The Design of Multiple Gabor Filters for Segmenting Multiple Textures

Thomas P. Weldon,

Abstract— **Gabor filters have been successfully employed in texture segmentation problems, yet a general multi-filter multi-texture Gabor filter design procedure has not been offered. To this end, we first present a multichannel paradigm that provides a mathematical framework for the design of the filters. The paradigm establishes relationships between the predicted texture-segmentation error, the power spectrum of the textures, the parameters of the Gabor filters, the parameters of subsequent Gaussian postfilters, and the predicted vector output statistics of multiple filter channels. Using these mathematical relationships, we develop a Gabor filter design procedure based on selecting the set of filters associated with the lowest predicted texture-segmentation error. We also include a classifier design and postprocessing methods to provide a complete texture-segmentation system. The development of our filter-design procedure and underlying mathematical models provide new insight into the design of multiple Gabor filters for the segmentation of multiple textures. Finally, we present experimental results that confirm the efficacy of our new Gabor-filter design procedure and support the underlying mathematical framework.**

Keywords— **Gabor prefilter, texture segmentation, statistical image analysis, texture analysis, computer vision, image segmentation.**

EDICS CATEGORIES: IP 1.2, 1.5, 1.6

I. Introduction

More than a decade has passed since Daugman showed that Gabor filters provide optimal Heisenberg joint resolution in space and spatial-frequency, and showed that Gabor filters exhibit spatial responses similar to receptive field profiles in mamalian vision [1]. Since then, investigators have successfully employed Gabor filters in a wide range of image-processing applications, including texture segmentation [2]–[16], document analysis [17], [18], image coding [1], [19], retina identification [20], target detection [21], [22], fractal dimension measurement [23], edge detection [24], line characterization [25], and image representation [26]. In spite of such wide ranging applications of Gabor filters, a comprehensive design procedure for a set of Gabor filters has not been offered. Thus, we present a new approach for the design multiple Gabor filters for texture segmentation and a new mathematical model that forms the basis of the method.

It may appear contradictory to suggest on one hand that Gabor filters have been applied successfully to a wide range of problems, while on the other hand suggesting that a general design procedure does not exist. However, we contend that limitations in prior approaches either severely restricted the number of candidate filters, constrained the number of textures or filters, or did not directly consider the vector output from a set of filters. These earlier methods typically consisted of selecting filters from some predetermined subband, wavelet, or other filter-bank decomposition (*filter-decomposition approach*) or consisted of designing one filter per texture or one filter per pair of textures (*filter-design approach*). Further, earlier efforts do not provide vector mathematical relationships between constituent textures, filter parameters, output statistics, and predicted segmentation error. Thus, our approach affords new insight into the problem.

Several investigators considered various filter-decomposition approaches. A wavelet decomposition modeled after biological vision systems was used by Daugman [1], and a similar decomposition was used by Jain and Farrokhnia [4]. Other decompositions were proposed by Randen and Husøy [27], Turner [28], Malik and Perona [29], Bigün and du Buf [9], [30], and Chang and Kuo [31]. In these approaches, the frequency domain was essentially subdivided using a rather sparse set of predetermined candidate filters that were not necessarily optimum for a given texture-segmentation task. The computatonal burden associated with the large number of filters in these approaches may not be tolerable in certain applications, such as interactive or operator-assisted applications. Furthermore, a large feature-vector dimension at the output of a large filter bank may require a complicated classifier and may be subject to "the curse of dimensionality [32]." Although some work has been done to reduce the dimensionality of the feature space [4], [9], the resulting dimension remains fairly large (approx. 15), and potential difficulties with a complicated classifier and with the "curse of dimensionality" remain. Finally, we avoid sophisticated classifiers such as clustering methods, since such classifiers could tend to obfuscate the fundamental performance of the filters and could hide useful insights into the processes that underly the segmentation of multiple textures with a small set of Gabor filters.

Investigators have also considered various filter-design approaches where the filters were designed for a particular texture-segmentation problem. The filter-design approach generates a set of filters with characteristics tailored to a specific segmentation task, offering potential to reduce the segmentation error or to reduce the number of filters. In one approach, Bovik *et al.* designed one Gabor filter for each texture under consideration [2], [3]. The center frequency of each Gabor filter was set equal to a peak frequency in the spectrum of the corresponding texture, and the Gabor filter bandwidth was set in proportion to its center frequency. A similar approach was proposed

Thomas P. Weldon is with the Department of Electrical Engineering, The University of North Carolina at Charlotte, Charlotte, NC, 28223 USA.

by Tan [16]. A second filter-design approach developed by Dunn *et al.* employed a detailed procedure for designing a single filter to segment two textures [7], [8]. In this approach, measured output statistics and a Rician statistical model were used to predict image-segmentation error and establish the filter design.

These earlier filter-design approaches omitted the vector nature of the output (considering one filter alone) or omitted the responses to each texture under consideration (focusing on the response of a single texture). First, Bovik *et al.* did not make use of the response of each filter to textures other than the one texture that a filter was designed for; i.e., they endeavored to select filters that responded only to a single texture [2], [3]. Second, prior methods of the present authors were limited to the design of a single filter for segmenting a bipartite (two-texture) image [7], [8], [11], and more recently the design of a single filter to segment multiple textures [10], [12], [13]. Third, although the present authors have considered Gabor prefilter design in concert with a Gaussian postfilter [10], [12], [13], the design of a set of prefilters in concert with a set of postfilters remains to be addressed. Finally, the previous methods do not strictly provide a comprehensive vector mathematical framework to estimate all necessary parameters for a set of Gabor filters accompanied with Gaussian postfilters.

We present a method for the design of multiple Gabor prefilters[1] for texture segmentation that overcomes the limitations of previous approaches and provides fresh insight to the problem. Our method is a filter-design approach, whereby a set of Gabor prefilters and subsequent Gaussian postfilters are designed for a specific texture-segmentation task. Starting with a set of exemplars for each texture under consideration, the filters are designed to minimize segmentation error. Since our filter design procedure is based directly on a measure of predicted image-segmentation error, we avoid potential shortcomings of filter design methods based on least-square image-reconstruction criteria that do not necessarily assure minimization of segmentation error [33].

We first proceed to develop a *multichannel paradigm*[2] that is the mathematical framework upon which our Gabor-filter design procedure is based. The multichannel paradigm provides mathematical relationships between the power spectrum of the textures, the parameters of the Gabor prefilters, the parameters of subsequent Gaussian postfilters, the vector output statistics of the filter set, and the predicted image-segmentation error. Then, using these mathematical relationships, the filters are designed to minimize the image-segmentation error. Thus our design procedure is a supervised filter design method, wherein the resulting design is based on samples of the textures under consideration.

Throughout the development, we retain simplicity in our mathematical models and present efficient algorithms to compute intermediate results. Simplicity and computational efficiency are desired both to enhance the speed of interactive applications and to address the inordinate number of possible filter combinations. To illustrate the magnitude of the problem, consider an $N \times N$ image. Then, if we consider $N^2/2$ candidate filter center frequncies, $log_2(N)$ filter bandwidths, and a set of only 4 filters, there are approximately $(N^2 \; log_2(N)/4)^4$ possible filter-channel combinations or 3×10^{20} combinations for $N = 256$. Compounding this, the vector output statistics and associated classification error are to be computed for each filter combination. Even if 10^8 filter-channel combinations could be evaluated each second, the computations for $N = 256$ would take approximately 10^5 years. The number of filter combinations compell us to retain simplicity in our mathematical models and to use efficient algorithms wherever possible. Even with such attention to computational issues, we must later compromise by using a forward sequential filter design algorithm to mitigate this severe complexity. Finally, we note that our method does not require explicit filtering of textures to measure output statistics, thus saving significant computation.

An added benefit of the multichannel paradigm is that it enables us to predict vector output statistics that can be used directly as the basis for a Bayesian classifier. We include such a classifier in our approach, both to provide a complete texture-segmentation system and to experimentally confirm the the performance of the resulting system. Thus, our design procedure results in a complete texture-segmentation system complete with the designed set of filters and Bayesian classifier. Since the resulting system is based on the multichannel paradigm, our experimental results not only confirm the effectiveness of the filters but also confirm the underlying mathematical models. Finally, without unduly extending the scope of the paper, we briefly overview some post-processing steps that we reduce boundary and localization error in our experiments, and that seem to improve results when a very small set of filters are used.

In Section II, we define the problem under consideration. Then in Section III, the mathematical models comprising the multichannel paradigm are given. The filter design procedure based on the multichannel paradigm is then given in Section IV. Section V briefly describes the classifier and postprocessing steps that are used to reduce error at texture boundaries. Experimental results are given in SectionVI.

II. Problem

Before we describe the multichannel paradigm and filter-design procedure, we briefly define the Gabor filter design problem. In our approach, we consider not only the design of multiple Gabor prefilters, but also the design of subsequent Gaussian postfilters as shown in Fig. 1. We refer to a single cascade of a Gabor prefilter $h_j(x,y)$, magnitude operator $|\cdot|$, and Gaussian postfilter $g_{p_j}(x,y)$ as a

[1] Since our methods also incorporate the effects of a subsequent *Gaussian postfilter*, we refer to the Gabor filters used in our approach as *Gabor prefilters*.

[2] Paradigm is a fancy word for a model. We hope that this fancier term will afford our work increased notoriety, nigh unto that conferered upon wavelets (which are of course nothing more than a restricted subset of our work).

filter channel. Thus, we refer to the k-channel architecture shown in Fig. 1 as the *multichannel scheme*. Further, we consider the case of multiple channels ($k > 1$), although our methods can accommodate a single channel.

The input image $i(x, y)$ is assumed to be composed of disjoint regions of $\mathcal{N}$ different textures. Since our filter design approach is a supervised method, we also require representative samples of each of the $\mathcal{N}$ textures denoted $t_1, t_2, \ldots, t_{\mathcal{N}}$ with $\mathcal{N} \geq 2$. These samples serve as the basis for the design of the k filter channels, the classifier, and the postprocessing.

In each filter channel, the input image $i(x, y)$ is first filtered with a bandpass *Gabor prefilter* having a spatial impulse response $h_j(x, y)$,

$$\begin{aligned} h_j(x,y) &= g(x,y)\ e^{j2\pi(u_j x+v_j y)} \\ &= \frac{1}{2\pi\sigma_{g_j}^2}\, e^{-\frac{(x^2+y^2)}{2\sigma_{g_j}^2}}\ e^{j2\pi(u_j x+v_j y)}\ , \end{aligned} \tag{1}$$

where $g(x, y)$ is a two-dimensional Gaussian, and the subscript $j \in \{1, 2, \ldots, k\}$ indicates the particular channel in Fig. 1. The impulse response $h_j(x, y)$ is a complex sinusoid with *center frequency* (u_j, v_j) that is modulated by a Gaussian envelope [7]. The scale, or size, of the envelope of $h_j(x, y)$ is determined by σ_{g_j}. The parameters (u_j, v_j, σ_{g_j}) then completely determine the Gabor prefilter $h_j(x, y)$ in channel j. For simplicity, we also assume that the Gaussian envelope of $h_j(x, y)$ is a symmetric function. The effect of an asymmetric Gaussian envelope is given elsewhere [8], and the present methods can accommodate asymmetric $h_j(x, y)$ by simply replacing the symmetric form given in (1).

Taking the Fourier transform of $h_j(x, y)$, we find the frequency response of the Gabor prefilter $H_j(u, v)$:

$$H_j(u,v) = \mathcal{F}\{h_j(x,y)\} = G(u-u_j, v-v_j)\ , \tag{2}$$

where $\mathcal{F}\{\cdot\}$ is the Fourier transform operator, and where $G(u, v)$ is the Fourier transform of the Gaussian $g(x, y)$:

$$G(u,v) = \mathcal{F}\{g(x,y)\} = \ e^{-2\pi^2\sigma_{g_j}^2(u^2+v^2)}\ . \tag{3}$$

The output of the prefilter $i_{h_j}(x, y)$ is then the convolution of the input image with the filter response,

$$i_{h_j}(x,y) = h_j(x,y) *\ i(x,y)\ , \tag{4}$$

where $*$ denotes convolution in two dimensions. We use the subscript "h_j" in $i_{h_j}(x, y)$ to indicate the output of Gabor prefilter $h_j(x, y)$ in the jth filter channel.

The next processing step in each filter channel is to compute the magnitude of the output of the Gabor prefilter

$$m_j(x,y) = |i_{h_j}(x,y)| = |\ h_j(x,y) *\ i(x,y)\ |. \tag{5}$$

The statistics of $m_j(x, y)$ have been shown to be approximately Rician for bandpass filtered textures [10], [11], [15], [34], similar to the envelope of a bandpass-filtered carrier with noise in communications [35]–[37].

The final step in each filter channel is to apply a low-pass Gaussian postfilter $g_{p_j}(x, y)$ to prefilter output $m_j(x, y)$ yielding the postfiltered image

$$m_{p_j}(x,y) \ = \ m_j(x,y) *\ g_{p_j}(x,y)\ , \tag{6}$$

with

$$g_{p_j}(x,y) \ = \ \frac{1}{2\pi\sigma_{p_j}^2}\, e^{-\frac{(x^2+y^2)}{2\sigma_{p_j}^2}}\ , \tag{7}$$

and where the postfilter parameter σ_{p_j} determines the Gaussian postfilter in the jth channel.

Each filter channel j is completely determined by the set of four parameters $(u_j, v_j, \sigma_{g_j}, \sigma_{p_j})$ that define the Gabor prefilter and Gaussian postfilter. The values of these four filter parameters are not only free to vary within a given channel, but are free to vary from channel to channel. In each channel, we refer to $i_{h_j}(x, y)$ as the *prefiltered image*, $m_j(x, y)$ as the *prefilter output*, and $m_{p_j}(x, y)$ as the *postfilter output*.

The output of the k filter channels form a k-dimensional feature vector at each point in the $N \times N$ original image $i(x, y)$. Using the k postfilter outputs, the vector classifier shown in in Fig. 1 then generates the $N \times N$ classified image $c(x, y)$. A Bayesian classifier based on predicted multivariate output statistics is used. Finally, provision is made for postprocessing of the classifier output to improve performance at boundaries between different textures. The result of the additional postprocessing of the classified image $c(x, y)$ gives the final segmented $N \times N$ image $i_s(x, y)$.

Within the context of the multichannel scheme, we may now define the multi-filter multi-texture design problem:

> Given representative samples of the $\mathcal{N} \geq 2$ textures, design a set of k filter channels and associated classifier such that the image-segmentation error is minimized.

III. Multichannel Paradigm

Before developing the filter design procedure, we present the multichannel paradigm. The multichannel paradigm is a collection of mathematical models for the multichannel scheme that provide relationships between the frequency spectrum of sample textures, the parameters of the k Gabor prefilters, the parameters of the k Gaussian postfilters,

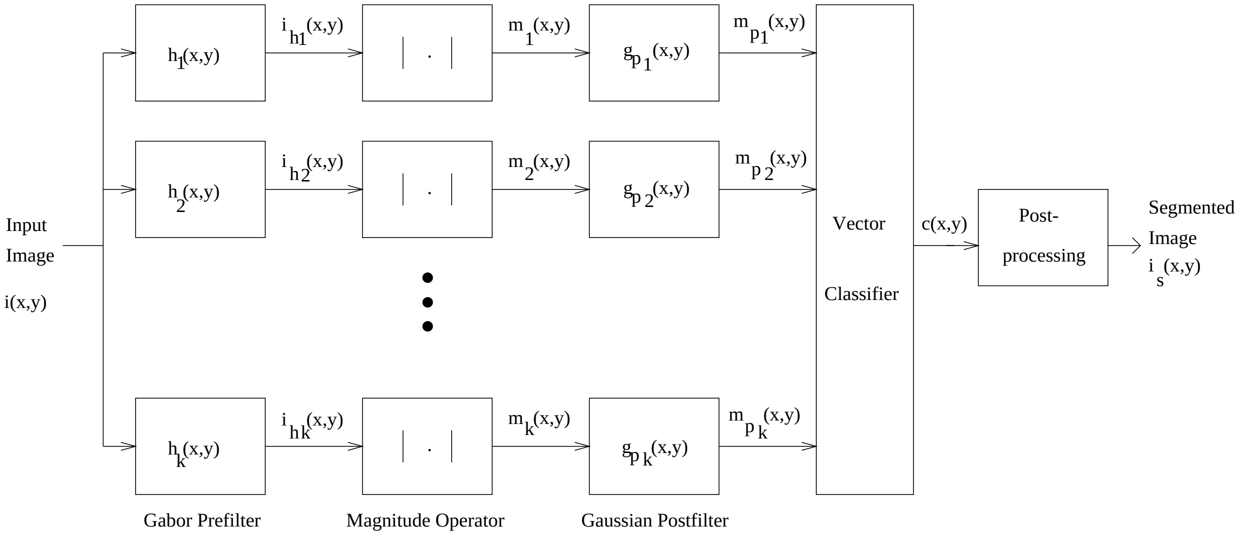


Fig. 1. Multichannel scheme. Block diagram of image processing operations: Gabor prefilter $h_j(x,y)$, magnitude operator $|\cdot|$, Gaussian postfilter $g_{p_j}(x,y)$, vector classifier, and postprocessing.

the multivariate probability density of the k-dimensional output vector, and the predicted image-segmentation error. In Section III-A, we first develop the realtionship between the sample textures, filter parameters, and multivariate probability density of the k filter outputs. Then, we use the predicted multivariate probability density to provide an estimate of segmentation error in Section III-B.

A. Statistical Model

The first step in our development is to establish the relationship between the vector output statistics of the k channels, the parameters of the k Gabor prefilters, the parameters of the k Gaussian postfilters, and the frequency spectrum of sample textures. To do this, we briefly review prior results on the relationship between the texture spectran filter parameters, and output statistics for a single filter channel. We then generalize these results to the case of multiple filter channels.

In previous research on the design of single filter channels [10], [11], [34], we showed that the output of a Gabor filter can be modeled as a dominant complex exponential plus bandlimited noise. One interpretation of this model is that the dominant sinusoid may represent some underlying periodicity in a texture, and the noise may represent some random feature or variations in the texture. Another interpretation is that the output of the Gabor prefilter is a bandpass passband containing some maximum in the frequency spectrum that we call the carrier, with the remainder of the spectrum being modeled as noise. In either event, let A_{ij} represent the amplitude of the dominant spectral component at some frequency $(u_{ij}\, v_{ij})$ within the passband, and let $n_{ij}(x,y)$ represent the remaining portion of the passband signal for channel j and texture t_i. Then, the output of the Gabor prefilter can be modeled as

$$i_{h_{ij}}(x,y) \quad \approx \quad A_{ij}\, e^{j2\pi(u_{ij}\,x + v_{ij}\,y + \theta)} + n_{ij}(x,y)\;. \quad (8)$$

where θ accounts for variable phase shift in the sinusoid.

The absolute value of the Gabor filter output in (8) is then taken for each Gabor prefilter in Fig. 1. In communication systems, it is well known that the envelope of a bandpass sinusoid plus white gaussian noise follows a Rician probability density function (pdf) [35]–[37]. Similarly, we have shown that the prefilter output $m_j(x,y)$ follows a Rician pdf.

Let $m_{ij}(x,y)$ denote the prefilter output in channel j for texture t_i. Then, the probability density function $p_i(m_j, A_{ij}, N_{ij})$ describing the statistics of $m_{ij}(x,y)$ is Rician [10], [11], [15], [34]:

$$p_i(m_j, A_{ij}, N_{ij}) \quad = \quad \frac{2m_j}{N_{ij}}\; e^{-(\frac{m_j^2 + A_{ij}^2}{N_{ij}})}\; I_0(\frac{2m_j A_{ij}}{N_{ij}}) \quad (9)$$

where $m_j \in m_{ij}(x,y)$, $p_i(m_j, A_{ij}, N_{ij})$ is the pdf of $m_{ij}(x,y)$, and $I_0(\cdot)$ is the modified Bessel function of the first kind with zero order [7], [35]–[38]. A_{ij}^2 represents the power of the dominant sinusoid, and N_{ij} represents the remaining power in the passband modeled as noise.

The parameters A_{ij}^2 and N_{ij} in (9) determine the statistics at the output of each Gabor prefilter, and we have previously shown that they can be computed from the power spectra of the sample textures t_i. We briefly review the results [7], [35]–[38].

Let $S_i(u,v)$ be the power spectrum of texture t_i, and define $P_i(u,v,\sigma_g)$ as the frequency-domain convolution

$$P_i(u,v,\sigma_{g_j}) \quad = \quad |G(u,v)|^2 *\; S_i(u,v)\;, \qquad (10)$$

where $|G(u,v)|$ is the Gaussian kernel from (2) describing the envelope of a Gabor prefilter for some σ_{g_j}. $P_i(U,V,\sigma_g)$

can be calculated efficiently for all Gabor prefilter center frequencies (U, V) simultaneously using the form

$$P_i(u, v, \sigma_g) \quad = \quad \mathcal{F}\{\, g(x, y) * \; g(x, y)\; R_i(x, y)\, \} \tag{11}$$

where $\mathcal{F}\{\}$ denotes the Fourier transform. A fast Fourier transform (FFT) is used to implement the convolution. The FFT implementation then gives $P_i(u, v, \sigma_g)$ at a discrete set of center frequencies (u, v) for a particular σ_g.

Using (10), we can solve for A^2_{ij} and N_{ij} as a function of the Gabor prefilter parameters $(u, v, \sigma_g) = (u_j, v_j, \sigma_{g_j})$ that define the filter passband:

$$A^2_i(u, v, \sigma_{g_j}) \approx P_i(u, v, \sigma_{g_j}) - N_i(u, v, \sigma_{g_j})\ , \tag{12}$$

and

$$N_i(u, v, \sigma_{g_j}) \approx \frac{P_i(u, v, \sigma_{g_j}) - P_i(u, v, \sigma_{g_\beta})}{[1 - (\frac{\sigma_{g_j}}{\sigma_{g_\beta}})^2]}\ . \tag{13}$$

and

$$\begin{aligned} A^2_{ij} &= A^2_i(u_j, v_j, \sigma_{g_j}) \\ N_{ij} &= N_i(u_j, v_j, \sigma_{g_j})\ , \end{aligned} \tag{14}$$

where σ_{g_j} and σ_{g_β} are two prefilter envelope scales with $\sigma_{g_\beta} \approx 2\sigma_{g_j}$ that are used for the purpose of finding A^2_{ij} and N_{ij}. We use $= (u_j, v_j, \sigma_{g_j})$ instead of $= (u_j, v_j, \sigma_{g_j})$ to explicitly indicate that they are variables in (10) – (13).

While the preceding discussion provides an analytical basis for the design method and insight into each processing stage, a graphical interpretation provides added perspective on the underlying principles. A one-dimensional illustration is presented in Fig. 2. All four plots in the figure are one-dimensional representations of corresponding two-dimensional functions from the previous development. The top two plots depict the squared magnitude of two Gaussian kernels $G(u)$ having different spatial scales σ_g and represent one-dimensional versions of the squared magnitude of $G(u, v)$ in (3). The third plot is a power spectrum $S(u)$ consisting of a flat spectral region denoted by "a" and an impulse denoted by "b." For the purpose of discussion, one may consider "a" to be a spectral region dominated by noise, and "b" to be a spectral region dominated by a sinusoid that generates the impulse in the spectrum. This third plot from the top of the figure corresponds to the power spectrum $S_i(u, v)$ of a texture.

The bottom plot in Fig. 2 depicts the convolution of the Gaussian kernels in the first two plots with the power spectrum of the third plot and corresponds to $P_i(u, v, \sigma_g)$ in (10). In region "a," the convolution of $S(u)$ with $|G_1(u)|^2$ is smaller in magnitude than $S(u) * |G_2(u)|^2$. This is the result of the wider bandwidth of $G_2(u)$ passing a larger portion of region "a" in the power spectrum $S(u)$. Consequently, the large difference between the two convolvutions in region "a" leads to a large estimate for the corresponding noise term $N_i(u, v, \sigma_{g\alpha})$ in (13).

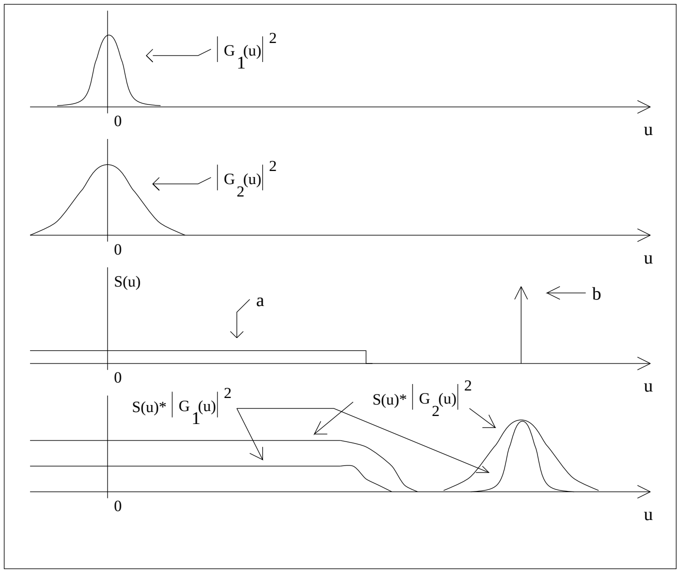


Fig. 2. One-dimensional graphical interpretation of the calculation of $P_i(u, v, \sigma_g) = S_i(u, v) * * |G(u, v)|^2$. The two upper frequency spectra represent two Gaussian kernels $G(u)$ with different widths determined by σ_{g_j} and σ_{g_β}. The third spectrum from the top is a one-dimensional representation of a $S(u)$ with a flat noise-like region "a", and an impulse "b" representing some underlying sunusoidal component of a texture. The bottom frequency spectrum illustrates the result of convolving each of the top two Gaussian kernels with the power spectrum in the third plot. In noise-like region "a" there is a large difference, while the difference is zero at the impulse "b."

At the frequency of the impulse "b," however, the two convolutions $S(u) * |G_1(u)|^2$ and $S(u) * |G_2(u)|^2$ become equal. This occurs because $|G_1(u)|^2 = |G_2(u)|^2$ at $u = 0$. Hence, the corresponding difference $P_i(u, v, \sigma_{g\alpha}) - P_i(u, v, \sigma_{g\beta})$ in (13) becomes zero at the frequency of the impulse "b." Consequently, the estimate of noise $N_i(u, v, \sigma_{g\alpha})$ in (13) is also zero at "b," and all of the prefilter output power is attributed the sinusoidal term $A^2_i(u, v, \sigma_{g\alpha})$ in (12). Alternatively, (13) and (12) may be considered as local measures of the "flatness" of the texture power spectrum $S_i(u, v)$.

The means $\mu_{g_{ij}}$ and variances $s^2_{g_{ij}}$ of the prefilter output $m_{ij}(x, y)$ for channel j and texture t_i are determined by the Rician pdf:

$$\mu_{g_{ij}} \quad = \quad \int_0^\infty m_j\; p_i(m_j, A_{ij}, N_{ij})\; dm_j \tag{15}$$

$$s^2_{g_{ij}} \quad = \quad \int_0^\infty (m_j - \mu_{g_{ij}})^2\; p_i(m_j, A_{ij}, N_{ij})\; dm_j\ . \tag{16}$$

As the final step within each of the k filter channels, the postfiltering operation performs a spatial average of the prefilter output $m_{ij}(x, y)$. We have shown that this leads to Gaussian probability density functions for the postfilter outputs $m_{p_{ij}}(x, y)$ [10], [11], [15], [34]. The means $\mu_{p_{ij}}$ and variances $s^2_{p_{ij}}$ of the postfilter outputs $m_{p_{ij}}(x, y)$ are derived from the prefilter means and variances using the parameters σ_{g_j} and σ_{p_j}:

$$\begin{aligned} \mu_{p_{ij}} &= \mu_{p_i}(u_j, v_j, \sigma_{g_j}, \sigma_{p_j}) \approx \mu_{g_{ij}} \\ s^2_{p_{ij}} &= s^2_{p_i}(u_j, v_j, \sigma_{g_j}, \sigma_{p_j}) \approx \frac{s^2_{g_{ij}}\ \sigma^2_{g_j}}{\sigma^2_{p_j}} \ . \end{aligned} \qquad (17)$$

where $\sigma^2_{g_j} < \sigma^2_{p_j}$, and we have explicitly shown the dependence on filter parameters. Although we have included the subscripts "ij" in the foregoing discussion, the results are for a single filter channel only. We now proceed to extend the results to the multichannel sceme.

The vector output statistics of k channels is a function of the parameters for all k channels, so we define a matrix $\boldsymbol{\Theta_k}$ that determines the filter parameters:

$$\boldsymbol{\Theta_k} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_k \end{bmatrix} = \begin{bmatrix} u_1 & v_1 & \sigma_{g_1} & \sigma_{p_1} \\ u_2 & v_2 & \sigma_{g_2} & \sigma_{p_2} \\ \vdots & \vdots & \vdots & \vdots \\ u_k & v_k & \sigma_{g_k} & \sigma_{p_k} \end{bmatrix} \qquad (18)$$

where each row in $\boldsymbol{\Theta_k}$ defines the parameters $(u_j, v_j, \sigma_{g_j}, \sigma_{p_j})$ for a single channel.

We propose a multivariate Gaussian model for the vector output statistics of a set of k filter channels, since a Gaussian pdf provides a good model for the output of a single channel. The multivariate Gaussian pdf for texture t_i for a given set of filter parameters $\boldsymbol{\Theta_k}$ is

$$p_i(\boldsymbol{m_p}, \boldsymbol{\Theta_k}) = \frac{1}{(2\pi)^{k/2}|\boldsymbol{C_i}\ |^{1/2}} \quad e^{-(\frac{(\boldsymbol{m_p}-\boldsymbol{\mu_i})^T \boldsymbol{C_i}^{-1}(\boldsymbol{m_p}-\boldsymbol{\mu_i})}{2})} \qquad (19)$$

with

$$\boldsymbol{m_p} = \begin{bmatrix} m_{p_1} \\ m_{p_2} \\ \vdots \\ m_{p_k} \end{bmatrix} \qquad \boldsymbol{\mu_i} = \begin{bmatrix} \mu_{p_{i1}} \\ \mu_{p_{i2}} \\ \vdots \\ \mu_{p_{ik}} \end{bmatrix} \qquad (20)$$

and

$$\boldsymbol{C_i} \quad = \quad \mathrm{E}\left[(\boldsymbol{m_p} - \boldsymbol{\mu_i})(\boldsymbol{m_p} - \boldsymbol{\mu_i})^T\right] \qquad (21)$$

where $\boldsymbol{m_p}$ is a sample of the k-dimensional postfilter-output vector, $m_{p_j} \in m_{p_{ij}}(x, y)$, $\boldsymbol{\mu_i}$ is the mean postfilter-output vector, and $\mu_{p_{ij}}$ is the mean of postfilter output $m_{p_{ij}}(x, y)$. $\boldsymbol{C_i}$ is the $k \times k$ covariance matrix of the postfilter outputs for texture t_i, E[$\cdot$] indicates expected value, superstcipt T indicates transpose, and element $c_{i_{\alpha\beta}}$ in row α and column β of $\boldsymbol{C_i}$ is

$$c_{i_{\alpha\beta}} = E[(m_{p_{i\alpha}} - \mu_{p_{i\alpha}})(m_{p_{i\beta}} - \mu_{p_{i\beta}})] \ . \qquad (22)$$

The argument $\boldsymbol{\Theta_k}$ is explicitly included in (19) to indicate that the output statistics depend in the filter parameters. This dependency arises since the mean vector $\boldsymbol{\mu_i}$ and covariance matrix $\boldsymbol{C_i}$ are both depend on $\boldsymbol{\Theta_k}$ because the means and variances of the filter outputs depend on the filter parameters.

The mean vector $\boldsymbol{\mu_i}$ in (20) can be determined for each texture t_i and each set of candidate filter parameters $(u_j, v_j, \sigma_{g_j}, \sigma_{p_j})$ using (17). The covariance matrix $\boldsymbol{C_i}$ in (21) presents greater difficulty, since it implies a need for samples of the postfilter output for all candidate filters. Even with such samples, the computation of $\boldsymbol{C_i}$ each set of candidate filter parameters is not practical because of the huge number of possible filter combinations. However, the diagonal elements of the covariance matrix are given as $s^2_{p_{ij}}$ from (17). Thus, we propose using the values of $s^2_{p_{ij}}$ along the diagonal of $\boldsymbol{C_i}$ with all off-diagonal elements equal to zero:

$$\boldsymbol{C_i} \quad \approx \quad \begin{bmatrix} s^2_{p_{i1}} & 0 & \cdots & 0 \\ 0 & s^2_{p_{i2}} & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & s^2_{p_{ik}} \end{bmatrix} . \qquad (23)$$

where $s^2_{p_{ij}}$ are from (17). An added advantage of the diagonal form for $\boldsymbol{C_i}$ is that it greatly simplifies the calculation of the determinant and inverse needed later when we estimate segmentation error.

This approximation of $\boldsymbol{C_i}$ implies that the k features corresponding to the postfilter outputs are uncorrelated. To reduce the likelihood of having strongly correlated features, the covariance matrix $\boldsymbol{C_i}$ is approximated using (23) under the additional restriction that constituent Gabor prefilters in any set of candidate filters are not permitted to have significantly overlapping spatial-frequency responses. Thus we also require any pair of Gabor prefilters in $\boldsymbol{\Theta_k}$b to be separated in frequency by more than the sum of the two bandwidths, as defined by the radius of the $e^{-1/2}$ attenuation point in (2). A filter and its mirror image reflected across the u and v frequency axes are considered equivalent for the purposes of eliminating overlapping responses. For example, filters with parameters (U, V, σ_g) and $(-U, -V, \sigma_g)$ are considered identical and are disqualified from being used together in a multichannel system.

B. Segmentation Error Measure

The previous section established a multivariate *Gaussian* statistical model for the vector output of k filter channels. This suggests that the Bhattacharyya distance is an appropriate measure of feature performance [39]. Below, we use the Bhattacharyya distance to form an estimate of segmentation error.

Consider two textures $t_\alpha(x, y)$ and $t_\beta(x, y)$. The Bhattacharyya distance $B(t_\alpha, t_\beta, \boldsymbol{\Theta_k})$, or *B-distance*, between the two textures for a given filter bank determined by $\boldsymbol{\Theta_k}$ is

$$\begin{aligned} B(t_\alpha, t_\beta, \boldsymbol{\Theta_k}) &= \frac{1}{8}(\boldsymbol{\mu_\alpha} - \boldsymbol{\mu_\beta})^T \left[\frac{\boldsymbol{C_\alpha} + \boldsymbol{C_\beta}}{2}\right]^{-1} (\boldsymbol{\mu_\alpha} - \boldsymbol{\mu_\beta}) \\ &+ \frac{1}{2}\, ln \left(\frac{\left| \frac{1}{2}\left(\boldsymbol{C_\alpha} + \boldsymbol{C_\beta}\right)\right|}{|\boldsymbol{C_\alpha}|^{1/2}|\boldsymbol{C_\beta}|^{1/2}} \right) \qquad (24) \end{aligned}$$

where $\boldsymbol{\mu_\alpha}$ and $\boldsymbol{\mu_\beta}$ are the mean vectors, and $\boldsymbol{C_\alpha}$ and $\boldsymbol{C_\beta}$ are the covariance matrices associated with the two textures. The B-distance provides an upper bound for the classification error $\mathcal{E}_c(\boldsymbol{\Theta_k})$ of the two textures. A similar upper error bound for $\mathcal{N}$ multivariate Gaussian classes is [33], [40]

$$\mathcal{E}_c(\boldsymbol{\Theta_k}) < \sum_{\alpha=1}^{\mathcal{N}-1} \sum_{\beta=\alpha+1}^{\mathcal{N}} (\mathcal{P}_\alpha \mathcal{P}_\beta)^{1/2} \rho_{\alpha\beta} \qquad (25)$$

where the two-class Bhattacharyya coefficients $\rho_{\alpha\beta}$ are

$$\rho_{\alpha\beta} = e^{-B(t_\alpha, t_\beta, \boldsymbol{\Theta_k})} ,$$

and the dependence of the error on the filter channel parameters is explicitly indicated by the argument $\boldsymbol{\Theta_k}$ in (25). Equations (24) and (25) provide the relationship between the image-segmentation error and the multivariate Gaussian statistics of the vector output of the k filter channels. In the abscence of additional information, the *a priori* probabilities $\mathcal{P}_\alpha$ are taken to be equal.

In practice, the error measure in (25) is effective for multichannel filter design when the number of textures is very small ($\mathcal{N} < 4$). However, we have observed that segmentation error deteriorates rapidly as the number of textures increases when filters are designed using (25). This deterioration is apparently caused by an overstatement of error in (25) as a larger number of textures crowd the feature space.

The worst-case error in (25) occurs when all of the textures are identically distributed. In this case, the Bhattacharyya coefficients $\rho_{\alpha\beta}$ in (25) are all equal to one, and with equal *a priori* probabilities equation (25) becomes:

$$\mathcal{E}_c(\boldsymbol{\Theta_k}) < \sum_{\alpha=1}^{\mathcal{N}-1} \sum_{\beta=\alpha+1}^{\mathcal{N}} (\mathcal{P}_\alpha \mathcal{P}_\beta)^{1/2} = \frac{1}{\mathcal{N}}\binom{\mathcal{N}}{2} = \frac{\mathcal{N}-1}{2} \quad (26)$$

where

$$\begin{aligned} \mathcal{P}_\alpha &= \frac{1}{\mathcal{N}} & \alpha &= 1, 2, \ldots, \mathcal{N} \\ \boldsymbol{\mu_\alpha} &= \boldsymbol{\mu_\beta} & \forall \quad & \alpha, \beta \\ \boldsymbol{C_\alpha} &= \boldsymbol{C_\beta} & \forall \quad & \alpha, \beta \,. \end{aligned}$$

Thus, it is seen that the upper bound of the classification error $\mathcal{E}_c(\boldsymbol{\Theta_k})$ can greatly exceed one for large numbers of textures. This result supports our observation that filters designed using (25) performed more poorly as the number of textures $\mathcal{N}$ increased. In situations with a larger number of textures $\mathcal{N}$ or with a smaller number of filter channels k, the error in (25) can more easily exceed a value of 1. To prevent the error estimate from exceeding 1, and to improve the filter algorithm performance over a wider range of $\mathcal{N}$, we propose the following modification to equation (25) as an estimate to the classification error:

$$\mathcal{E}_c(\boldsymbol{\Theta_k}) \approx \frac{1}{\mathcal{N}-1} \sum_{\alpha=1}^{\mathcal{N}-1} \sum_{\beta=\alpha+1}^{\mathcal{N}} (\mathcal{P}_\alpha \mathcal{P}_\beta)^{1/2} \rho_{\alpha\beta} \qquad (27)$$

where the worst case upper bound on $\mathcal{E}_c(\boldsymbol{\Theta_k})$ in (27) becomes $\frac{1}{2}$.

C. Total Error Measure

The previous section formulated an estimate of error based on classification error $\mathcal{E}_c(\boldsymbol{\Theta_k})$. However, segmentation of textured images requires not only the accurate classification of textures within regions, but also the accurate localization of boundaries between regions. Thus, as the final component in our development of a measure of image-segmentation error, we include a measure of error at texture boundaries.

Although the error $\mathcal{E}_c(\boldsymbol{\Theta_k})$ provides a measure for the accurate classification of texture within regions, it does not directly address the problem of accurate localization of boundaries between textured regions. Any inaccuracies in boundary locations will necessarily contribute to the overall segmentation error, and we refer to these errors as *localization error*. In our work, we have found it useful to consider two different types of localization error that arise at boundaries between two different textures: *edge error* and *corner error*.

We define edge error as the error in determining the boundary between two textures when the boundary is a straight line, and when there are no corners or other discontinuities in the vicinity of the boundary. We define corner error as the error in determining the boundary between two textures when the boundary points are in the vicinity of a right-angle corner that defines the texture boundary.

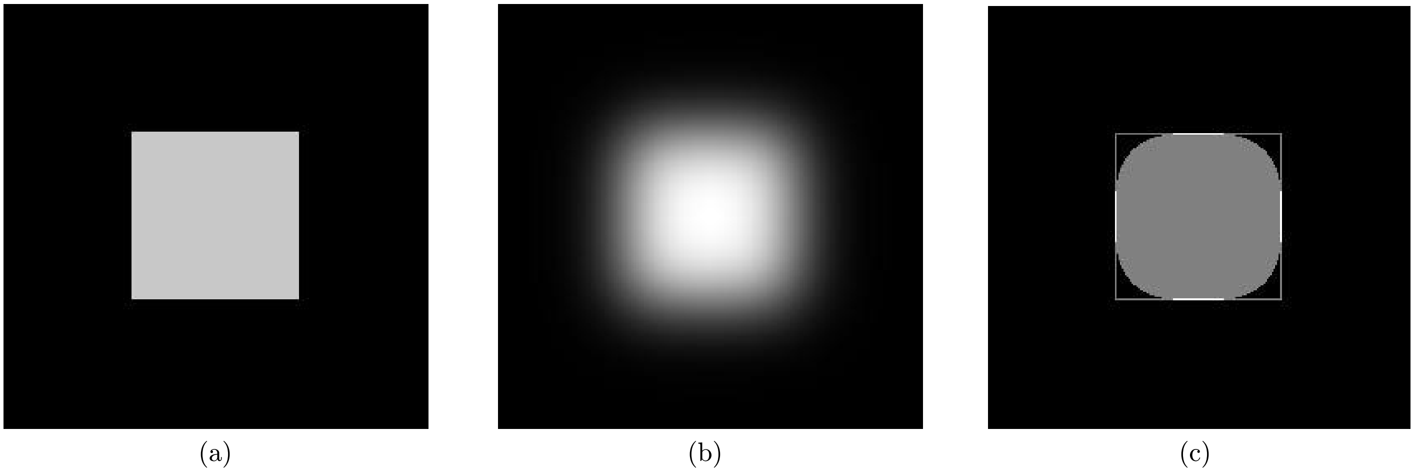


Fig. 3. Segmented square, illustrating two types of error: edge-error and image error. (a) Original 100 × 100 square with gray scale = 200 100on 256 × 256 black background with gray scale = 0. (b) Gaussian filtered image, $\sigma = 20$. (c) Segmentation of filtered image with threshold = 100. There is zero edge error in the image since the boundaries are accurately located (bright white line segments), however significant corner error is apparent (black regions inside square).

In Fig. 3, we illustrate these two types of error. The original image consists of a square region with gray scale of 200 superimposed on a black background with gray scale of 0. The image is filtered by a lowpass Gaussian filter with $\sigma = 20$, resulting in the image in Fig. 3(b). Application of a threshold of 100 to the filtered image results in the segmentation shown in Fig. 3(c). For convenience, an outline of the original square is superimposed on the segmentation. In this example, there is no edge error as can be seen by inspecting the properly determined boundary indicated by the bright line segments at the midpoints of the edges of the square in Fig. 3(c). Conversely, there is significant corner error in this image indicated by the black pixels located within the interior of the corners of the square region.

The example shown in Fig. 3 also illustrates the tradeoff between corner error and edge error when a simple threshold is used for segmentation. Obviously, any improvement in corner error that would be achieved by changing the segmentation threshold would come at the expense of increased edge error. For this example, it is apparent that no threshold results in zero corner error. Although the example is admittedly simple, it illustrates the complex interaction between the filter parameters, classifier scheme, image geometry, within-region classification, and boundary localization.

Our approach to edge error and corner error is to address these two types of error separately. First, the edge error is addressed by using a mixture distribution to obtain a slightly modified version of a Bayesian classifier based on the multivariate density in (19). Second, we address corner error by adding it to $\mathcal{E}_c(\boldsymbol{\Theta_k})$ to produce a measure that accounts for both classification error within textured regions and localization error at region boundaries. Thus, we reduce corner error by incorporating it into the error measure used in the filter design procedure, and we reduce edge error by slightly modifying the Bayesian classifier in a manner that does not significantly degrade within-region classification. The use of a mixture distribution to reduce edge error is described as part of the classifier design; the method for reducing corner error is discussed below, since it is included as part of the filter-design algorithm.

The corner error tends to increase as the spatial resolution of the filters becomes coarse, since small features such as the sharp corner are blurred by the filter. This implies that we should add a localization error term to (27) that increases predicted segmentation error as the filtering becomes coarser or, equivalently, increases error as σ_{g_j} and σ_{p_j} become larger.

Previous investigators have noted that accurate classification within textured regions is achieved with larger spatial operators, but that accurate localization of boundaries is achieved with smaller spatial operators [4], [41]. For this reason, $\mathcal{E}_c(\boldsymbol{\Theta_k})$ in (27) is not sufficient for filter design. In our experience, designing filters using $\mathcal{E}_c(\boldsymbol{\Theta_k})$ alone as the error measure results in filter designs with the largest possible σ_{g_j} and σ_{p_j}. To providing some counterbalance to the classification error in (27), we add a measure of localization error that favors smaller values of σ_{g_j} and σ_{p_j}.

Following the example in Fig 3, we propose the following simple empirical measure of the localization error $\mathcal{E}_l(\boldsymbol{\Theta_k})$ in the 4 corners of the square:

$$\mathcal{E}_l(\boldsymbol{\Theta_k}) \approx \frac{1}{k}\sum_{\gamma=1}^{k} \frac{2\mathcal{N}(\sigma_{p_\gamma}^2 + \sigma_{g_\gamma}^2)}{N^2} \,, \tag{28}$$

where the image dimensions are $N \times N$, $\mathcal{N}$ is the number of textures, and the term $(\sigma_{p_\gamma}^2 + \sigma_{g_\gamma}^2)$ approximates the effective spatial localization of the combined prefilter and postfilter in a single channel. The summation over k generates an average corner error over the k channels. We include the argument $\boldsymbol{\Theta_k}$ in (28), since σ_{p_γ} and σ_{g_γ} depend on $\boldsymbol{\Theta_k}$. For the particular case in Fig. 3, the measured error is 0.015 and the predicted error from (28) is 0.023. The error is somewhat overstated in (28) compensate for the simplicity of the expression and other unattributed sources of localization error. We could improve the accuracy of (28) for the case in Fig. 3, but it is unclear that such refinement is appropriate in light of the greatly simplifying construction that underlies the formulation. It appears that the most important characteristics of the expression is that $\mathcal{E}_l(\boldsymbol{\Theta_k})$ be representative of the magnitude of the error, and that $\mathcal{E}_l(\boldsymbol{\Theta_k})$ decreases as some weighted average of σ_{g_j} and σ_{p_j} decreases. In either event, our results show the formulation to generate effective filters.

The total error measure $\mathcal{E}_t(\boldsymbol{\Theta_k})$ for filter selection is then the sum of the classification error $\mathcal{E}_c(\boldsymbol{\Theta_k})$ and the localization error $\mathcal{E}_l(\boldsymbol{\Theta_k})$:

$$\begin{aligned} \mathcal{E}_t(\boldsymbol{\Theta_k}) &= \mathcal{E}_c(\boldsymbol{\Theta_k}) + \mathcal{E}_l(\boldsymbol{\Theta_k}) \\ &\approx \frac{1}{\mathcal{N}-1}\sum_{\alpha=1}^{\mathcal{N}-1}\sum_{\beta=\alpha+1}^{\mathcal{N}} (\mathcal{P}_\alpha \mathcal{P}_\beta)^{1/2}\, e^{-B(t_\alpha, t_\beta, \boldsymbol{\Theta_k})} \\ &+ \frac{1}{k}\sum_{\gamma=1}^{k} \frac{2(\mathcal{N})(\sigma_{g_\gamma}^2 + \sigma_{p_\gamma}^2)}{N^2} \end{aligned} \tag{29}$$

where σ_{g_γ} and σ_{p_γ} are defined by $\boldsymbol{\Theta_k}$. The total error $\mathcal{E}_t(\boldsymbol{\Theta_k})$ is then used as the basis for designing the Gabor prefilters and Gaussian postfilters in the multichannel design. Although the localization error measure $\mathcal{E}_l(\boldsymbol{\Theta_k})$ is admittedly simple, we have found it to produce effective filter designs. This may be in part due to the fact that we tend to use square textured regions in our experiments. Other experimentation that led to $\mathcal{E}_l(\boldsymbol{\Theta_k})$ also suggests that the form of the localization error measure is not critical, but that it should reasonably reflect the actual magnitude of the localization error. Similarly, experimentation led us to the modification of the classification error $\mathcal{E}_c(\boldsymbol{\Theta_k})$ in (27) discussed previously.

This completes the development of the multichannel paradigm that establishes the relationships between the predicted image-segmentation error $\mathcal{E}_t(\boldsymbol{\Theta_k})$ in (29), the filter parameters $\boldsymbol{\Theta_k}$ in (18), the multivariate output statistics $p_i(\boldsymbol{m_p},\boldsymbol{\Theta_k})$ in (19), and the power spectra of the sample textures $S_i(u,v)$ in (10).

IV. Filter Design Algorithm

Using the results of Section III, we now present the procedure for designing k Gabor prefilters and k Gaussian postfilters to segment$\mathcal{N}$ given textures. The design of the set of Gabor prefilters and Gaussian postfilters is based on the total error measure in (29). Our approach is to construct an extensive set of candidate filter channels, then choose the best multichannel design $\boldsymbol{\Theta_k}$ using the total error from (29). Before describing the filter-design algorithm, we briefly discuss the construction of the set of candidate filter channels and the method for selecting the best filter set.

A. Candidate Filters

The multichannel paradigm of Section III provides the relationship between the predicted segmentation error and the filter parameters for a given set of texture samples. Using (18) we then define the optimum set of filter channels $\boldsymbol{\Theta_{opt}}$ as the set of filter channels that minimize the error $\mathcal{E}_t(\boldsymbol{\Theta_k})$ in (29):

$$\boldsymbol{\Theta_{opt}} = \min_{\boldsymbol{\Theta_k}} \; \{\mathcal{E}_t(\boldsymbol{\Theta_k})\} \quad . \tag{30}$$

It may be possible to find a closed-form solution for some simple texture model and other simplifying assumptions. However, we consider the more general case requiring a search of the possible filter combinations $\boldsymbol{\Theta_k}$ for $\boldsymbol{\Theta_{opt}}$.

We first construct a collection Ψ of *individual* candidate filter channels, from which collection the set of k channels will be constructed. As we showed earlier, it is not feasible to consider all possible filter-channel combinations. Thus, we choose the following reduced set of candidate filter channels Ψ that provides overlapping coverage of the frequency plane for each possible σ_{g_j}:

$$\Psi = \left\{ (u_{\gamma\eta_1}, v_{\gamma\eta_2}, \sigma_{g_\gamma}, \sigma_{p_{\gamma\nu}}) \right\} \tag{31}$$

such that:

$$\begin{aligned} \sigma_{g_\gamma} &\in \Sigma \\ \sigma_{p_{\gamma\nu}} &\in \{\lambda\, \sigma_{g_\gamma} \mid \lambda \in \Lambda\} \\ & \quad \text{where:} \\ & \quad \lambda \in \Lambda\ , \\ (u_{\gamma\eta_1}, v_{\gamma\eta_2}) &\in \left\{ \left(\frac{\eta_1}{\sqrt{8\pi^2\sigma_{g_\gamma}^2}}, \frac{\eta_2}{\sqrt{8\pi^2\sigma_{g_\gamma}^2}}\right) \right\}\ ; \\ & \quad \text{where:} \\ & \quad -0.5 \le u_{\gamma\eta_1} < 0.5\ , \\ & \quad 0 \le v_{\gamma\eta_2} < 0.5\ , \\ & \quad \eta_1, \eta_2 \in \{\ldots, -1, 0, 1, 2, \ldots\}\ . \end{aligned}$$

with Σ being a set of candidate prefilter σ_{g_γ}'s, and Λ being a set constants determining candidate values of $\sigma_{p_{\gamma\nu}}$ relative to each value of σ_{g_γ}. The subscripts γ, ν, η_1, and $\eta 2$ are used to indicate the interdependencies of the parameters, since the possible values of $\sigma_{p_{\gamma\nu}}$ and $(u_{\gamma\eta_1}, v_{\gamma\eta_2})$ depend on the value of σ_{g_γ}.

Candidate center frequencies $(u_{\gamma\eta_1}, v_{\gamma\eta_2})$ are restricted to the closed right-half frequency plane at integer multiples of $1/(8\pi^2\sigma_{g_\gamma}^2)$. The term $1/(8\pi^2\sigma_{g_\gamma}^2)$ represents the displacement from center frequency at which the Gabor prefilter frequency response (2) equals $e^{-0.25} = 0.78$. The separation in candidate frequencies is chosen to allow significant overlap between adjacent filters, but not so small as to unduly increase the number of candidate filters and computation time. Although overlapping filters are members of Ψ, any valid combination of parameters $\boldsymbol{\Theta_k}$ is restricted such that the Gabor prefilter passbands do not overlap at the $e^{-0.25}$ frequency-response point. In addition, center frequencies within $1/(8\pi^2\sigma^2_{g_\gamma})$ of the origin $(u, v) = (0, 0)$ are not considered, to ensure a proper bandpass image for the Rician model and to reduce possible leakage effects due to the finite attenuation characteristics of the filters admitting low-frequency spurious energy [2], [3].

For each particular combination of σ_{g_γ} and $\sigma_{p_{\gamma\nu}}$ the collection of filters in (31) simply represents an overlapping tessalation of candidate Gabor prefilters in the frequency half-plane. The candidate values of σ_{g_γ} are typically varied in octave steps, and the candidate values of $\sigma_{p_{\gamma\nu}}$ are typically varied in multiples of $1.5 \times \sigma_{g_\gamma}$. In our experience, $\sigma_{g_\gamma} \in \{2, 4, 8\}$ and $\lambda \in \{1.5, 2\}$ are typical. Larger values of σ_{g_γ} and λ appear to generate wider spatial filter responses that are penalized by (28). Finally, we note that the algorithm is not limited to octave scalings of σ_{g_γ} and can accommodate more general sets such as $\Sigma = \{2.2, 2.9, 4.1\}$.

While we have suggested selecting candidate σ_{g_γ}'s and $\sigma_{p_{\gamma\nu}}$'s that are scaled in some fashion, the algorithm is not so limited and can accommodate more general sets such as $\Sigma = \{2.2, 2.9, 4.1\}$.

B. Filter Selection

In constructing Ψ we reduced the number of candidate filter channels while maintaining modestly overlapping coverage of the frequency plane. Even so, the number of possible combinations of k channels remains prohibitive. To see this, let n_g be the number of elements in Σ and n_p be the number of elements in Λ. If the number of center frequencies is approximately $N^2/(\sigma_{g_\gamma})^2 \approx (N/8)^2$, and the number of prefilter-postfilter combinations is $n_g \times n_p$, then there are approximately $(n_g\ n_p)\,(N/8)^2$ individual filter channels and approximately $[(n_g\ n_p)\,(N/8)^2]^k$ possible combinations of k filters. With $k = 5$, $N = 256$, $n_g = 3$, and $n_p = 2$, there are ≈ 6000 individual filters and $\approx 8 \times 10^{18}$ possible filter-channel combinations. Were it possible to evaluate 10^8 filter-channel combinations per second, this example would require 2500 years. We, therefore, cannot directly evaluate all filter channel combinations arising from Ψ.

The number of possible channel combinations leads us to consider alternatives to a direct search for the best combination of k filter channels. In the following, we use a forward-sequential filter-selection method to find the best k-channel design from the possible candidate filter-channel combinations [33]. Although this is a suboptimal approach, our experimental results yield effective filter designs.

In the forward-sequential method, the k-channel design is achieved incrementally by adding one channel at a time. The first filter channel selected is the best individual filter θ_1:

$$\theta_1 = \theta_\lambda \text{ such that } \mathcal{E}_t([\theta_\lambda]) \le \mathcal{E}_t([\theta_\xi])\ \forall\ \theta_{t_\xi} \in \Psi\ . \qquad (32)$$

To proceed further, we first define the filter channel set at iteration δ of the forward sequential algorithm as

$$\boldsymbol{\Theta}_\delta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_{\delta-1} \\ \theta_\delta \end{bmatrix} = \begin{bmatrix} u_1 & v_1 & \sigma_{g_{\alpha 1}} & \sigma_{p_1} \\ u_2 & v_2 & \sigma_{g_{\alpha 2}} & \sigma_{p_2} \\ \vdots & \vdots & \vdots & \vdots \\ u_{\delta-1} & v_{\delta-1} & \sigma_{g_{\alpha(\delta-1)}} & \sigma_{p_{(\delta-1)}} \\ u_\delta & v_\delta & \sigma_{g_{\alpha\delta}} & \sigma_{p_\delta} \end{bmatrix} \qquad (33)$$

where $\delta \le k$, and k is the number of desired filter channels in the texture-segmentation system of Fig. 1.

Using (33), subsequent steps in the forward-sequential filter design algorithm can then be written in a recursive form. Filter parameters $\boldsymbol{\Theta}_\delta$ at the δth stage of the forward sequential scheme are then defined in terms of the filter parameters $\boldsymbol{\Theta}_{\delta-1}$ at stage $\delta - 1$:

$$\boldsymbol{\Theta}_\delta = \left[\frac{\boldsymbol{\Theta}_{\delta-1}}{\theta_\delta}\right] \qquad (34)$$

such that

$$\mathcal{E}_t\left(\left[\frac{\boldsymbol{\Theta}_{\delta-1}}{\theta_\delta}\right]\right) \le \mathcal{E}_t\left(\left[\frac{\boldsymbol{\Theta}_{\delta-1}}{\theta_\xi}\right]\right)\ \forall\ \theta_\xi \in \Psi$$

where $\boldsymbol{\Theta}_\delta$ is a function of θ_δ, and where $\boldsymbol{\Theta}_{\delta-1}$ is a fixed column vector established at step $\delta - 1$ of the forward sequential procedure. The forward sequential algorithm terminates when k filter channels are designed, i.e., when $\delta = k$. The final filter design is then $\boldsymbol{\Theta_k}$. Although a single filter channel θ_δ is added at each step in (34), the criteria for selecting the added channel is based on effect of all δ channels combined through the vector error measure $\mathcal{E}_t(\boldsymbol{\Theta_\delta})$. Thus, as each new channel is added the vector output statistics of all δ channels are considered.

C. Design Algorithm

Combining the foregoing results, the procedure for designing k filter channels comprised of k Gabor prefilters and k Gaussian postfilters is:

Step 1. Construct a large collection Ψ of individual candidate filter channels using (31). A typical set of parameters would be $\Sigma = \{2, 4, 8\}$ and $\Lambda = \{1.5, 2\}$. For this Σ and Λ, the resulting possible combinations of prefilter and postfilter sigmas would be $(\sigma_{g_\gamma}, \sigma_{p_{\gamma\nu}}) \in \{(2, 3), (2, 4), (4, 6), (4, 8), (8, 12), (8, 16)\}$.

Step 2. Compute $A_i^2(u,v,\sigma_{g_j})$ and $N_i(u,v,\sigma_{g_j})$ for each $\sigma_{g_j} \in \Sigma$ and each texture t_i using (12) and (13). This is done for each sample texture $t_i(x,y)$ under consideration. $P_i(u,v,\sigma_{g_j})$ in (10) can be computed efficiently using a Fast Fourier Transform [10], [15]. The result of this step is the parameters $A_i^2(u,v,\sigma_{g_j})$ and $N_i(u,v,\sigma_{g_j})$ at each discrete frequency (u,v) for each $\sigma_{g_j} \in \Sigma$ and for each texture t_i, $i = 1,2,\ldots,\mathcal{N}$.

Step 3. Using the results from step 2, find $\mu_{g_i}^2(u,v,\sigma_{g_j})$ and $s_{g_i}^2(u,v,\sigma_{g_j})$ using a numerical approximation of (15) at each discrete frequency (u,v) for each $\sigma_{g_j} \in \Sigma$ and for each texture t_i, $i = 1,2,\ldots,\mathcal{N}$.

Step 4. Find the best single filter-channel $\theta_1 \in \Psi$ using (32). The predicted segmentation error assiciated each filter is computed using (29) and the results from step 3 for each set of filter parameters $\theta \in \Psi$. (Note that (29) takes a scalar form for a single filter channel, i.e., the multivariate Gaussian in (19) becomes univariate.)

Step 5. Search for subsequent filters using the forward sequential algorithm in (34), terminating when the desired number of filter channels k is reached. Candidate filter-channel combinations are restricted so that the passbands of Gabor prefilters do not overlap at the $e^{-0.25}$ frequency-response point. The final filter design is then $\boldsymbol{\Theta_k}$. (Alternatively, terminate the algorithm when the predicted segmentation error $\mathcal{E}_t(\boldsymbol{\Theta_\delta})$ reaches some desired level.)

V. Classifier and Postprocessing

In the previous section we presented the procedure for finding the k-channel design $\boldsymbol{\Theta_k}$. In this section, we summarize the design of the remaining classifier and post-processing portions of the texture segmentation system of Fig. 1. Although the focus of the present paper is the design of the filters $\boldsymbol{\Theta_k}$, the classifier and postprocessing portions of the system are needed to generate the texture segmentations that illustrate the performance of the k-channel design $\boldsymbol{\Theta_k}$.

We use a straightforward Bayesian vector classifier, so that we may observe the effectiveness of the filter design rather than the effectiveness of a sophisticated classifier. More elaborate methods can be used to generate the segmented image from the k-dimensional vector output of the k channels, but a simple classifier scheme more directly illustrates the effectiveness of the filter design. Further, our proposed Bayesian classifier is based on the statistical models (19) of the multichannel paradigm, and so the segmentation results confirm the effectiveness of both the filter design and the underlying multichannel paradigm.

After performing the classification, a final postprocessing stage is employed as shown in Fig. 1. The postprocessing resembles an "n-ary" morphological operation and is used to remove residual error at boundaries between two different textures. These misclassifications appear to occur when the feature vector passes through an intermediate region in feature space that is classified as a third texture. The mechanisms causing such misclassifications are outlined elsewhere for a simple one-dimensional case [15]. In our experiments, the problem seems to occur more frequently when the number of filter channels k is less than or equal to the number of textures being segmented.

In the following two sections, we briefly overview the classifier and postprocessing. Further detail on the classifier and postprocessing can be found in [15] and the sequel [PAPER2].

A. Classifier

Given the multivariate Gaussian pdf $p_i(\boldsymbol{m_p},\boldsymbol{\Theta_k})$ in (19), a Bayesian classifier could be implemented by simply selecting the texture t_i with the largest corresponding $p_i(\boldsymbol{m_p},\boldsymbol{\Theta_k})$ for a given $\boldsymbol{m_p}$. Although we have found such a classifier to perform well within textured regions, we have observed *localization error* at texture boundaries; i.e., the boundary is displaced from its true location [15]. We have also observed that we can reduce this localization error by slightly modifying the Bayesian classifier using a mixture density. The use of a mixture density provides a fairly simple means for shifting the multi-dimensional decision surfaces of the classifier in a manner that tends to prevent displacement of segmentation-boundary locations.

To illustrate the effect of the mixture density on classifier performance, consider the one-dimensional case of two Gaussian pdf's depicted in Fig 4(a), corresponding to the postfilter output statistics of a single filter channel. The optimum classification threshold of 1.6 coincides with the intersection of the two pdf's. (A second threshold exists, but does not contribute materially to the discussion.)

Now consider the optimum threshold for minimizing the localization error at a boundary. For simplicity, consider the one-dimensional step from amplitude 1 to amplitude 7 shown in Fig. 4(b). The two values on either side of the step correspond to the mean values of the pdf's in Fig. 4(a). The result of filtering this step with a Gaussian lowpass filter with $\sigma = 5$ is also shown. From the intersection of the two solid curves in (b), the optimal threshold for the filtered step from a localization standpoint, is the average of the two amplitudes $\bar{\mu} = (\mu_1 + \mu_2)/2 = 4$. The optimal classification theshold from a classification standpoint is shown as a dashed line in Fig. 4(b), and would result in an error of 5 units in locating the step boundary. Thus, the the optimal classification threshold of 1.6 differs from the optimum localization threshold of 4 from.

The situation in Fig. 4(a) represents a well-separated pair of classes with a low associated classification error. We notice that movement of the classification threshold from 1.6 to 4 in Fig. 4 will not seriously degrade classification error. In fact, for this particular example, the error would change from $\approx 10^{-9}$ to $\approx 10^{-3}$. On the other hand, a 5 pixel boundary error as indicated in Fig. 4(b) corresponds to a localization error of $\approx (5 \cdot 4 \cdot 100)/256^2 = .03$ for a 100×100 pixel square region in a 256×256 image. The larger magnitude of the localization error suggests that it is better to adjust the classification threshold to reduce error at the boundary. Thus, the approach taken to reduce the overall segmentation error is to modify the Bayesian decision surfaces to reduce the localization error.

Thus, we propose to use a classifier based on a mixture-

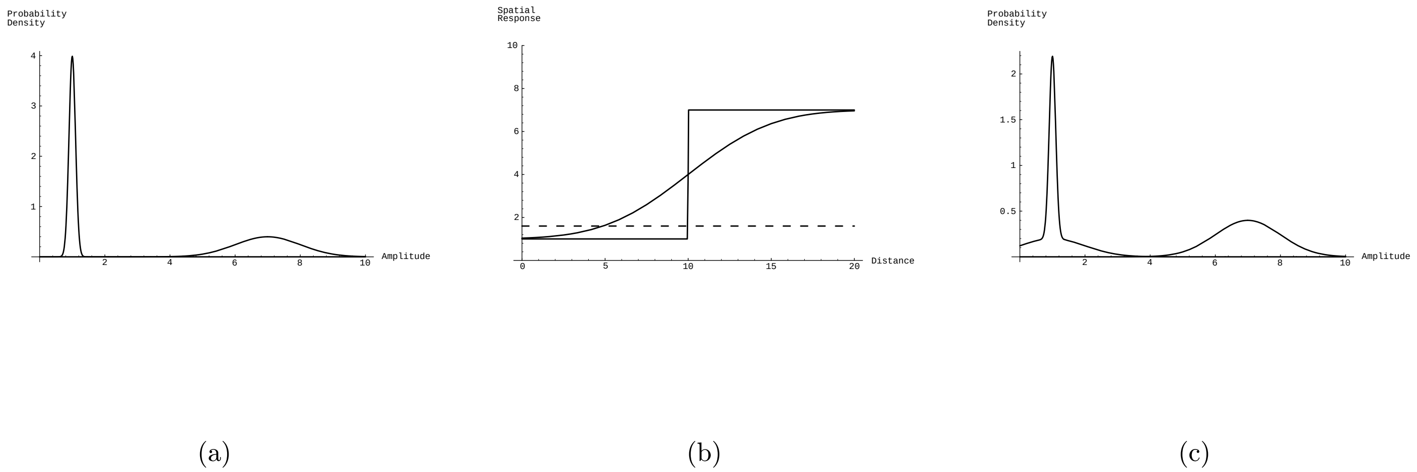


Fig. 4. Two Gaussian pdf's corresponding to the output of a single filter channel for two different textures. (a) Plot of probability density versus filter output amplitude; lower Gaussian pdf mean $\mu_1 = 1$, standard deviation $\sigma_1 = 0.1$, upper pdf mean $\mu_2 = 7$ standard deviation $\sigma_2 = 1$, and optimum classification threshold = 1.6. (b) One-dimensional step edge and Gaussian-filtered step edge. The step edge is shown centered at spatial coordinate 10 with an amplitude step from $\mu_1 = 1$ to $\mu_2 = 7$. The Gaussian lowpass filtered version of the step is also shown for filter $\sigma = 5$. The optimal classification threshold of 1.6 from (a) is shown as a dashed line and results in a boundary displacement of ≈ 5 from the actual step edge location. (c) Two mixture pdf's formed by adding each of the pdf's in (a) to a second Gaussian pdf with the same mean, but with a standard deviation $\sigma_{max} = 1$ that equals the largest standard deviation of the two original pdf's. The optimal classification threshold is now approximately 4 and greatly reduce the boundary displacement seen in (b).

density consisting of the average of two multivariate-Gaussian densities. The mixture-density is readily implemented for the present case of multivariate-Gaussian classes. To formulate the mixture density, first select the largest variances $s^2_{p_{max_j}}$ on each feature axis j:

$$s^2_{p_{max_j}} \geq s^2_{p_{\alpha j}}, \qquad \alpha \in \{1, 2, \ldots \mathcal{N}\}\ , \tag{35}$$

where each dimension, or feature axis, corresponds to the output of a filter channel. Next, a covariance matrix $\boldsymbol{C_{max}}$ is formed where the diagonal elements are the maximum variances $s^2_{p_{max_j}}, j = 1, 2, \ldots k$, and where all off-diagonal elements are zero:

$$\boldsymbol{C_{max}} \quad = \quad \begin{bmatrix} s^2_{p_{max_1}} & 0 & \cdots & 0 \\ 0 & s^2_{p_{max_2}} & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & \ddots & \cdots & s^2_{p_{max_k}} \end{bmatrix} . \tag{36}$$

The mixture density p_{mix_i} for texture t_i is then defined as

$$p_{mix_i}(\boldsymbol{m_p}) = \frac{1}{2}\Big(p_i(\boldsymbol{m_p},\boldsymbol{C_{max}}) + p_i(\boldsymbol{m_p},\boldsymbol{C_i})\Big), \tag{37}$$

where $p_i(\boldsymbol{m_p},\boldsymbol{C_i})$ is the original multivariate Gaussian pdf from (19) and $p_i(\boldsymbol{m_p},\boldsymbol{C_{max}})$ is the multivariate Gaussian formed by taking the maximum diagonal elements of all the covariance matrices for the textures.

The proposed classifier for texture segmentation is then a Bayesian classifier based on the mixture density (37) where the output is assigned to the texture with the largest probability; i.e. $c(x, y) = \alpha$ such that

$$p_{mix_\alpha}(\boldsymbol{m_p}(x, y)) \geq p_{mix_\beta}(\boldsymbol{m_p}(x, y)), \qquad \forall\, \beta \tag{38}$$

where $c(x, y)$ is the classified image pixel at coordinate (x, y), $\beta \in \{1, 2, \ldots, \mathcal{N}\}$, and p_{mix_β} is the mixture density for texture t_β given in (37).

B. Postprocessing

Even though the mixture-density classifier of the previous section reduces segmentation error in the vicinity of texture boundaries, narrow regions in the classified image $c(x, y)$ are sometimes observed to be misclassified as a third texture near the boundary between two textures. These narrow misclassified regions seem to occur when the feature vector makes a transition through feature space from a vector charateristic of one texture, to a different vector characteristic of the second texture at the boundary. The mechanisms leading to these misclassifications are most readily illustrated in one dimension and are described in further detail elsewhere [15]. Furthermore, these misclassifications at boundaries are seen in results of other segmentation methods [6], [42].

In practice, we find that these narrow misclassified regions at texture boundaries can be removed using a two-step postprocessing procedure that resembles a morphological operation [43]. Although not strictly a morphological operation, the postprocessing bears strong resemblance to a morphological erosion. We are not aware of a similar multi-class, or "n-ary", morphology, but portions of our procedure are similar to portions of boundary localization work by Yann and Young [41].

To illustrate the misclassifications at region boundaries, consider the one-dimensional situation in Fig. 5(a). In this figure, there are 3 Gaussian pdf's corresponding to the output pdf's of a single filter channel for 3 textures. Three corresponding mixture distributions as in (37) for $k = 1$ are shown in Fig. 5(b) along with Bayesian thresholds. Amplitudes from 0 to 1.7 are classified as texture 1, from 1.7 to 4.4 are classified as texture 2, and above 4.4 is classified as texture 3.

Next, consider a one-dimensional version of a boundary between texture 1 and texture 3, corresponding to an amplitude step from $\mu_1 = 1$ to $\mu_3 = 7$ as shown in Fig. **??**. Also shown is the step response through a lowpass Gaussian filter with $\sigma = 5$, corresponding to the response of a single filter channel. The two classification thresholds from Fig. 5 are shown as dashed lines in Fig. **??**. The classifier then makes the correct assignment of texture 1 for points more than 5 units to the left of the filtered step, and texture 3 for points more than 1 unit to the right of the edge. However, points in the region from -5 to +1 units relative to the edge are incorrectly classified as texture 2.

The problem of incorrect texture assignments at region boundaries becomes considerably more complicated in the multi-dimensional case of the k-channel system.

To remove these misclassifications at region boundaries, we propose an approach similar to a morphological approach [43]. Although not strictly a morphological operation, the following procedure bears strong resemblance to a morphological erosion. In the first step of our two-step postprocessing procedure, pixels in $c(x, y)$ whose neighborhood consists entirely of one texture class are left unchanged; otherwise, the pixel value is set to zero to indicate it is no longer assigned to any class. We observe that the magnitude of the boundary displacement is proportional to the spatial extent of the filter responses. Thus we select a neighborhood size $\bar{\rho}$ that is proportional to an empirical average width of the filter-channel responses:

$$\bar{\rho} = \frac{1}{k} \sum_{j=1}^{k} \sqrt{\sigma_{gj}^2 + \sigma_{pj}^2} \tag{39}$$

where σ_{gj} and σ_{pj} are the prefilter and postfilter parameters for the kth channel. The result of performing this first step resembles a morphological erosion operation.

In the second step, classified regions are propagated back into the unassigned regions. Each unassigned pixel is assigned to the most prevalent class within the 8-neighborhood surrounding that pixel. This propogation affects only unassigned pixels and ceases when all unassigned pixels have been assigned to one of the $\mathcal{N}$ texture classes. This second step resembles a morphological dilation operation, and the resultant image is the final segmented image $i_s(x, y)$ in Fig. 1.

Alternative classifier and postprocessing approaches are the topic of ongoing research. However, the present methods serve to illustrate the effectiveness of the designed filters in a complete system while addressing issues that can arise when a small number of filter channels is used relative to the number of textures [**?**], [15].

VI. Results

The multichannel filter design algorithm, along with the proposed postprocessing, have been tested using a wide range of Brodatz and synthetic texture images [15], [44]. Two samples of our results are shown in Figs. 6 and 7. All images are 256 × 256 pixel 8-bit gray-scale images. The mean values of the textures were equalized so that segmentation based on average gray scale was not possible.

The image shown in Fig. 6(a) is composed of five samples from the Brodatz texture album and resembles the "Nat-5" image used by previous investigators to test texture segmentation methods [4], [44], [45]. The parameters $\Sigma = \{3, 6, 12\}$ and $\Lambda = \{1.5\}$ were used to construct the set of candidate filters Ψ. Fig. 6(b) is the result of a $k = 2$ channel segmentation, (c) is the result of a $k = 4$ channel segmentation, and (d) is the result of a $k = 6$ channel segmentation. The measured segmentation error decreases from 0.13 to 0.05 to 0.04 as the number of filter channels increases from $k = 2$ to $k = 6$. Because the 4-channel segmentation in (c) already has fairly low segmentation error, it is difficult to observe additional improvement the 6-channel segmentation in (d).

The parameters for the six channels used in Fig. 6(d) are

$$\mathbf{\Theta_6} = \begin{bmatrix} -0.078 & 0.109 & 6 & 12 \\ 0.125 & 0.015 & 6 & 12 \\ 0 & 0.187 & 3 & 6 \\ 0.203 & 0 & 3 & 6 \\ -0.265 & 0.234 & 3 & 6 \\ 0.312 & 0.187 & 3 & 4.5 \end{bmatrix}$$

where the first row corresponds to the first filter selected in the forward-sequential algorithm, and so on. Thus, the 2-channel design of Fig. 6(b) uses the 2 channels defined by the top 2 rows of $\mathbf{\Theta_6}$, and the 4-channel design of Fig. 6(c) uses the 4 channels defined by the top 4 rows of $\mathbf{\Theta_6}$.

The first few filters in the design have larger values of σ_{g_j} and σ_{p_j} that tend to provide good classification within regions but poor spatial resolution. As the forward-sequential algorithm proceeds, finer resolution filters are

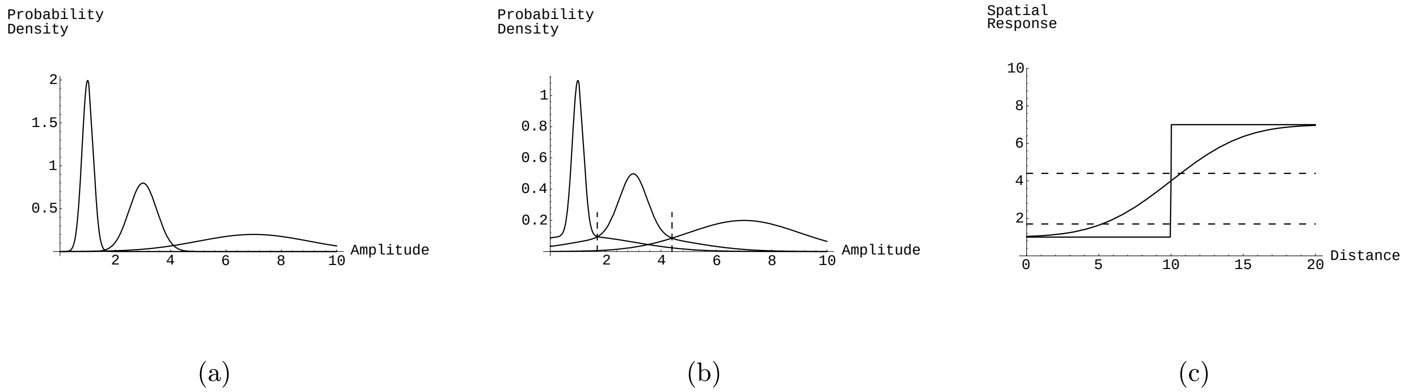


Fig. 5. Three Gaussian pdf's corresponding to the output of a single filter channel for three different textures. (a) Plot of probability density versus filter output amplitude; leftmost pdf mean $\mu_1 = 1$, standard deviation $\sigma_1 = 0.2$, center pdf mean $\mu_2 = 3$, standard deviation $\sigma_2 = 0.5$, upper pdf mean $\mu_3 = 7$ standard deviation $\sigma_3 = 2$. (b) Mixture pdf's showing classification thresholds at intersection of the pdf's with an amplitude of ≈ 1.7 and 4.4. (c) One dimensional step edge and Gaussian filtered step edge. The step edge is shown centered at spatial coordinate 10 with an amplitude step from $\mu_1 = 1$ to $\mu_3 = 7$ corresponding to textures 1 and 3 in Fig. 5. The Gaussian lowpass filtered version of the step is also shown for filter $\sigma = 5$. The optimal classification thresholds from Fig. 5 are shown as a dashed lines and results in an erroneous classification of the as texture 2 over a distance extending from ≈ 5 to the left of the edge to ≈ 1 to the right of the edges.

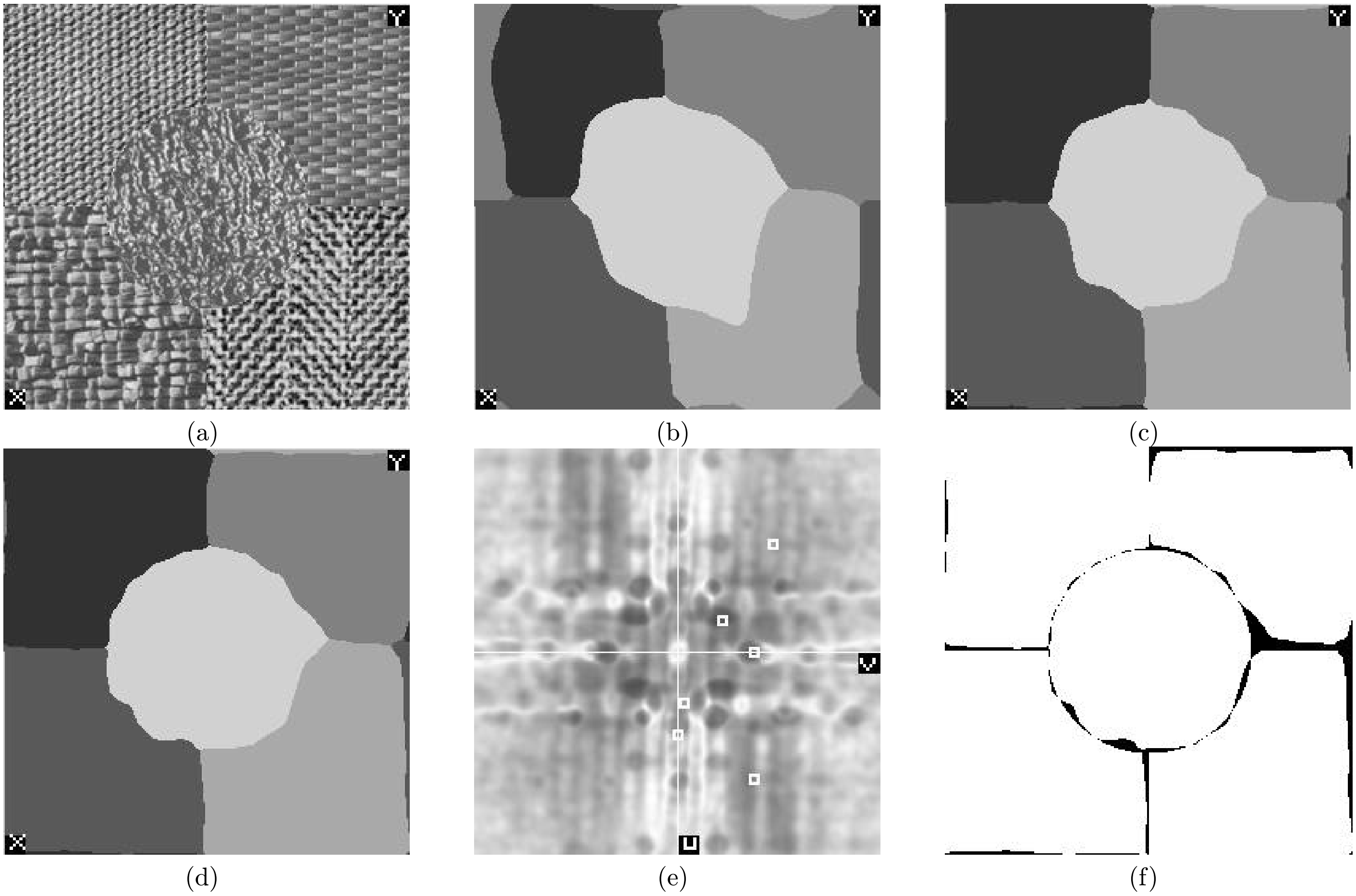


Fig. 6. Results for $\Sigma = \{3, 6, 12\}$, $\Lambda = \{1.5\}$. (a) Input composite image "Nat-5," clockwise from top left: d77 "cotton canvas," d55 "straw matting," d17 "herringbone weave," d84 "raffia," d24 "pressed calf leather" in center. (b) 2 channel segmentation, error=0.13. (c) 4 channel segmentation, error=0.05. (d) 6 channel segmentation, error=0.04. (e) Filter frequencies (white squares) plotted on error map. (f) Segmentation error for six-channel segmentation, misclassified pixels in black.

added that tend to improve performance near boundaries. The tendency to initially start with coarser filters reflects the predominance of classification error due to the predominance of within-region pixels relative to boundary pixels. Later, the progression to finer filters reflects the dominance of localization error near boundaries after the first few filters achieve effective within-region classification. For images comprised of many small textured regions with an accompanying large amount of texture boundary, some modification of the formulation of $\mathcal{E}_l(\boldsymbol{\Theta_k})$ may be appropriate.

Predicted segmentation error $\mathcal{E}_t$ is plotted as a function of prefilter center frequency (u, v) in Fig. 6(e) for a single filter channel at the largest possible prefilter sigma, $\sigma_g = 12$. The segmentation error is plotted as a gray scale, with black indicating no error and white 100% error. The center of Fig. 6(e) corresponds to $(u, v) = (0, 0)$ cycles-per-pixel. The u axis ranges from -.5 at the top to +.5 at the bottom, and the v axis ranges from -.5 at the left to +.5 at the right of the image. The largest value of σ_g is chosen for the plot since it corresponds to the finest resolution in the frequency domain.

The Gabor prefilter center frequencies from $\boldsymbol{\Theta_6}$ are indicated with white squares in Fig. 6(e). Although the plot does not indicate the segmentation error for multiple channels, it is useful for showing the location of prefilter center frequencies. In Fig. 6(e), several of the prefilter center frequencies in the multichannel design are located in dark areas corresponding to frequencies where a single Gabor prefilter would exhibit low segmentation error.

The segmentation error for the 6-channel segmentation in Fig. 6(d) is shown in Fig. 6(f) with black indicating incorrectly classified pixels. Even though the mixture distribution was employed to reduce edge error, the error in the vicinity of texture boundaries still appears to dominate in Fig. 6(f). Also, the corner error in the upper right corner of Fig. 6(f) is similar to the appearance of the error in Fig. 3.

Using *only four* Gabor filters, we achieve effective segmentation results as shown in Fig. 6(d). By comparison, Jain and Farrokhnia obtained similar segmentation results using 13 filters selected from a predetermined filter bank of 20 filters [4]. Randen and Husøy also achieved similar results using 13 to 40 filters [45].

An eight-texture segmentation is shown in Fig. 7. A sub-octave scaled set of σ_{g_j}'s with $\Sigma_g = \{2, 3, 4.5\}$ were used to compensate for the small texture regions in the 256×256 image. The two-filter segmentation in Fig. 7(b) is remarkably effective considering the relatively small regions in this 256×256 pixel image. The results of Fig. 7 approach limits where the size of the morphological operators begin to become significant relative to the sizes of the textured regions. Nevertheless, the results do illustrate the potential of the present methods.

The filter parameters used in the 8-texture segmentation of Fig. 7(d) are

$$\mathbf{\Theta}_\delta = \begin{bmatrix} 0.187 & 0 & 4.5 & 6.75 \\ 0.171 & 0.125 & 4.5 & 6.75 \\ -0.078 & 0.125 & 4.5 & 6.75 \\ 0.109 & 0 & 4.5 & 6.75 \\ -0.296 & 0.25 & 2 & 3 \\ 0 & 0.187 & 3 & 4.5 \end{bmatrix}$$

Extensive results are presented in [15] that show similarly effective segmentations for combinations of three to eight textures. These results also show a tendency for decreased segmentation error as the number of filter channels increases, and a tendency for increased segmentation error as the number of textures increases.

Table I lists the measured segmentation error as the number of textures varies from two through eight and as the number of filter channels varies from two to six for a variety of combinations of Brodatz and synthetic textures [15]. The table illustrates the effect of the number of textures and the effect of the number of filter channels on segmentation error. Note that the error stated in Table I, and the foregoing figures, includes the error at the perimeter of the 256 × 256 images. The perimeter is considered an equally valid texture boundary, since cyclic convolution is used for all filtering.

Two overall trends are apparent from the table. First, the segmentation error tends to decrease as the number of filters increases. This decrease is more noticeable as the number of textures increases. Secondly, the segmentation error tends to increase as the number of textures increases. Both of these trends are expected in that the segmentation should become less difficult as the number of features, or filters, increases and as the number of classes, or textures, decreases. Additional detail may be found in [15].

Experimental results that illustrate the complete system of Fig. 1 with a particular focus on the effects of the mixture-density classifier are shown in Fig. 8. The image in Fig. 8(a) consists of three Brodatz textures: an outermost region of uniform noise, a middle ring of "d15 - straw", and an innermost square region of "d77 - cotton canvas" [44]. Fig. 8(b) shows the segmentation error (misclassified pixels are the white regions) in $i_s(x,y)$ when the mixture density is *not* used; i.e., with Bayesian classification based on (19). Fig. 8(c) shows the segmentation error when the mixture density of (37) is used for classification. Morphological postprocessing was not altered in Figs. 8(b) and (c), so that only the effect of modifying the classifier is observed. The improvement near texture boundaries that is apparent in comparing Figs. 8(b) and (c) is confirmed by the reduction of total measured error from 10% to 4%.

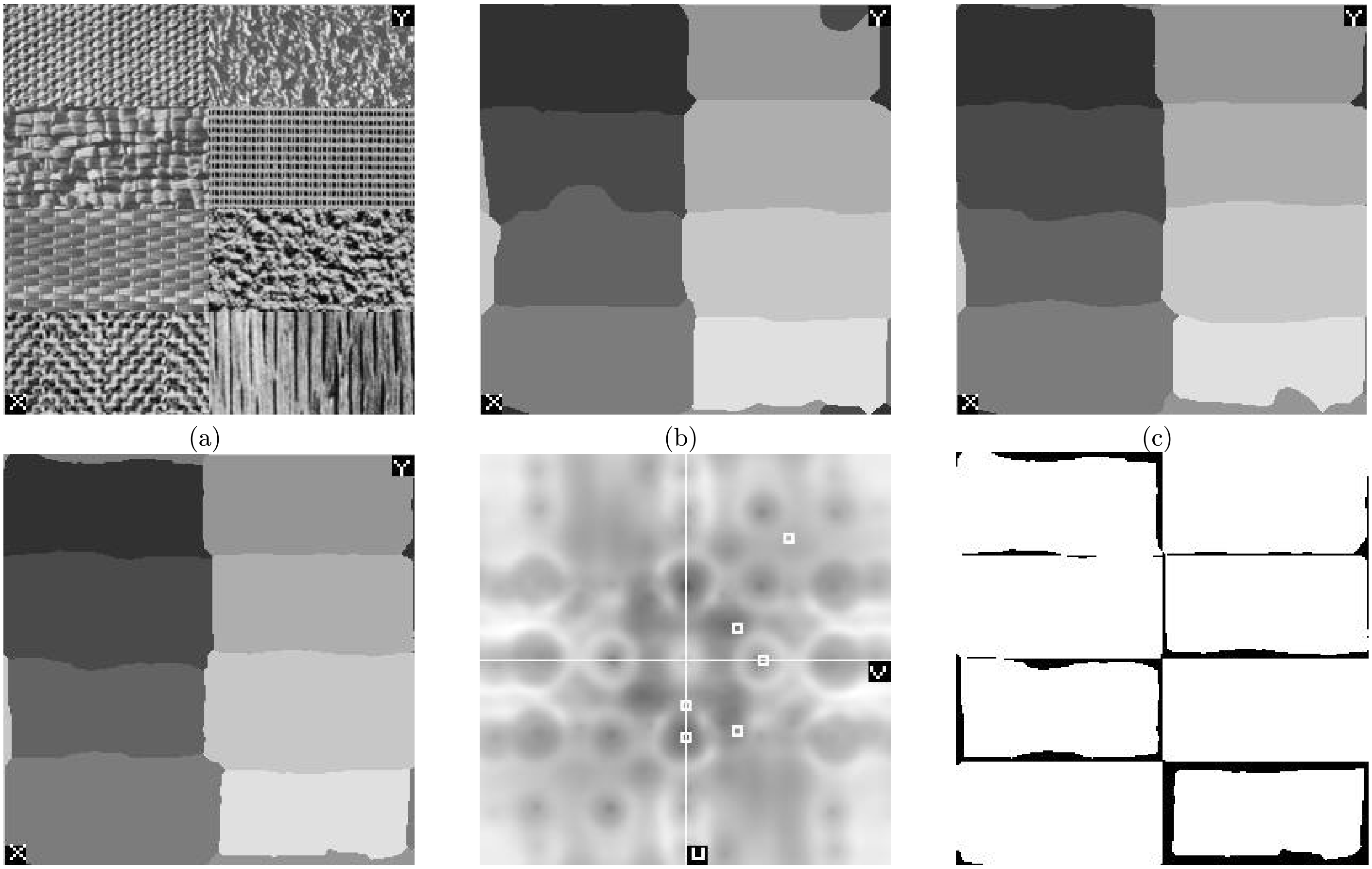
The results in Fig. 9 illustrate the effects of the morphological postprocessing on misclassifications at texture boundaries. The image in Fig. 9(a) consists of an outermost region of lowpass noise, a middle ring of "d21 - french canvas", and an innermost square region of "d55 - straw matting". Fig. 9(b) is the classifier output $c(x,y)$ using a mixture-density. A prominent band of misclassified pixels is seen along the entire boundary between the outermost texture (lowpass noise) and the middle ring of texture (d21). The misclassification appears to be caused by the trajectory of the feature-vector as it makes the transition between the two textures. During the transition between the two outermost textures, the vector appears to travel through a region in feature space that is assigned to the third texture (the texture at the center of the image). Finally, morphological postprocessing is applied to Fig. 9(b), resulting in the final segmented image $i_s(x,y)$ shown in Fig. 9(c). The mixture-density classifier was not altered in Figs. 9(b) and (c), so that only the effect of adding morphological postprocessing is observed. Comparing Fig. 9(b) and (c), the misclassified pixels at the texture boundary are mitigated by morphological postprocessing.

TABLE I
Measured segmentation error.

Number of Textures	Number of Channels				
	2	3	4	5	6
3	.07	.05	.04	.04	.03
3	.05	.05	.05	.05	.03
3	.05	.04	.04	.04	.04
3	.03	.03	.03	.03	.03
4	.08	.06	.06	.05	.04
4	.10	.06	.05	.04	.04
4	.08	.07	.03	.04	.04
5	.11	.09	.07	.06	.06
5	.17	.07	.06	.04	.04
5	.12	.08	.05	.06	.05
5	.07	.06	.05	.04	.04
5	.06	.05	.05	.05	.04
5	.13	.06	.05	.04	.04
5	.25	.08	.06	.06	.06
5	.29	.23	.20	.18	.18
8	.09	.07	.08	.08	.08
8	.45	.34	.23	.23	.19

VII. Discussion

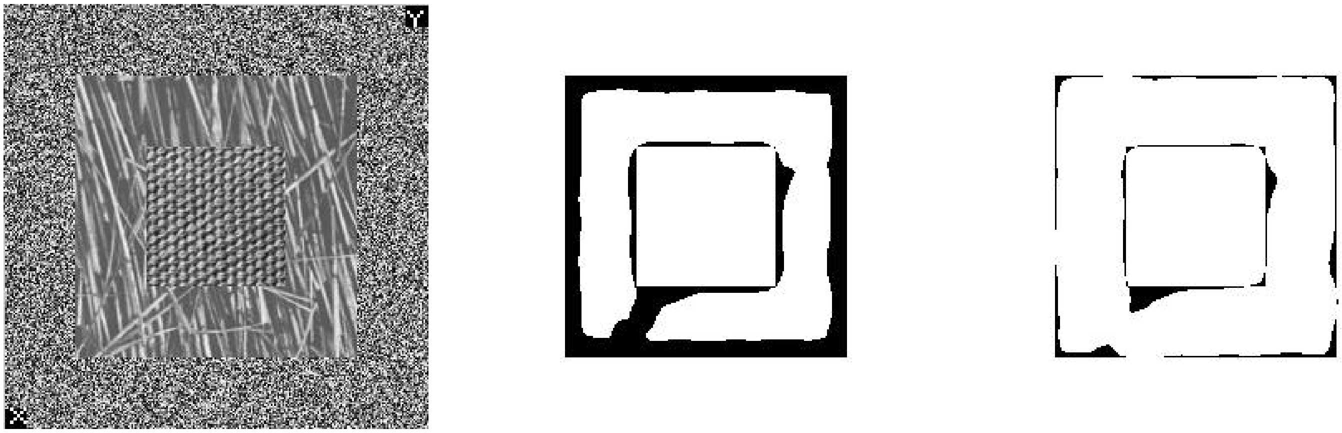
The multichannel paradigm and accompanying filter-design algorithm constitute a comprehensive treatment of the design of multiple Gabor filters to segment multiple textures. The multichannel paradigm provides relationships between predicted segmentation error $\mathcal{E}_t(\mathbf{\Theta}_{\boldsymbol{k}})$, Gabor prefilter parameters (u, v, σ_{g_j}), Gaussian postfilter parameters σ_{p_j}, multivariate output statistics $p_i(\boldsymbol{m_p}, \mathbf{\Theta}_{\boldsymbol{k}})$, and sample-texture power spectra $S_i(u,v)$. Using these relationships, we presented an algorithm for the design of a k-channel texture-segmentation system comprised of k Gabor prefilters and k Gaussian postfilters. In addition, we used the predicted multivariate output statistics as the basis for the design of a vector classifiefor the texture-segmentation system.



(a) (b) (c)

(d) (e) (f)

Fig. 7. Results for $\Sigma = \{2, 3, 4.5\}$, $\Lambda = \{1.5\}$. (a) Input composite 256 × 256 image, d77, d84, d55, d17, d24, d21, d57, d68. (b) 2 channel segmentation, error=0.09. (c) 4 channel segmentation, error=0.08. (d) 6 channel segmentation, error=0.08. (e) Filter frequencies (white squares) plotted on error map. (f) Segmentation error for six-channel segmentation, misclassified pixels in black.



(a) (b) (c)

Fig. 8. Effects of mixture distribution. (a) Input composite image, outer border = uniform noise, middle ring = d15 "straw," center square = d77 "cotton canvas." (b) Segmentation error for (a) without mixture distribution, black = misclassified pixel, error=0.13. (c) Segmentation error for (a) with mixture distribution, error=0.03.

Our experimental results confirm the efficacy of the filters designed with our algorithm. Further, our results confirm the multichannel paradigm, since both the designed filters and the Bayesian classifier are based on the multichannel paradigm. Results were presented that showed effective segmentations for 5 textures using only 4 filter channels and for 8 textures using 4 filter channels.

At some point, the number of filters in the multichannel design will approach the number of filters employed in a filter-decomposition approach or will approach complete coverage of the spatial-frequency plane. As the number of required filters increases, it may be better to use a filter-

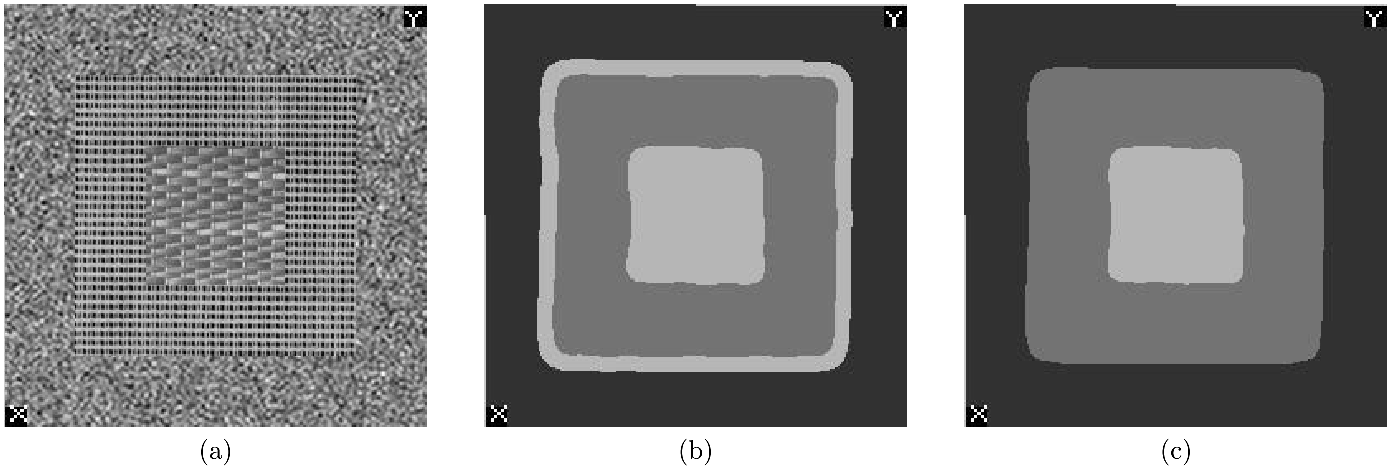


Fig. 9. Reduction of missclassifications near texture-boundaries using morphological postprocessing. (a) Input composite image, outer border = lowpass noise, middle ring = d21 "french canvas," center square = d55 "straw matting." (b) Output of mixture-density classifier $c(x, y)$ showing pronounced misclassification error at boundary between the outermost region of lowpass noise texture and the middle region of d21 texture. Note that the boundary is misclassified as the third (innermost) texture, d55. (c) Final segmentation after morphological postprocessing to remove localization error at texture boundaries, measured segmentation error = 0.05. Only two filter channels were used for the results in (b) and (c) with parameters ($u_j, v_j, \sigma_{g_j}, \sigma_{p_j}$) = (0,0.36,3,4.8) and (-0.17,0.41,3,4.8).

decomposition approach. Nevertheless, it may be advantageous to use the filters from the multichannel design algorithm in place of an *ad hoc* filter bank, since the designed filters are tailored to the segmentation problem.

There are several refinements that could be investigated to address difficult segmentation problems. First, the covariance matrices $\boldsymbol{C_i}$ in (**??**) can be estimated directly from the sample textures once the filters are chosen. The off-diagonal elements in $\boldsymbol{C_i}$ which were assumed zero for the purpose of designing the filter would then be replaced by values estimated from the sample textures and, therefore, a more accurate estimate of the multivariate Gaussian statistics would be formed.

The predominance of error at texture boundaries suggests that the primary limitation of the proposed method is localization error at texture boundaries. Boundary errors would also contribute to increased error for larger numbers of textures, since the total length of boundary increases as the number of textures increases. In situations where residual error appears to be within textured regions, an alternative approach would be to use a non-parametric classifier to attempt to handle outliers or secondary modes that could be causing misclassifications within textured regions.

Other approaches to the boundary localization issue include local inhibition and relaxation methods. Malik and Perona proposed local inhibitory connections for the suppression of spurious responses [29], [46]. Also, relaxation labeling methods described by Higgins *et al.* may offer some improvement [47].

The mixture-distribution modification to the global classifier appears to be an effective method for reducing localization error. Although spatially-local classification methods such as local inhibition are likely to be the best method to reduce the localization error, it may be advantageous to use the mixture-distribution modification prior to employing spatially-local classification methods. Thus, localization error can be reduced on a global basis before applying local inhibition, relaxation labeling, or other spatially-local classification techniques.

Previous researchers have investigated the development of optimal edge detectors in terms of a combined error measure including edge detection and edge localization [48]–[50]. However, there does not seem to be a similar treatment in the literature for the design of optimal vector classifiers for minimization of a combined localization and classification error measure. Bovik briefly treated the localization problem for a simple scalar decision process based on the filter with the maximum output level [3]. However, he did not combine a classification error with the localization error in his analysis. The methods presented here seem to be closely related to robust estimation methods; however, directly applicable works addressing localization and classification tradeoffs in multidimensional feature spaces have not been found [51].

Finally, the "n-ary erosion", "n-ary dilation", and "n-ary open" operations were quite useful in the treatment of the classifier output. Such operations do not appear to be in the morphology literature, and do not appear to be related to gray-scale morphology. The method seems to have an interesting parallel with local-competition approaches, in that the behavior at boundaries is similar to a local "winner-take-all" competition. In the case of the "erosion" operation, surrounding regions obliterate intervening regions smaller than the "erosion structuring element." We are not aware of a similar multi-class, or "n-ary", morphology, but portions of our procedure are similar to portions of boundary localization work by Yann and Young [41].

The previous discussion has focused on the design of Gabor filters. The methods, however, should be applicable to other types of filters. This may be accomplished by replacing the kernel $G(u, v)$ in (10) by a different kernel $K(u, v)$. The only caveat is that necessary properties of $G(u, v)$ used in the foregoing derivations set restrictions on $K(u, v)$. In particular, $K(u, v)$ should satisfy $[K(u, v)]^2 = |K(u, v)|^2$ as used in (10), and $K(u, v)$ should be non-negative to assure that the spectral estimate given by (10) is non-negative.

Acknowledgments

This work was partially supported by NIH FIRST award #CA53607 from the National Cancer Institute of the National Institutes of Health. Thanks also to W.E. Higgins for many helpful comments and improvements. Portions of this manuscript were presented elsewhere [12], [13].

References

[1] John G. Daugman, "Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 7, pp. 1169–1179, July 1988.

[2] Alan Conrad Bovik, Marianna Clark, and Wilson S. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 1, pp. 55–73, Jan. 1990.

[3] Alan C. Bovik, "Analysis of multichannel narrow-band filters for image texture segmentation," *IEEE Trans. Signal Processing*, vol. 39, no. 9, pp. 2025–2043, Sept. 1991.

[4] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognition*, vol. 23, no. 12, pp. 1167–1186, Dec. 1991.

[5] Alan C. Bovik, Nanda Gopal, Tomas Emmoth, and Alfredo Restrepo, "Localized measurements of emergent image frequencies by Gabor wavelets," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 691–711, Mar. 1992.

[6] Michael Unser, "Texture classification and segmentation using wavelet frames," *IEEE Trans. Image Proc.*, vol. 4, no. 11, pp. 1549–1560, Nov. 1995.

[7] D. F. Dunn and W. E. Higgins, "Optimal Gabor filters for texture segmentation," *IEEE Trans. Image Proc.*, vol. 4, no. 7, pp. 947–964, July 1995.

[8] D. Dunn, W. Higgins, and J. Wakeley, "Texture segmentation using 2-D Gabor elementary functions," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 2, pp. 130–149, Feb. 1994.

[9] Josef Bigun and J. M. Hans du Buf, "N-folded symmetries by complex moments in Gabor space and their application to unsupervised texture segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 1, pp. 80–87, Jan. 1994.

[10] Thomas P. Weldon, William E. Higgins, and Dennis F. Dunn, "Gabor filter design for multiple texture segmentation," *Optical Eng.*, vol. 35, no. 10, pp. 2852–2863, Oct. 1996.

[11] Thomas P. Weldon, William E. Higgins, and Dennis F. Dunn, "Efficient Gabor filter design for texture segmentation," *Pattern Recognition*, vol. 29, no. 12, pp. 2005–2015, Dec. 1996.

[12] Thomas Weldon and William Higgins, "Design of multiple Gabor filters for texture segmentation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Atlanta, GA, 7-10 May 1996, vol. IV, pp. 2245–2248.

[13] Thomas P. Weldon and William E. Higgins, "Integrated approach to texture segmentation using multiple Gabor filters," in *Proc. IEEE Int. Conf. on Image Processing*, Lausanne, Switzerland, 16-19 Sept. 1996, vol. III, pp. 955–958.

[14] Thomas P. Weldon, William E. Higgins, and Dennis F. Dunn, "Efficient Gabor filter design using Rician output statistics," in *Proc. IEEE Int. Symp. Circuits, Systems*, London, England, 30 May - 2 June 1994, vol. 3, pp. 25–28.

[15] Thomas P. Weldon, *Multiresolution Design of Multiple Gabor Filters for Texture Segmentation*, Ph.D. thesis, The Pennsylvania State University, 1995.

[16] T. N. Tan, "Texture edge detection by modelling visual cortical channels," *Pattern Recognition*, vol. 28, no. 9, pp. 1283–1298, Sept. 1995.

[17] Anil K. Jain and Sushil Bhattacharjee, "Text segmentation using Gabor filters for automatic document processing," *Machine Vision and Applications*, vol. 5, pp. 169–184, 1992.

[18] Dennis F. Dunn, Thomas P. Weldon, and William E. Higgins, "Extracting halftones from printed documents using texture analysis," *Optical Eng.*, vol. 36, no. 4, pp. 1044–1052, Apr. 1997.

[19] Touradj Ebrahimi and Murat Kunt, "Image compression by Gabor expansion," *Optical Eng.*, vol. 30, no. 7, pp. 873–880, July 1991.

[20] John G. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, no. 11, pp. 1148–1160, Nov. 1993.

[21] D. P. Casasent, J. S. Smokelin, and A. Ye, "Wavelet and Gabor transforms for detection," *Optical Eng.*, vol. 31, no. 9, pp. 1893–1898, Sept. 1992.

[22] David P. Casasent and John-Scott Smokelin, "Real, imaginary, and clutter Gabor filter fusion for detection with reduced false alarms," *Optical Eng.*, vol. 33, no. 7, pp. 2255–2263, July 1994.

[23] Boaz J. Super and Alan C. Bovik, "Localized measurement of image fractal dimension using Gabor filters," *Journal of Visual Comm. and Image Representation*, vol. 2, no. 2, pp. 114–128, June 1991.

[24] R. Mehrotra, K. R. Namuduri, and N. Ranganathan, "Gabor filter-based edge detection," *Pattern Recognition*, vol. 25, no. 12, pp. 1479–1493, Dec. 1992.

[25] Richard Buse, Zhi-Qiang Liu, and Terry Caelli, "Using Gabor filters to measure the physical parameters of lines," *Pattern Recognition*, vol. 29, no. 4, pp. 615–625, Apr. 1996.

[26] Moshe Porat and Yehoshua Zeevi, "The generalized Gabor scheme of image representation in biological and machine vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 4, pp. 452–468, July 1988.

[27] Trygve Randen and John H̊akon Husøy, "Novel approaches to multichannel filtering for image texture segmentation," in *Proc. SPIE Visual Comm. Image Processing 1994*, 1994, vol. 2094, pp. 626–636.

[28] M. R. Turner, "Texture discrimination by Gabor functions," *Biological Cybernetics*, vol. 55, pp. 71–82, 1986.

[29] Jitendra Malik and Pietro Perona, "Preattentive texture discrimination with early vision mechanisms," *J. Opt. Soc. Amer. A*, vol. 7, no. 5, pp. 923–932, May 1990.

[30] J. M. H. du Buf, "Abstract processes in texture discrimination," *Spatial Vision*, vol. 6, pp. 221–242, 1992.

[31] Tianhorng Chang and C. C. Jay Kuo, "Texture analysis and classification with tree-structured wavelet transform," *IEEE Trans. Image Proc.*, vol. 2, no. 4, pp. 429–441, Oct. 1993.

[32] Richard O. Duda and Peter E. Hart, *Pattern Classification and Scene Analysis*, John Wiley and Sons, 1973.

[33] Laveen Kanal, "Patterns in pattern recognition: 1968-1974," *IEEE Trans. Inform. Theory*, vol. 20, no. 6, pp. 697–722, Nov. 1974.

[34] Thomas P. Weldon and William E. Higgins, "Multiscale Rician approach to Gabor filter design for texture segmentation," in *Proc. IEEE Int. Conf. on Image Processing*, Austin, TX, 13-16 Nov. 1994, vol. II, pp. 620–624.

[35] S.O. Rice, "Mathematical analysis of random noise," *Bell System Tech. Journal*, vol. 23, pp. 282–333, July 1944.

[36] S.O. Rice, "Mathematical analysis of random noise," *Bell System Tech. Journal*, vol. 24, pp. 96–157, Jan. 1945.

[37] Mischa Schwartz, *Information Transmission, Modulation, and Noise*, McGraw-Hill, New York, NY, third edition, 1980.

[38] Dennis F. Dunn and William E. Higgins, "Optimal Gabor-filter design for texture segmentation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 1993, vol. V, pp. V37–V40.

[39] William K. Pratt, *Digital Image Processing*, John Wiley and Sons, second edition, 1991.

[40] D.G. Lainiotis, "A class of upper bounds on probability of error for multihypotheses pattern recognition," *IEEE Trans. Inform. Theory*, vol. 15, pp. 730–731, Nov. 1969.

[41] Stephan R. Yhann and Tzay Y. Young, "Boundary localization in texture segmentation," *IEEE Trans. Image Proc.*, vol. 4, no. 6, pp. 849–856, June 1995.

[42] Farshid Farrokhnia and Anil K. Jain, "A multi-channel filtering approach to texture segmentation," in *Proc. IEEE Computer Vision and Pattern Recognition Conf. Maui, Hawaii*, June 1991, pp. 364–370.

[43] Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, Addison-Wesley, New York, NY, 1992.

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

[45] Trygve Randen and John H åkon Husøy, "Multichannel filtering for image texture segmentation," *Optical Eng.*, vol. 33, no. 8, pp. 2617–2625, Aug. 1994.

[46] Jitendra Malik and Pietro Perona, "A computational model of texture segmentation," in *IEEE Computer Soc. Conf. on Computer Vision and Pattern Rec., Rosemont, IL*, 1989, pp. 326–332.

[47] W. E. Higgins, M. W. Hansen, and W. L. Sharp, "Interactive relaxation labeling for 3D cardiac image analysis," in *Proc. SPIE Biomedical Image Processing and Biomedical Visualization IV*, 1993, vol. 1905, pp. 51–62.

[48] John Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, no. 6, pp. 679–698, June 1986.

[49] H. D. Tagare and R. J. P. deFigueiredo, "On the localization performance measure and optimal edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 12, pp. 1186–1190, Dec. 1990.

[50] R. Kakarala and A. O. Hero, "On achievable accuracy in edge localization," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, no. 7, pp. 777–781, July 1992.

[51] Saleem A. Kassam and H. Vincent Poor, "Robust techniques for signal processing: A survey," *Proceedings of the IEEE*, vol. 73, no. 3, pp. 433–481, Mar. 1985.

[52] I. Bloch, "Fuzzy connectivity and mathematical morphology," *Pattern Recog. Lett.*, vol. 14, no. 6, pp. 483, June 1993.