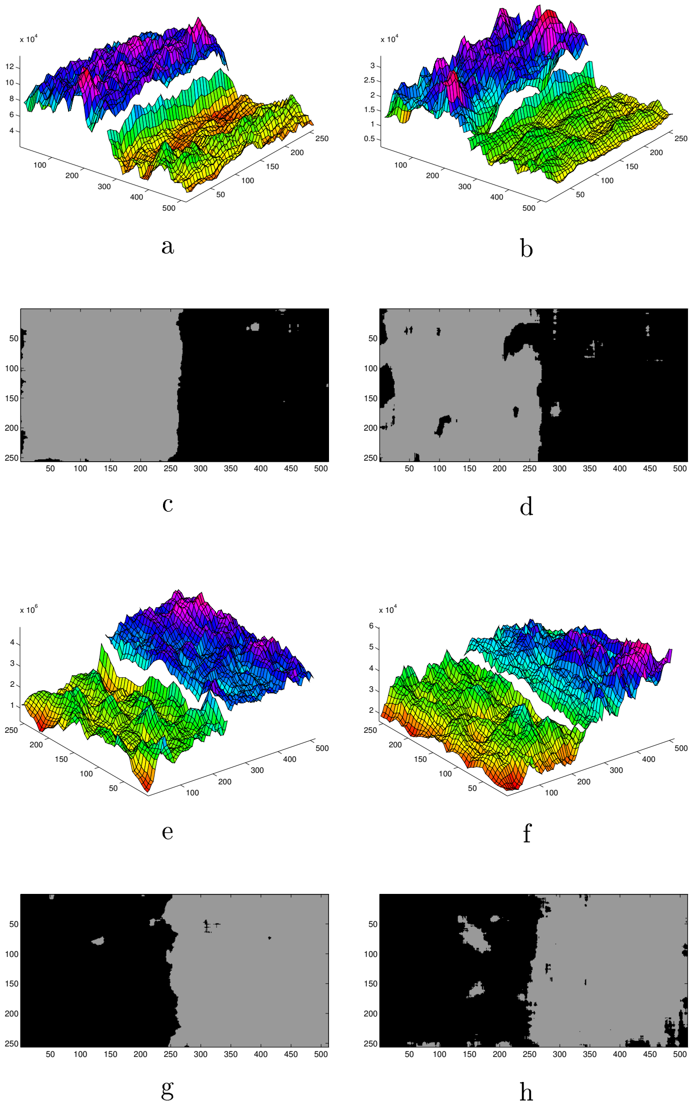


Figure 9

techniques examined so far, the histograms for the features for both textures of the texture pair D5-D92 are displayed in Fig. 7. As we can see, the Fisher criterion optimized filters yield features with less variances and better feature separation.

To illustrate what an optimal filter looks like, we give an example of an amplitude response in Fig. 8. Of course, $7 \times 7$ filter coefficients do not allow for much detail in the amplitude response. However, as we would intuitively expect, the frequency properties corresponds to some sort of a comb filter, emphasizing frequencies which discriminate the two textures well.

However, not even this criterion is ideal for all cases, as is illustrated in Figs. 3(d) and 6(b). We see that the features extracted according to the $J_{MS}$ criterion are better separated than the features extracted according to the $J_F$ criterion. The reason for this is obvious, the features in Fig. 3(c) corresponding to highest feature mean has a considerable variance. However, since this variance do not lead to any significant feature overlap, the extracted features are efficient. Hence, low variance is not a guarantee for low error rate.

The same textures have also been applied to filters optimized with gradient search using the closed form solution as starting point for the search. The results from this approach are shown in Fig. 9.

The gradient search overall doubled the estimated Fisher criterion value from that of the closed form solution for the texture pairs in this paper. However, due to the generally marginal classification error, no significant improvement was observed on the classification error. In some

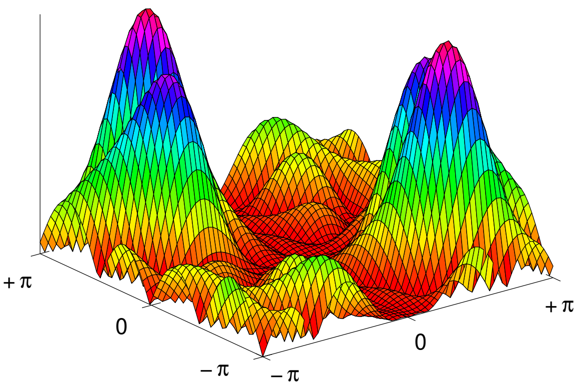


Figure 8

cases the classification accuracy was even reduced mainly due to increased inaccuracy at the texture edges (for which we do not optimize). See Sections 2.5 and 6.3 for further discussions on edge responses.

Both approaches involving the Fisher criterion show far better overall performance for the textures we tested than the approach based on the Mahalanobis-Singh criterion. It is worth noting that some of the texture pairs in this paper were the most difficult texture-pairs selected from a larger set of texture pairs.

## 5 Unified approach

As we have seen, all the closed form solutions presented so far yield the same eigen-problem, $({\mathbf{R}_{\mathbf{xx}2}}^{-1}\mathbf{R}_{\mathbf{xx}1})\mathbf{h} = \lambda\mathbf{h}$, where the optimal filter coefficient vector is the eigenvector yielding maximum criterion value. Different eigenvectors are, however, chosen as the optimal filters with the different approaches. Hence, it is likely to assume that this eigen-problem is not the solution to only these three criterion functions, but to a broader class of criteria.

This is indeed the case. Consider the generalized criterion function

$$J = \frac{\nu_1 {\mu_{v1}}^2 + \nu_2 \mu_{v1}\mu_{v2} + \nu_3 {\mu_{v2}}^2 + \nu_4 \sigma_{v1}^2 + \nu_5 \sigma_{v2}^2}{\xi_1 {\mu_{v1}}^2 + \xi_2 \mu_{v1}\mu_{v2} + \xi_3 {\mu_{v2}}^2 + \xi_4 \sigma_{v1}^2 + \xi_5 \sigma_{v2}^2}\mathbf{h}, \quad (29)$$

where the $\nu_i$s and $\xi_i$s are arbitrary scalar constants. All the criterion functions we have considered thus far fit into this formulation. Given the same assumptions and following the same steps as in Section 4.2, the closed form optimization yields

$${\mathbf{R}_{\mathbf{xx}2}}^{-1}\mathbf{R}_{\mathbf{xx}1}\mathbf{h} = \frac{(\mathbf{h}^T\mathbf{R}_{\mathbf{xx}1}\mathbf{h})}{(\mathbf{h}^T\mathbf{R}_{\mathbf{xx}2}\mathbf{h})}\mathbf{h}. \quad (30)$$

The proof is straightforward, but tedious. We have shown that the above expression has valid solutions for all the eigenvectors in Section 3.1.

We also note that if we omit the variance terms, the only assumption we have to make to obtain the closed form solution is that the outputs, $y_i(m,n)$, of the filter $h(m,n)$ are Gaussian.

If there is doubt on which of several object functions that fit into the framework of Eq. (29) is best for a given application, one possible solution is to apply all filters corresponding to the eigenvectors of ${\mathbf{R}_{\mathbf{xx}2}}^{-1}\mathbf{R}_{\mathbf{xx}1}$ and select the one yielding maximum separation of the histograms of the extracted features. This approach was applied to the textures shown earlier in this papers, and some of the results are shown in Fig. 10. The classification with this technique was, as we might expect, very good. The feature histogram for the texture D5-D92 is shown in Fig. 11. By comparison, we see that, as we might expect, the histogram is not worse separated than any of the histograms in Fig. 7.

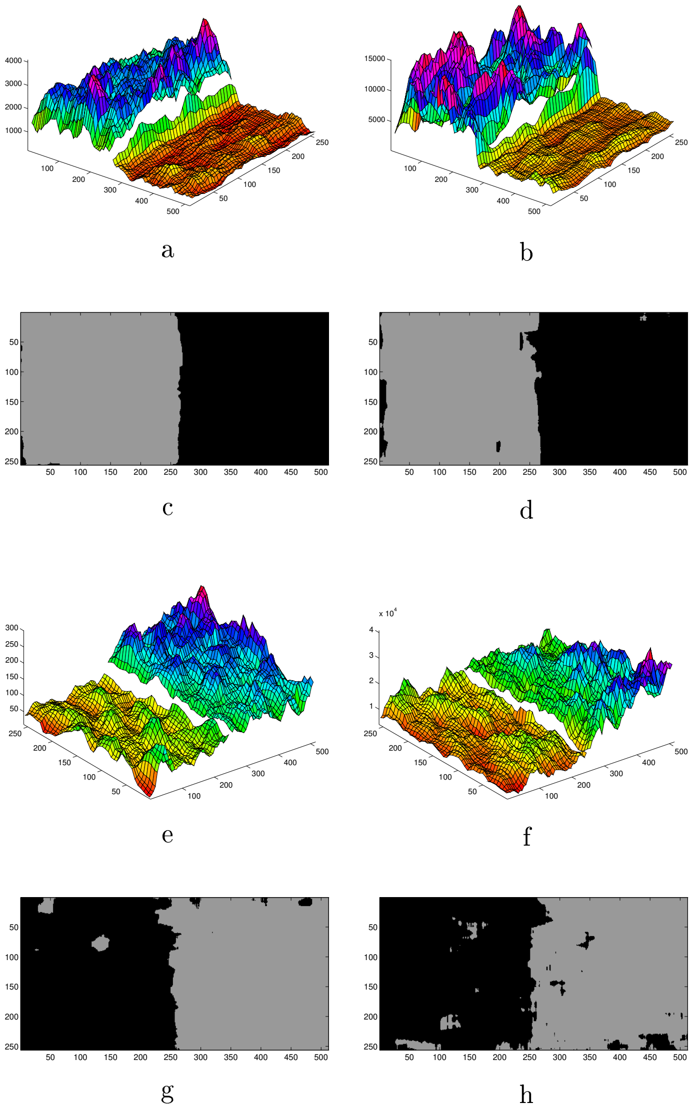


Figure 10

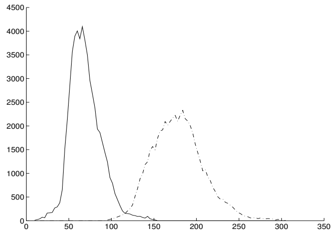

Figure 11

# 6 Multi texture optimal filter-bank design

In the preceding sections, we have seen some approaches to the design of optimal filters for the discrimination of two textures. This may be interesting in an inspection application for discriminating normal from abnormal, as in cancer detection [?]. However, the majority of texture segmentation and classification problems typically involve more than two textures. Hence, extensions to multiple texture cases are desirable. To have reliable features for more than two textures, we generally believe that more than one filter is needed. In this section, we will be dealing with the design and application of multiple filters for multiple texture problems.

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

Building on the two-texture approaches, we suggest a more or less hierarchical feature extraction scheme, where each filter separate one group of textures from another. The problems that needs to be solved are

1. How to design a filter for optimal feature separation between two groups of textures.
2. How to quantify the similarity of the textures.
3. How to group the textures into groups allowing robust feature extractors to be constructed.
4. Given a set of filters, how to discriminate the textures.

In the following, we will try to give answers to these issues.

Due to the fundamental importance, we will start by addressing the last issue, issue 4, first. We illustrate how this is solved by an example. Assume we have four textures in a problem, $x_1$, $x_2$, $x_3$, and $x_4$. Assume further we have three filters, capable of separating textures $x_1$, $x_2$ from $x_3$, $x_1$, $x_2$, $x_3$ from $x_4$, and $x_1, x_3$ from $x_2$, respectively. Then, texture $x_1$ will be the only texture belonging to the first group from all filters, texture $x_4$ will be the only belonging to the last group from the second filter etc.

## 6.1 Optimal texture-group filters

We recall from Eq. (5) that the mean value of the features extracted from texture $x_i(m,n)$ by filter $h(m,n)$ can be written as $\mu_{vi} = (\mathbf{h}^T\mathbf{R}_{\mathbf{xx}i}\mathbf{h})$. 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

$$\begin{aligned}\mu_v &= \frac{1}{I}\left(\mu_{v1} + \cdots + \mu_{vI}\right) \\ &= \frac{1}{I}\left((\mathbf{h}^T\mathbf{R}_{\mathbf{xx}1}\mathbf{h}) + \cdots + (\mathbf{h}^T\mathbf{R}_{\mathbf{xx}I}\mathbf{h})\right) \\ &= \mathbf{h}^T\frac{1}{I}\left(\mathbf{R}_{\mathbf{xx}1} + \cdots + \mathbf{R}_{\mathbf{xx}I}\right)\mathbf{h}. \end{aligned} \tag{31}$$

This expression is identical to the expression for the mean of one texture if we replace the matrix $\mathbf{R}_{\mathbf{xx}i}$ for that texture with the average $\mathbf{R}_{\mathbf{xx}i}$ of all the textures in the group. Letting the average $\mathbf{R}_{\mathbf{xx}i}$ for each group of textures replace the $\mathbf{R}_{\mathbf{xx}1}$ and $\mathbf{R}_{\mathbf{xx}2}$ in the feature mean optimization approaches, Sections 3.1 and 3.2, it is readily seen that the same filter design approaches may be used, including closed form solution.

The overall variance of the features for the group of textures is

$$\sigma_v{}^2 = E\left\{(v(m,n) - \mu_v)^2\right\},$$

which may be rewritten as

$$\sigma_v{}^2 = E\left\{\frac{1}{I}\left((v_1(m,n) - \mu_v)^2 + \cdots + (v_I(m,n) - \mu_v)^2\right)\right\},$$

and

$$\begin{aligned}\sigma_v{}^2 &= E\left\{\frac{1}{I}\left((v_1(m,n) - \mu_{v1} + \mu_{v1} - \mu_v)^2 + \cdots\right.\right. \\ &+ \left.\left.(v_I(m,n) - \mu_{vI} + \mu_{vI} - \mu_v)^2\right)\right\},\end{aligned}$$

yielding

$$\begin{aligned}\sigma_v{}^2 &= \frac{1}{I}\left(\sigma_{v1}^2 + \cdots + \sigma_{vI}^2\right) + \frac{1}{I}\left((\mu_{v1} - \mu_v)^2 + \cdots\right. \\ &+ \left.(\mu_{vI} - \mu_v)^2\right).\end{aligned} \tag{32}$$

Unfortunately, we have not been able to separate $\mathbf{h}$'s and the $\mathbf{R}_{\mathbf{xx}i}$'s (even with the assumptions we made in Section 4.1) in this expression as we did for the mean value. Hence, we have not been able to find a closed form solution for the Fisher criterion for multiple textures. Still we may, of course, use gradient search, starting for instance with one of the mean-optimized closed form solutions.

From Eqs. (13) and (14) we understand that computing the variance and its derivative is computationally complex for large filters $h(m,n)$ and $w(m,n)$. However, computing the mean and its derivative is much less complex. Hence, a gradient search solution may be obtained significantly faster by approximating the variance by the mean scatter of Eq. (32), $\frac{1}{I}\left((\mu_{v1} - \mu_v)^2 + \cdots + (\mu_{vI} - \mu_v)^2\right)$. None of these approaches have been thoroughly tested.

## 6.2 Texture similarity and grouping

If a filter is expected to separate one group of textures well from another, it is reasonable to assume that the textures in one of the groups all must share some characteristics which are less prevalent in the textures of the other group. In order to make such a grouping, some measure of the similarity, i.e. the degree of shared properties, is needed. It can in general not be expected that very dissimilar textures may be grouped with success.

Of course, filters corresponding to all possible groupings of the textures may be designed and the expected feature statistics computed for each texture to determine which grouping is best. However, if there are many textures at a specific level, such an approach soon becomes exhaustive, e.g. with 16 textures to be divided into two groups, there are $2^{15} - 1 = 32767$ possible meaningful groupings. Hence, another way of determining the groupings are necessary.

Several texture measures for describing the similarity between textures have been proposed in the literature, and in principle most global measures may be applied to group textures in this application. Specifically we tested an approach describing the texture by visual characteristics as coarseness, contrast, directionality, line-likeness, regularity, and roughness as suggested by Tamura et al. [?]. Another approach may be to derive measures describing the surface of the autocorrelation function. This is similar to the objective for the measures describing the cooccurrence matrix [?], and some of these measures were implemented and applied to describe the autocorrelation surface. We also attempted describing the autocorrelation surface by several different moments of scan-lines in different directions. Common to all these approaches was the fact that the resemblance between the measures and the success of grouping textures left much to be desired.

A measure which has a closer tie to elements in the filter design may be a good candidate. Inherent in the closed form optimal filter design is the eigenvectors and eigenvalues of the matrix ${\mathbf{R}_{\mathbf{xx}_2}}^{-1}\mathbf{R}_{\mathbf{xx}_1}$, Equations (17), (20), (28), and (30). It is known that if textures $x_1(m,n)$ and $x_2(m,n)$ are identical we will have $\mathbf{R}_{\mathbf{xx}_2} = \mathbf{R}_{\mathbf{xx}_1}$ and ${\mathbf{R}_{\mathbf{xx}_2}}^{-1}\mathbf{R}_{\mathbf{xx}_1} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. We know that all eigenvalues of the identity matrix are identical and equal to one. Hence, it may be assumed that the more $x_1(m,n)$ and $x_2(m,n)$ are different, the more ${\mathbf{R}_{\mathbf{xx}_2}}^{-1}\mathbf{R}_{\mathbf{xx}_1}$ will differ from the identity matrix, and the more the eigenvalues will be dissimilar. Hence, a measure of the dissimilarity, or spread, of these eigenvalues may be a suitable measure for the dissimilarity of two textures. One function expressing the dissimilarity of the eigenvalues, $\lambda_i$, is the arithmetic to geometric mean,

$$d_{i,j} = \frac{\frac{1}{N}\sum_{i=1}^{N}\lambda_i}{\left(\prod_{i=1}^{N}\lambda_i\right)^{\frac{1}{N}}}, \tag{33}$$

of the eigenvalues of ${\mathbf{R}_{\mathbf{xx}_i}}^{-1}\mathbf{R}_{\mathbf{xx}_j}$, as suggested in [?]. Not formally a *metric* [?], a number of experiments on texture-group filter design and on texture cell clustering [?] have given us confidence that this measure is very useful in determining the similarity between two textures.

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

$$\begin{aligned} \max\{d_{i_k,i_l}\} &< \min\{d_{i_p,j_q}\} \;\; \forall k,l,p < I,\; q < J \\ &\wedge \\ \max\{d_{j_k,j_l}\} &< \min\{d_{j_p,i_q}\} \;\; \forall k,l,p < J,\; q < I. \end{aligned} \tag{34}$$

We did some attempts with groupings where one or several relations broke the condition above, and the results were poor in many cases.

A disadvantage by this measure, with the above condition, is that the textures are fragmented into a lot of groups, since typically only small groups satisfy the condition (34). To be able to separate two groups, one filter is needed, and with many groups the number of group pairs will be large, hence a large number of filters must be applied.

However, extracting features with the large number of groups following from condition (34) in some experiments clearly indicated that it would be possible to combine several of the groups. Since this measure does not give any good indication as to which groups may be combined, we finally settled for a scheme involving designing filters for several group combinations, and testing the overall feature separation (by evaluating the means for the features for each texture). Since this is a more definitive measure than any of the others, we also applied this measure to assess the goodness of the initial groups. The ideas suggested here are detailed in Algorithm 1.

## 6.3 Experimental results

An example of the results of Algorithm 1 for texture grouping along with the Unser criterion based filter design for the textures D4, D9, D19, D21, D24, D28, D29, D36, D37, and D38 [?], and the original image and the classification is shown in Fig. 12.

In this particular example, the texture grouping yielded nine texture group pairs, i.e. nine filters were designed. Note that due to the more or less hierarchical nature of the solution, a smart filtering approach may recognize that the regions of an image belonging to one group, need only be filtered by a subset of the total number of filters, thus reducing the effective number of filters needed.

In an artificial mosaic of textures like the one in Fig. 12, the edges between textures are very discontinuous, i.e. having large high-frequency contents. Depending on the textures, the feature extraction filter may very well have high-pass characteristics, thus having a boosted response at the edge, instead of a smooth transition between the texture regions. In combination with the smoothing filter, this implies reduced edge accuracy, an effect which can clearly be seen on the classification result. Texture edges in real world images are likely to be much smoother. Hence, such a

1. Compute distance measure (33) between all pairs of textures.
2. Start with two most distant textures as seeds for a new pair of texture groups.
3. Examine each of the rest of the textures and add it to one of the groups if condition (34) is fulfilled. If adding a texture to a group do not increase the total discrimination (see 9.), do not add it.
4. Design a filter intended for separating the two groups. If there's less than 20% separation of the modeled feature means, remove one of the textures being closest to the other group (other minimum distance margins than 20% may of course be selected, but 20% was found to be appropriate in our experiments). Of course, since we have a model for the feature variance for a known texture and a known filter, analysis of the variance could also have been incorporated into this step.
5. Continue 4 until 20% separation is obtained, or only one texture is left in each group.
6. For all remaining not yet grouped textures:
   (a) Start with the texture which is most distant from any of the two groups as a seed.
   (b) For each non-discriminated texture, add it to this group, if it increases discrimination and condition (34) is satisfied.
7. Try to add this group to the two others by designing filters and performing the test of 4. If the test is passed, add this group to the corresponding group.
8. Repeat steps 6 and 7 until all textures have been examined.
9. Mark off (in a binary table) the discrimination the resulting grouping yields, e.g. if texture $x_i$ is in group one and texture $x_j$ is in group two, element $(i, j)$ of the table is marked as discriminated.
10. With all remaining non-separated textures, go to 2.

**Alg. 1.** Algorithm for constructing texture group pairs.

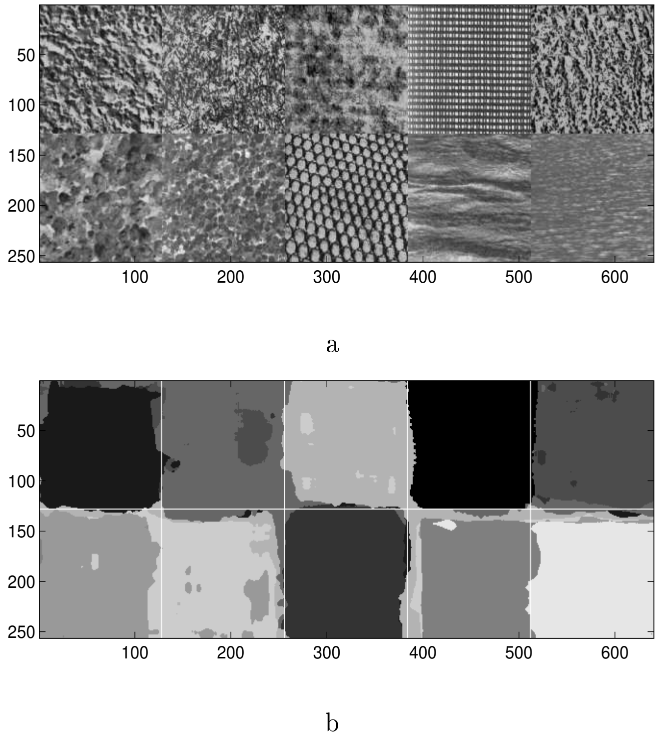


Figure 12

boosted response will probably not occur to a great extent in a real world mosaic.

However, if strongly discontinuous edges are present, it can either be alleviated by edge modeling, Section 2.5, and compensation, as in [?], by neighborhood examination [?], or by applying smaller filters and/or applying smaller smoothing windows in the regions near the edges. In order to keep this paper at a manageable size, we do neither detail nor show any examples of this here.

# 7 Summary and conclusions

In this paper, we have developed estimators for the mean and variance of texture features extracted with a feature filter, a quadratic nonlinearity, and a smoothing filter. Based on the mathematical models for these estimators, we then developed filter optimization procedures for different criterion functions for both two- and multi-texture problems.

To find a closed form solution to the two-texture optimization problems incorporating the variance, we had to make some approximations in the model. For multi-texture problems, the only closed form solutions we obtained were solutions only incorporating the feature means. Gradient or random searches are also possible using the estimators we have developed.

The two-texture design techniques were thoroughly tested on several kinds of textures. The performance of the design technique based on the Unser-criterion was satisfactory in most cases, but the Fisher-criterion based techniques were generally better. Best performance was, however, achieved with the technique resulting from the unified approach. However our effort in designing filters maximizing the feature goodness criteria discussed herein, 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.

One important issue that is not addressed in this paper is 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. If for instance the filter is too small compared to the texture elements of the image, no good results can be expected.

Generally, compared to using fixed filter-banks as the Gabor filter, wavelet filters etc. the main disadvantages with an optimal approach is the need for better insight, possibly complex design, and in some cases increased sensitivity to non-stationarity (in a multichannel approach using a supervised neural net or similar classifier as in [**?**, **?**], non-stationarities may be compensated by the classifier). The main advantages are the optimality (important for very similar textures), fewer filters, and much simpler classification (using one feature at a time thresholding is far less complex than large feature vector neural nets or statistic classifiers). Furthermore, compared to the commonly used Gabor filters, as in [**?**], successful results are obtained with significantly smaller, i.e. faster, filters.

# 8 Acknowledgment

The authors would like to thank Dr. Anil K. Jain, Michigan State University for useful ideas on the extension to multiple texture classes.

This work was partially supported by Norwegian Research Council grant 110914/410.