# Unsupervised Texture Segmentation Using Gabor Filters[1]

Anil K. Jain
Department of Computer Science
Michigan State University
East Lansing, MI 48824-1027, U.S.A.
jain@cpswh.cps.msu.edu

Farshid Farrokhnia
Department of Electrical Engineering
Michigan State University
East Lansing, MI 48824-1226, U.S.A.
farrokh@pixel.cps.msu.edu

**Abstract**

We present a texture segmentation algorithm inspired by the multi-channel filtering theory for visual information processing in the early stages of human visual system. The channels are characterized by a bank of Gabor filters that nearly uniformly covers the spatial-frequency domain. We propose a systematic filter selection scheme which is based on reconstruction of the input image from the filtered images. Texture features are obtained by subjecting each (selected) filtered image to a nonlinear transformation and computing a measure of "energy" in a window around each pixel. An unsupervised square-error clustering algorithm is then used to integrate the feature images and produce a segmentation. A simple procedure to incorporate spatial adjacency information in the clustering process is also proposed. We report experiments on images with natural textures as well as artificial textures with identical 2nd- and 3rd-order statistics.

## 1 Introduction

Image segmentation is a difficult yet very important task in many image analysis or computer vision applications. Differences in the mean gray level or in color in small neighborhoods alone are not always sufficient for image segmentation. Rather, one has to rely on differences in the spatial arrangement of gray values of neighboring pixels — that is, on differences in texture. The problem of segmenting an image based on textural cues is referred to as *texture segmentation problem.*

The diversity of natural and artificial textures makes it impossible to give a universal definition of texture. A large number of techniques for analyzing image texture has been proposed in the past two decades [11, 22]. In this paper, we focus on a particular approach to texture analysis which is referred to as the *multi-channel filtering* approach. This approach is inspired by a multi-channel filtering theory for processing visual information in the early stages of the human visual system. First proposed by Campbell & Robson [4], the theory holds that the visual system decomposes the retinal image into a number of filtered images, each of which contains intensity variations over a narrow range of frequency (size) and orientation. The psychophysical experiments that suggested such a decomposition used various grating patterns as stimuli and were based on adaptation techniques [4]. Subsequent psychophysiological experiments provided additional evidence supporting the theory [10].

The multi-channel filtering approach to texture analysis is intuitively appealing because the dominant spatial-frequency components of different textures are different. An important advantage of the multi-channel filtering approach to texture analysis is that one can use simple statistics of gray values in the filtered images as texture features. This simplicity is the direct result of decomposing the original image into several filtered images with limited spectral information. The main issues involved in the multi-channel filtering approach to texture analysis are: 1) functional characterization of the channels and the number of channels, 2) extraction of appropriate texture features from the filtered images. 3) the relationship between channels (dependent vs. independent), and 4) integration of texture features from different channels to produce a segmentation. Different multi-channel filtering techniques that are proposed in the literature differ in their approach to one or more of the above issues.

We use a bank of Gabor filters to characterize the channels. We show that the filter set forms an approximate basis for a wavelet transform, with the Gabor function as the wavelet. We propose a systematic filter selection scheme which is based on reconstruction of the input image from the filtered images. Each (selected) filtered image is subjected to a bounded nonlinear transformation that behaves as a 'blob detector'. The combination of multi-channel filtering and the nonlinear stages can be viewed as performing a multi-scale blob detection. Texture discrimination is associated with differences in the attributes of these blobs in different regions. A statistical approach is then used where the attributes of the blobs are captured by texture features defined by a measure of "energy" in a small window around each pixel in each response image. This process generates one 'feature image' corresponding to each filtered image (see Figure 1). The size of the window for each response image is determined using a simple formula involving the radial frequency to which the corresponding filter is tuned. A square-error clustering algorithm is then used to identify the texture categories. A simple procedure for inclusion of contextual (spatial adjacency) information in the clustering process is also proposed.

## 2 Channel Characterization

We represent the channels with a bank of two-dimensional Gabor filters. A two-dimensional Gabor function consists of a sinusoidal plane wave of some frequency and orientation, modulated by a two-dimensional Gaussian envelope. A 'canonical' Gabor filter in the spatial domain is given by

$$h(x,y) = \exp\left\{-\frac{1}{2}\left[\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right]\right\} \cos(2\pi u_0 x + \phi), \quad (1)$$

[1]This work was supported in part by the National Science Foundation infrastructure grant CDA-8806599, and by a grant from E. I. Du Pont De Nemours & Company Inc.

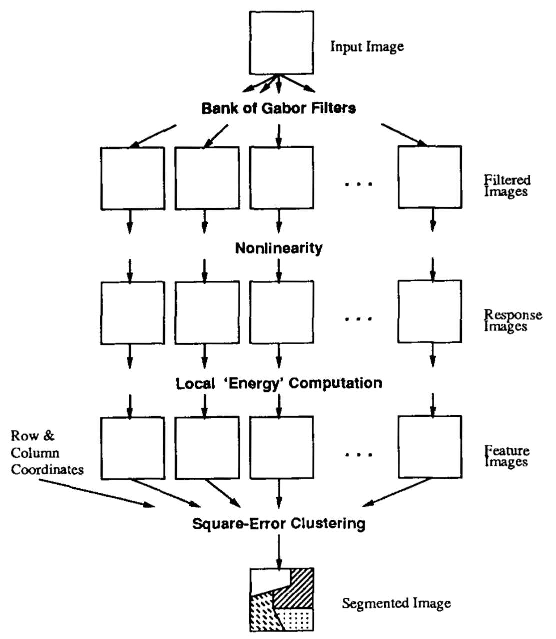

Figure 1: An overview of the texture segmentation algorithm.

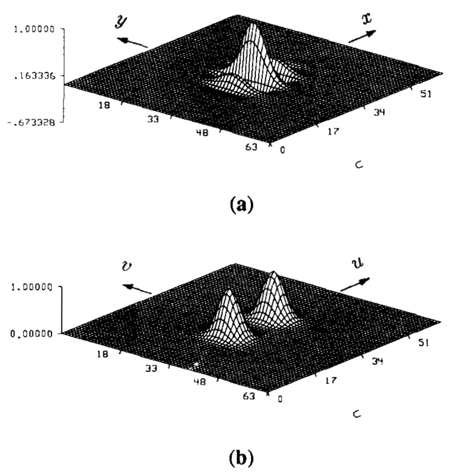

Figure 2: (a) An even-symmetric Gabor filter in the spatial domain. (b) Corresponding MTF. The origin is at $(r, c) = (32, 32)$.

where $u_0$ and $\phi$ are the frequency and phase of the sinusoidal plane wave along the $x$-axis (i.e. the 0° orientation), and $\sigma_x$ and $\sigma_y$ are the space constants of the Gaussian envelope along the $x$- and $y$-axis, respectively. A Gabor filter with arbitrary orientation, $\theta_0$, can be obtained via a rigid rotation of the $x$-$y$ coordinate system. These two-dimensional functions have been shown to be good fits to the receptive field profiles of simple cells in the striate cortex [18, 7].

The frequency- and orientation-selective properties of a Gabor filter are more explicit in its frequency domain representation. With $\phi = 0$, the Fourier transform of the Gabor function in (1) is real-valued and given by

$$H(u,v) = A\left(\exp\left\{-\frac{1}{2}\left[\frac{(u-u_0)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]\right\} + \exp\left\{-\frac{1}{2}\left[\frac{(u+u_0)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]\right\}\right), \qquad (2)$$

where $\sigma_u = 1/2\pi\sigma_x$, $\sigma_v = 1/2\pi\sigma_y$, and $A = 2\pi\sigma_x\sigma_y$. The Fourier domain representation in (2) specifies the amount by which the filter modifies or modulates each frequency component of the input image. Such representations are, therefore, referred to as modulation transfer functions (MTF). Figure 2 shows an even-symmetric Gabor filter and its MTF in a $64 \times 64$ array.

Texture segmentation requires simultaneous measurements in both the spatial and the spatial-frequency domains. Filters with smaller bandwidths in the spatial-frequency domain are more desirable because they allows us to make finer distinctions among different textures. On the other hand, accurate localization of texture boundaries requires filters that are localized in the spatial domain. However, the effective width of a filter in the spatial domain and its bandwidth in the spatial-frequency domain are inversely related. An important property of Gabor filters is that they have optimal joint localization, or resolution, in both the spatial and the spatial-frequency domains [8].

The use of Gabor filters in texture analysis is not new. For example, Turner [21] used a fixed set of Gabor filters and demonstrated their potential for texture discrimination. Similarly, Perry & Lowe [19] use a fixed set of Gabor filters in their texture segmentation algorithm. Bovik *et al.* [1] have used complex Gabor filters, where the real part of each filter is an even-symmetric Gabor filter (i.e., $\phi = 0$) and the imaginary part is an odd-symmetric Gabor filter (i.e., $\phi = \pi/2$). Instead of using a fixed set of filters, Bovik *et al.* apply a simple peak finding algorithm to the power spectrum of the image in order to determine the radial frequencies of the appropriate Gabor filters. In our texture segmentation algorithm, we model the channels with a *fixed* set of Gabor filters that preserve almost all the information in the input image.

## 2.1 Choice of Filter Parameters

We implement each Gabor filter as a discrete realization of the MTF in (2). We use four values of orientation $\theta_0$: 0°, 45°, 90°, and 135°. For an image array with a width of $N_c$ pixels, where $N_c$ is a power of 2, the following values of radial frequency $u_0$ are used:

$$1\sqrt{2},\ 2\sqrt{2},\ 4\sqrt{2},\ \cdots,\ \text{and}\ (N_c/4)\sqrt{2} \qquad \text{cycles/image-width}$$

Note that the radial frequencies are 1 octave apart. (The frequency bandwidth, in octaves, from frequency $f_1$ to frequency $f_2$ is given by $\log_2(f_2/f_1)$.) We let the orientation and frequency bandwidths of each filter be 45° and 1 octave, respectively. Several experiments have shown that the frequency bandwidth of simple cells in the visual cortex is about 1 octave [20]. Psychophysical experiments show that the resolution of the orientation tuning ability of the human visual system is as high as 5 degrees. Therefore, in general, finer quantization of orientation will be needed. The restriction to four orientations is made for computational efficiency.

The above choice of the radial frequencies, guarantees that the passband of the filter with the highest radial frequency, *viz.* $(N_c/4)\sqrt{2}$ cycles/image-width, falls inside the image array. For an image with 256 columns, for example, a total of 28 filters can be used — 4 orientations and 7 frequencies. Note that filters with very low radial frequencies (e.g., $1\sqrt{2}$ and $2\sqrt{2}$ cycles/image-width) can often be left out, because they capture spatial variations that are too large to correspond to texture. To assure that the filters

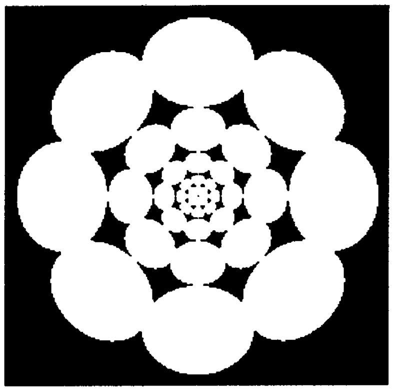

Figure 3: The filter set in the spatial-frequency domain (256 × 256). There are a total of 28 Gabor filters. Only the half-peak support of the filters are shown. The origin is at (row,col) = (128,128).

do not respond to regions with constant intensity, we have set the MTF of each filter at $(u, v) = (0,0)$ to zero. As a result each filtered image has a zero mean.

The set of filters used in our algorithm results in nearly uniform coverage of the frequency domain (Figure 3). This filter set constitutes an approximate basis for a wavelet transform, with the Gabor filter as the wavelet. The wavelet transform is closel) related to the window Fourier transform. However, unlike window Fourier transforms where the window is fixed, in a wavelet transform the window size is allowed to change according to frequency [17]. Intuitively, a wavelet transform can be interpreted as a band-pass filtering operation on the input image. The Gabor function is an admissible wavelet, however, it does not result in an orthogonal decomposition. This means that a wavelet transform based on Gabor wavelets is redundant. A decomposition obtained by our filter set is nearly orthogonal, as the amount of overlap between the filters (in the spatial-frequency domain) is small.

Figure 4 shows examples of filtered images for an image containing 'straw matting' (D55) and 'wood grain' (D68) textures from the photographic album of textures by Brodatz [2]. The ability of the filters to exploit differences in spatial-frequency (size) and orientation in the two textures is evident in these images. The differences in the strength of the responses in regions with different textures is the key to the multi-channel approach to texture analysis. To maximize visibility, each filtered image has been scaled to full contrast.

## 2.2 Filter Selection

Using only a subset of the filtered images can reduce the computational burden at later stages, because this directly translates into a reduction in the number of texture features. Let $s(x, y)$ be the reconstruction of the input image obtained by adding all the filtered images. Let $\hat{s}(x, y)$ be the *partial* reconstruction of the image obtained by adding a given subset of filtered images. The error involved in using $\hat{s}(x, y)$ instead of $s(x, y)$ can be measured by

$$SSE = \sum_{x,y} [\hat{s}(x,y) - s(x,y)]^2$$

The fraction of intensity variations in $s(x, y)$ that is explained by $\hat{s}(x, y)$ can be measured by the coefficient of determination

$$R^2 = 1 - \frac{SSE}{SSTOT},$$

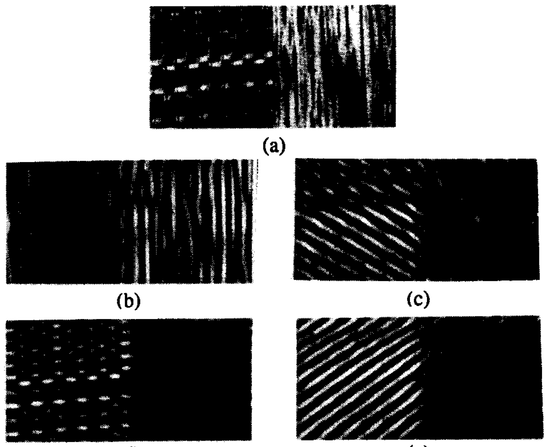

Figure 4: Examples of filtered images for *D55-D68* texture pair (128×256). (a) Input image. (b-e) Filtered images corresponding to Gabor filters tuned to $16\sqrt{2}$ c/i-w and to 0°, 45°, 90°, and135°, respectively.

where

$$SSTOT = \sum_{x,y} s(x,y)^2.$$

Note that $s(x, y)$ has a mean of zero, since the mean gray value of each filtered image is zero.

For computational efficiency, we determine the "best" subset oı .he filtered images (filters) by the following suboptimal sequential forward selection procedure:

1. Select the filtered image that best approximates $s(x, y)$, i.e. results in the highest value of $R^2$.
2. Select the next filtered image that together with previously selected filtered image(s) best approximate $s(x, y)$.
3. Repeat Step 2 until $R^2 \geq 0.95$.

A minimum value of 0.95 for $R^2$ means that we will use only as many filtered images as necessary to account for at least 95% of the intensity variations in $s(x, y)$.

Let $r_i(x, y)$ be the $i^{\text{th}}$ filtered image and $R_i(u, v)$ be its Discrete Fourier Transform. The amount of overlap between the MTFs of the Gabor filters in our filter set is small. Therefore, the total energy $E$ in $s(x, y)$ can be approximated by

$$E \approx \sum_{i=1}^{n} E_i,$$

where

$$E_i = \sum_{x,y} [r_i(x,y)]^2 = \sum_{u,v} |R_i(u,v)|^2.$$

Now, it is easily verified that for any subset $\mathcal{A}$ of filtered images

$$R^2 \approx \frac{\sum_{j \in \mathcal{A}} E_j}{E}$$

An *approximate* filter selection would then consists of computing $E_i$ for $i = 1, \cdots, n$. These energies can be computed in the Fourier domain, hence avoiding unnecessary inverse Fourier transforms. We then sort the filters (channels) based on their energy and pick as many filters as needed to achieve $R^2 \geq 0.95$. Computationally, this procedure is much more efficient than the sequential forward selection procedure described before.

## 3 Computing Feature Images

We use the following procedure to compute features from each filtered image. First, each filtered image is subjected to a nonlinear transformation. Specifically, we use the following bounded nonlinearity

$$\psi(t) = \tanh(\alpha t) = \frac{1 - e^{-2\alpha t}}{1 + e^{-2\alpha t}}, \qquad (3)$$

where $\alpha$ is a constant. This nonlinearity bears certain similarities to the sigmoidal activation function used in artificial neural networks. In our experiments, we have used an empirical value of $\alpha = 0.25$ which results in a rapidly saturating, threshold-like transformation. As a result, the application of the nonlinearity transforms the sinusoidal modulations in the filtered images to square modulations and, therefore, can be interpreted as a blob detector. However, the detected blobs are not binary, and unlike the blobs detected by Voorhees & Poggio [23] they are not necessarily isolated from each other. Also, since each filtered image has a zero mean and the nonlinearity in (3) is odd-symmetric, both dark and light blobs are detected.

Instead of identifying individual blobs and measuring their attributes, we simply compute the average absolute deviation (AAD) from the mean in small overlapping windows. This is similar to the 'texture energy' measure that was first proposed by Laws [15]. Formally, the feature image $e_k(x,y)$ corresponding to filtered image $r_k(x,y)$ is given by

$$e_k(x,y) = \frac{1}{M^2} \sum_{(a,b)\in W_{xy}} |\psi(r_k(a,b))|, \qquad (4)$$

where $\psi(\cdot)$ is the nonlinear function in (3) and $W_{xy}$ is an $M \times M$ window centered at the pixel with coordinates $(x,y)$.

The size, $M$, of the averaging window in (4) is an important parameter. More reliable measurement of texture features calls for larger window sizes. On the other hand, more accurate localization of region boundaries calls for smaller windows. Furthermore, using Gaussian weighted windows, rather than unweighted windows, is likely to result in more accurate localization of texture boundaries. For each filtered image we use a Gaussian window whose space constant $\sigma$ is proportional to the *average* size of the intensity variations in the image. For a Gabor filter with radial frequency $u_0$ this average size is given by

$$T = N_c/u_0 \qquad \text{pixels}, \qquad (5)$$

where $N_c$ is the width (number of columns) of the image. We found a $\sigma \approx 0.5T$ to be appropriate in most of our segmentation experiments.

## 4 Integrating Feature Images

Having obtained the feature images, the main question is how to integrate features corresponding to different filters to produce a segmentation. Let's assume that there are K texture categories, $C_1, \ldots, C_K$, present in the image. If our texture features are capable of discriminating these categories then the patterns belonging to each category will form a cluster in the feature space which is "compact" and "isolated" from clusters corresponding to other texture categories. Pattern clustering algorithms are ideal vehicles for *recovering* such clusters in the feature space. In our texture segmentation experiments we use a square-error clustering algorithm known as CLUSTER [13]. Prior to clustering each feature is normalized to have a zero mean and a constant variance. This normalization is intended to avoid the domination of features with small numerical ranges by those with larger ranges.

Clustering a large number of patterns becomes computationally demanding. Therefore, we first cluster a small randomly selected subset of patterns into a specified number of clusters. Patterns in each cluster are given a generic category label that distinguishes them from those in other clusters. These labeled patterns are then used as training patterns to classify patterns (pixels) in the entire image using a minimum distance classifier.

In texture segmentation, neighboring pixels are very likely to belong to the same texture category. We propose a simple method that incorporates the spatial adjacency information directly in the clustering process. This is achieved by including the spatial coordinates of the pixels as two additional features.

## 5 Experimental Results

We now apply our texture segmentation algorithm to several images in order to demonstrate its performance. These images are created by collaging subimages of natural as well as artificial textures. We start by a total of 20 Gabor filters in each case. Each filter is tuned to one of the four orientations and to one of the five highest radial frequencies. For an image with a width of 256 pixels, for example, $4\sqrt{2}, 8\sqrt{2}, 16\sqrt{2}, 32\sqrt{2}$, and $64\sqrt{2}$ cycles/image-width radial frequencies are used. We then use our filter selection scheme to determine a subset of filtered images that achieves an

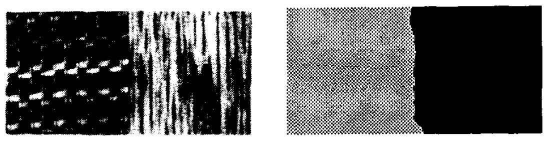

Figure 5: (a) *D55-D68* texture pair. (b) Two-category segmentation obtained using a total of 13 Gabor filters.

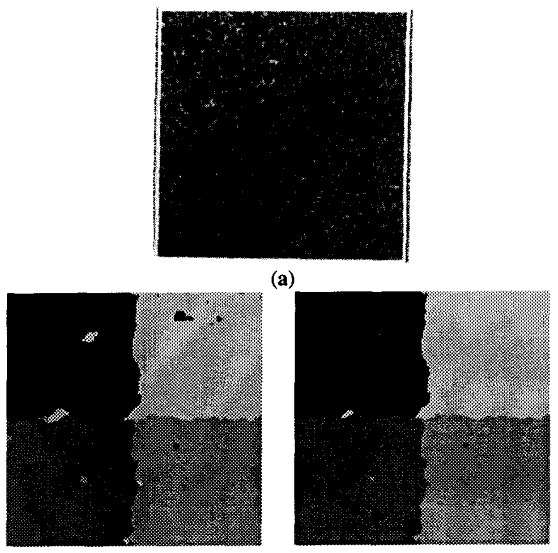

Figure 6: (a) A $256 \times 256$ image containing four different Gaussian Markov random field textures. (b) Four-category segmentation obtained using a total of 11 Gabor filters. (c) Same as **b**, but with pixel coordinates used as additional features.

$R^2$ value of at least 0.95. The number of randomly selected feature vectors, that are used as input to the clustering program, is proportional to the size of the input image. For a $256 \times 256$ image, for example, 4000 patterns are selected at random, which is about 6% of the total number of patterns. The same percentage is used in all the following experiments. The segmentation results are displayed as gray-level images, where regions belonging to different categories are shown with different gray levels.

Figure 5 shows the segmentation results for the *D55-D68* texture pair. The algorithm successfully discriminates the two textured regions and detects the boundary between them quite accurately. The segmentation with pixel coordinates included as additional features was essentially the same for this example.

Figure 6 (a) shows a $256 \times 256$ image containing four different Gaussian Markov random field (GMRF) textures generated using non-causal finite lattice models [5]. These four textures can not be discriminated based on their mean gray values. The four-category segmentation of the image is shown in Figure 6 (b). The segmentation improves considerably when pixel coordinates are used as additional features (Figure 6 (c)).

Figure 7 (a) shows another $256 \times 256$ image containing natural textures D77, D55, D84, D17, and D24 from the Brodatz album. The five-category segmentation of this image, using 13 Gabor filters and the pixel coordinates, is shown in Figure 7 (b).

Figure 8 (a) shows a $512 \times 512$ image containing sixteen natural textures, also from the Brodatz album. Our filter selection (with a threshold of 0.95 for $R^2$) indicated that only 14 filtered images are sufficient. However, the resulting segmentation was not very good. Using all 20 filtered images (and the pixel coordinates) we obtained the 16-category segmentation in Figure 8 (b). Recall that the fitting criterion in our filter selection scheme is computed globally over the entire image. A larger threshold for $R^2$ should, therefore, be used when one or more texture categories occupy a small fraction of the image.

Figure 9 shows the segmentation of a number of texture pair images that have been used in the psychophysical studies of texture perception [13]. The two textures in the 'L_and_+' texture pair have identical power spectra. The textures in the 'even-odd'

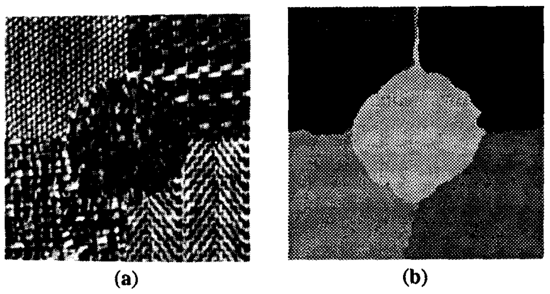

Figure 7: (a) A $256 \times 256$ image containing five natural textures (D77, D55, D84, D17, and D24) from the Brodatz album. (b) Five-category segmentation obtained using a total of 13 Gabor filters and the pixel coordinates.

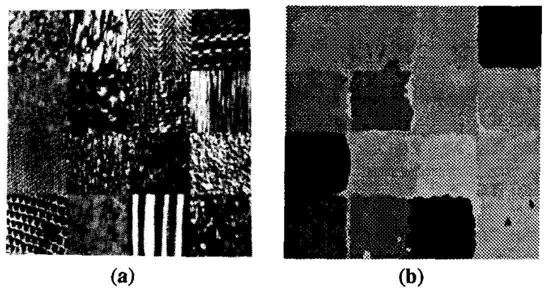

Figure 8: (a) A $512 \times 512$ image containing sixteen natural textures (D29, D12, D17, D55; D32, D5, D84, D68; D77, D24, D9, D4; D3, D33, D51, D54) from the Brodatz album (b) 16-category segmentation obtained using a total of 20 Gabor filters and the pixel coordinates.

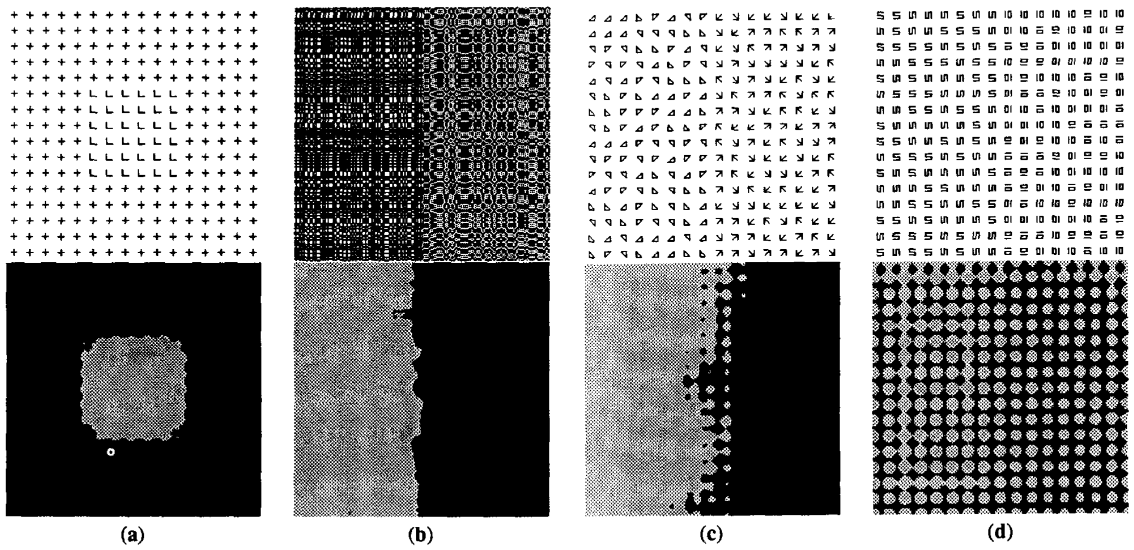

Figure 9: Segmentation of texture pairs that have been used in the psychophysical studies of texture perception. All images are $256 \times 256$. The number of Gabor filters used varied between 8 – 11. (a) 'L_and_+'. (b) 'even-odd'. (c) 'triangle-arrow' (d) 'S-IO'.

| Input Image | | | Misclassified (%) | |
|---|---|---|---|---|
| name | size | cats | without (r,c) | with (r,c) |
| D55-D68 | $128 \times 256$ | 2 | 0.61 | 0.58 |
| GMRF | $256 \times 256$ | 4 | 2.68 | 1.78 |
| Five | $256 \times 256$ | 5 | 4.87 | 2.96 |
| Sixteen | $512 \times 512$ | 16 | 12.85 | 7.47 |
| L_and_+ | $256 \times 256$ | 2 | 2.21 | 2.21 |
| Tri-Arr | $256 \times 256$ | 2 | 6.12 | 5.20 |
| Even-Odd | $256 \times 256$ | 2 | 3.05 | 2.60 |

Table 1: Percentage of misclassified pixels.

texture pair have identical third-order ~~order~~ statistics. The textures in the 'tri-arr' and 'S-IO' texture pairs, on the other hand, have identical second-order statistics. The 'even-odd' and 'tri-arr' textures are two of the counter-examples to the original Julesz conjecture that texture pairs with identical second-order statistics cannot be preattentively discriminated [14]. While the first three texture pairs in Figure 9 are easily discriminated, the 'S-IO' texture pair is not *preattentively* discriminable. Our algorithm appears to perform as predicted by preattentive texture discrimination by human — the algorithm successfully segments the first three texture pairs, but fails to do so for the 'S-IO' texture pair.

The lack of appropriate quantitative measures of the goodness of a segmentation makes it very difficult to evaluate and compare texture segmentation algorithms. One simple criterion that is often used is the percentage of misclassified pixels. Table 1 gives the percentage of misclassified pixels for the segmentation experiments reported here. As seen in this table, there is a clear advantage in using the pixel coordinates (spatial information) as additional features.

## 6 Summary and Conclusions

We have presented an unsupervised texture segmentation algorithm that uses a fixed set of Gabor filters. Our choice of Gabor filters and their parameters were motivated by a goal to construct an approximate basis for a wavelet transform. The use of a nonlinear transformation following the linear filtering operations has been suggested as one way to account for inherently nonlinear nature of biological visual systems [9]. We argued that the localized filtering by our Gabor filter set followed by a nonlinear transformation can be interpreted as a multi-scale blob detection operation.

One of the limitations of our texture segmentation algorithm is the lack of a criterion for choosing the value of $\alpha$ in the nonlinear transformation. Also, the algorithm assumes that different channels are independent from each other. However, there is psychophysical and physiological evidence indicating inhibitory interactions between different spatial frequency channels [9]. Allowing inhibitory interactions among the channels is shown to have the potential to reduce the effective dimensionality of the feature space [3].

In our texture segmentation algorithm, we assumed that the number of texture categories is given. The pattern clustering technique that is used by our segmentation algorithm will produce a clustering with the desired number of clusters, even if it does not make "sense". We believe that an integrated approach that uses both a region-based *and* an edge-based segmentation can be used to resolve the question of determining the number of texture categories. Malik & Perona [16], for example, have developed a multi-channel filtering technique that produces edge-based segmentations. The basic idea is to generate an oversegmented solution using our region-based texture segmentation algorithm. Then, based on the "evidence" for an edge provided by the edge-based segmentation, one can test the validity of boundaries between regions in the oversegmented solution. This integrated approach is currently being investigated.

## References

[1] Bovik, A.C., Clark, M., and Geisler, W.S. (1990), "Multichannel Texture Analysis Using Localized Spatial Filters," *IEEE Trans. PAMI*, vol. 12, no. 1, pp. 55-73.

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

[3] Caelli, T.M. (1988), "An Adaptive Computational Model for Texture Segmentation," *IEEE Trans. SMC*, vol. 18, no. 1, pp. 9-17.

[4] Campbell, F.W. and Robson, J.G. (1968), "Application of Fourier Analysis to the Visibility of Gratings," *J. Physiology*, vol. 197, pp. 551-566.

[5] Chellappa, R., Chatterjee, S., and Bagdazian, R. (1985), "Texture Synthesis and Compression Using Gaussian-Markov Random Field Models," *IEEE Trans. SMC*, vol. 15, no. 2, pp. 298-303.

[6] Coggins, J.M. and Jain, A.K. (1985), "A Spatial Filtering Approach to Texture Analysis," *Pattern Recognition Letters*, vol. 3, no. 3, pp. 195-203.

[7] Daugman, J.G. (1980), "Two-dimensional Spectral Analysis of Cortical Receptive Field Profiles," *Vision Research*, vol. 20, pp. 847-856.

[8] Daugman, J.G. (1985), "Uncertainty Relation for Resolution in Space, Spatial-Frequency, and Orientation Optimized by Two-dimensional Visual Cortical Filters," *J. OSA*, vol. 2, no. 7, pp. 1160-1169.

[9] De Valois, R.L. and De Valois, K.K. (1988), *Spatial Vision*, Oxford University Press.

[10] De Valois, R.L., Albrecht, D.G., and Thorell, L.G. (1982), "Spatial-Frequency Selectivity of Cells in Macaque Visual Cortex," *Vision Research*, vol. 22, pp. 545-559.

[11] Haralick, R.M. (1979), "Statistical and Structural Approaches to Texture," *Proc. IEEE*, vol. 67, no. 5, pp. 786-804.

[12] Jain, A.K. and Dubes, R.C. (1988), *Algorithms for Clustering Data*, Prentice-Hall, New Jersey.

[13] Julesz, B. and Bergen, J.R. (1983), "Textons, The fundamental Elements in Preattentive Vision and Perception of Textures," *Bell Syst. Tech. J.*, vol. 62, no. 6, pp. 1619-1645.

[14] Julesz, B., Gilbert, E.N., Shepp, L.A., and Frisch, H.L. (1973), "Inability of Humans to Discriminate Between Visual Textures That Agree in Second-Order Statistics — Revisited," *Perception*, vol. 2, pp. 391-405.

[15] Laws, K.I. (1980), *Textured Image Segmentation*, Tech. Rep. USCIPI-940, Image Process. Inst., Univ. of Southern California.

[16] Malik, J. and Perona, P. (1990), "Preattentive Texture Discrimination with Early Vision Mechanisms," *J. OSA: A*, vol. 7, no. 5, pp. 923-932.

[17] Mallat, S.G. (1989), "Multifrequency Channel Decomposition of Images and Wavelet Models," *IEEE Trans. ASSP*, vol. 37, no. 12, pp. 2091-2110.

[18] Marcelja, S. (1980), "Mathematical Description of the Responses of Simple Cortical Cells," *J. OSA*, vol. 70, pp. 1297-1300.

[19] Perry, A. and Lowe, D.G. (1989), "Segmentation of Textured Images," *Proc. CVPR*, pp. 326-332.

[20] Pollen, D.A. and Ronner, S.F. (1983), "Visual Cortical Neurons as Localized Spatial Frequency Filters," *IEEE Trans. SMC*, vol. 13, no. 5, pp. 907-916.

[21] Turner, M.R. (1986), "Texture Discrimination by Gabor Functions," *Biol. Cybern.*, vol. 55, pp. 71-82.

[22] Van Gool, L., Dewaele, P., and Oosterlinck, A. (1985), "Texture Analysis Anno 1983," *CVGIP*, vol. 29, pp. 336-357.

[23] Voorhees, H. and Poggio, T. (1988), "Computing Texture Boundaries from Images," *Nature*, vol. 333, no. 6171, pp. 364-367.