Pattern Recognition Letters 3 (1985) 195-203
North-Holland

May 1985

A spatial filtering approach to texture analysis

James M. COGGINS
Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA 01609, USA

Anil K. JAIN
Department of Computer Science, Michigan State University, East Lansing, MI 48824, USA

Received 11 October 1984
Revised 18 January 1985

Abstract: A theory regarding the information processing strategies in human vision motivates the development of a texture feature space. The feature space is used in texture classification and segmentation problems. A statistic for determining the number of different textures in the image is developed and demonstrated.

Key words: Texture, spatial filtering, human vision, segmentation, classification, clustering.

1. Introduction

Texture is generally recognized as being fundamental to the perception of regions and surfaces in images. The potential importance of visual texture for automatic vision systems has inspired many attempts to develop algorithms which characterize the texture of an image region [10]. These efforts have been hampered by the lack of a precise definition or characterization of texture.

Human observers are capable of some image segmentation and discrimination tasks under conditions (such as brief exposure to a test image) which prevent detailed scrutiny of the image. This ability is referred to as 'effortless' or 'preattentive' visual discrimination. In this sense, two images (which do not portray particular objects or forms) are said to have the same 'texture' if they are not effortlessly discriminable to human observers, regardless of the image generation procedures. The characterization of texture in terms of human visual performance is intuitively satisfying, since the goal of texture analysis algorithms is clearly the duplication of preattentive visual performance and since human vision can serve as a convenient performance evaluation standard for texture algorithms. Unfortunately, this characterization of texture does not suggest an algorithm which can duplicate human texture perception. This paper will present a texture analysis algorithm which is compatible with the characterization of texture in terms of human visual performance. The performance of the algorithm on texture classification and segmentation tasks will be demonstrated.

2. A texture analysis method based on a theory of human vision

In the late 1960's, researchers found that the threshold visibility of sinusoidal gratings depends on the spatial frequency of the gratings [2, 3]. The potential power of Fourier analysis as a tool for studying human vision was noted immediately [3]. Further research suggested that the analysis of a stimulus by the visual system might involve a set of quasi-independent mechanisms, called channels,

0167-8655/85/$3.30

which could be conveniently characterized in the spatial frequency domain [7].

The channels in the visual system were modelled by filters defined in the spatial frequency domain with characteristics based on the results of psychophysical and neurological studies. Thus, the decomposition of an image in early stages of the visual system is modelled by a sequence of spatial domain filtered images [7]. These studies did not develop algorithms for automatic image analysis, but other papers (e.g. [8, 9, 11, 12, 14–17]) suggested that filtering in spatial frequency channels might be useful for machine pattern recognition, including texture analysis. Several different models for channels in human vision have been developed and some arguments against a Fourier model of vision have been advanced (see reviews in [4, 7]).

The filters used in this study are adapted from Ginsburg [7]. The transfer functions for spatial frequency channels are defined by a Gaussian

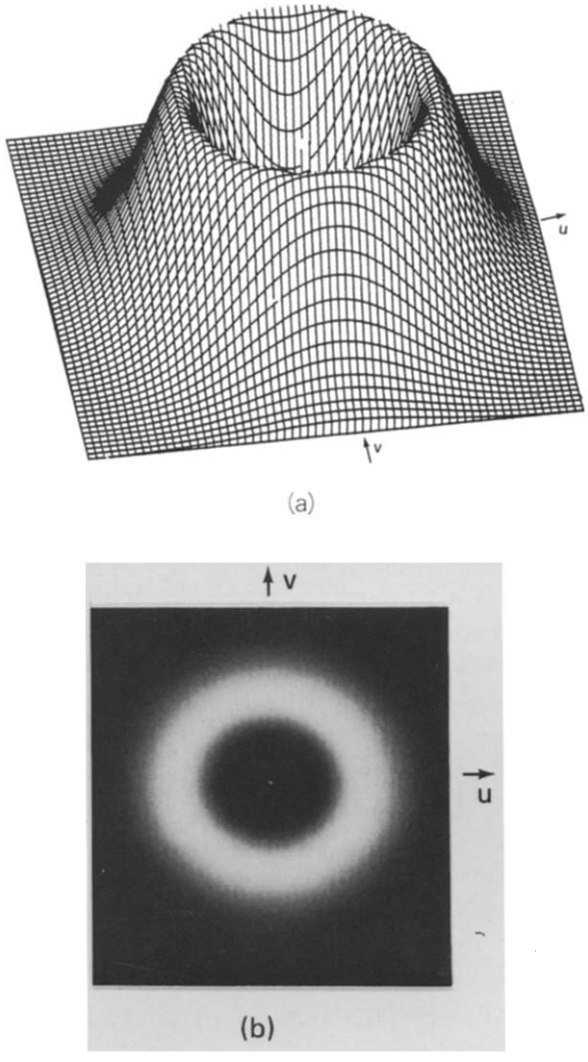

Figure 1. A spatial frequency filter. (a) A transect plot portraying the amplitude of the real part of the filter transfer function in the spatial frequency domain. The zero-frequency component is in the center of the region; (b) an image representation of the same filter transfer function.

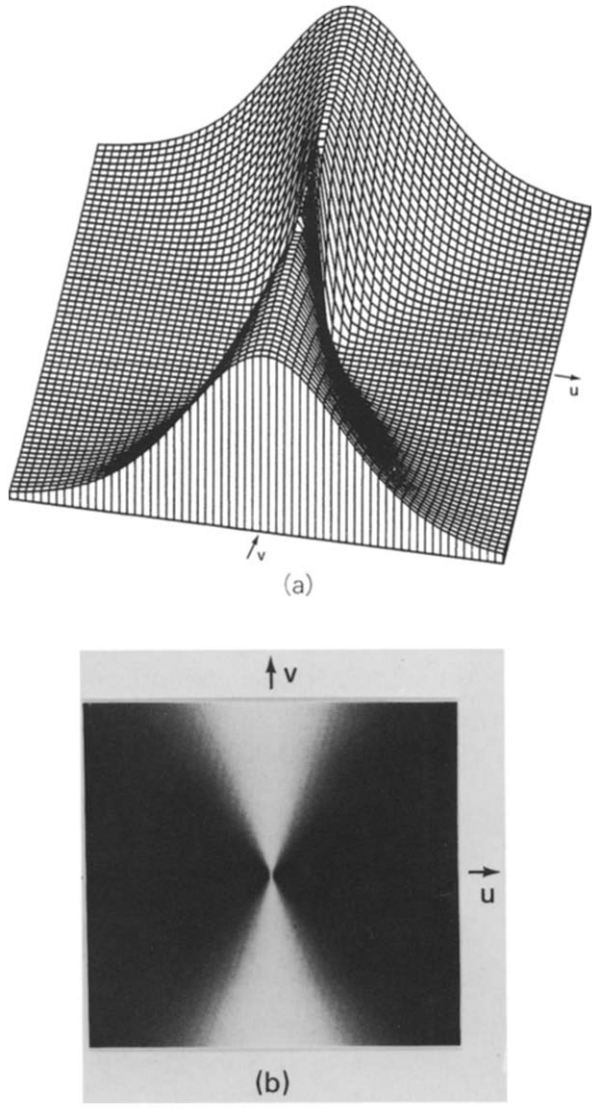

Figure 2. An orientation channel filter. (a) A transect plot portraying the amplitude of the real part of the filter transfer function in the spatial frequency domain. The zero-frequency component is in the center of the region; (b) an image representation of the same filter transfer function.

function (on a log scale); their center frequencies are one octave apart and their widths are between 1 and 2 octaves. The number of spatial frequency channels used depends on the size of the image. Figure 1 shows two representations of a spatial frequency filter in the spatial frequency domain (u-v plane). The height of the surface above the u-v plane (Figure 1a) and the gray level values (Figure 1b) represent the filter amplitudes $F(u, v)$. Four orientation channels are implemented with center orientations directed horizontally, vertically, and along the two diagonals. Figure 2 shows two representations of an orientation filter in the u-v plane. The appendix contains details of filters. For this study, the filtering is implemented by multiplication in the spatial frequency domain.

The output of the filtering procedure is a sequence of filtered images, each of which contains limited spectral information from the original image. This decomposition will now be exploited by computing simple texture features from the filtered images.

3. Definition of texture features

Certain properties of the channel-filtered images which are consequences of the filter definitions and of the spatial filtering operation provide guidance in constructing texture features. The gray level frequency histograms of the filtered images tend to be symmetrical about the mean gray level due to the symmetrical response of the filters to objects in the original image. Asymmetry in the gray level histograms is caused by the interaction of the responses of spatially close objects. The magnitude of the deviation of the gray level of a pixel in a filtered image from the mean gray level is generally related to the spectral energy originating from a neighborhood of that pixel in the given image. Since phase information is retained in the filtered images, the spatial distribution of the spectral energy passed by the filter is reflected in the gray level distribution in the filtered image. Thus, the spectral energy arising from small spatial areas can be measured from filtered images without recomputing a Fourier transform for each area of interest. This property makes it possible to speak of 'local spectral energy'. The differences among the channel-filtered images lie in their sensitivities to gray level variations in the original image over regions of different sizes and orientations.

This interpretation motivates the definition of a texture feature which measures the spread of the gray level frequency histogram of the filtered images. For an $N \times N$ image in which $f_k(g)$ denotes the number of occurrences of gray level g in the kth filtered image, the texture feature value for the kth filtered image is defined by

$$F_1 = \frac{1}{N^2} \sum_{g=0}^{G-1} f_k |g - \bar{G}|,$$

where $\bar{G}$ is average gray level. Note that the zero-frequency component of each filter is set to 1, so the average gray level of the filtered images is the same as the average gray level of the given image. This allows meaningful displays and comparisons of the channel filtered images and results in features which are invariant over global, constant gray level changes. An image will be represented by a feature vector consisting of the values of the average local energy feature computed on each filtered image.

Another texture analysis method based on a human visual model is described by Faugeras [6]. The features are similar to feature F_1, but many more filters are used, resulting in a feature space of higher dimensionality. A texture feature similar to feature F_1 was investigated by Laws [13] for texture classification and segmentation. The filtering operation was performed by spatial domain convolution using filters designed to detect 'spots', 'edges', and 'rings'. The algorithm was motivated by heuristic considerations and was not associated with a human visual model.

4. Texture classification

Eight natural images from Brodatz's texture album [1] will be used here. The images (and their abbreviations) are as follows: ceiling tile (TILE), beach pebbels (ROCK), beach sand (SAND), handmade paper (PAPE), pressed cork (CORK), grass lawn (GRAS), wood grain (WOOD), and straw screening (SCRE). No preprocessing was applied to the

digitized images. Twenty-five 64×64 samples were extracted from a 256×256 image of each class.

Each of the 200 subimages will be represented by a feature vector composed of the values of feature F_1 computed over eleven filtered images (four orientation channels and seven spatial frequency channels are used). The decision rule is to classify an unlabelled point into the class of its nearest labelled neighbor, where the distances are computed by the Euclidean distance metric. The error rate will be estimated using the Leave-One-Out rule.

The performance of feature F_1 will be compared to the performance of another feature, the total power spectral energy (PSE) in the sequence of filtered spectra [18]. The PSE feature uses no phase information, by definition, but it will use the same power spectral information as the channel filtering features.

The classification results for feature F_1 are given in Table 1. The PSE feature provides 69% correct classification while feature F_1 provides 98% accuracy. Values of the channel filtering features can be interpreted as visible image properties. For example, ROCK is the only class with high feature values in low frequency channels due to the presence of large objects in the ROCK image. SCRE has a unique pattern in orientation channels; the horizontal channel has a high value and the other orientation channels have low values. This reflects the horizontal directional tendency of SCRE. In addition, SCRE has low feature values in spatial frequency channels except for the channel centered at 32 cycles per image which captures the periodicity of the bars in the screen.

This classification experiment was repeated with 32×32 and with 16×16 subimages in [4]. Using 32×32 subimages degraded the performance only slightly; 91% accuracy was obtained. Using 16×16 subimages, feature F_1 was unable to discriminate the TILE and SAND texture classes, but there was only moderate degradation of performance on other texture pairs. In another series of experiments, it was demonstrated that feature F_1 is insensitive to global variations in average gray level. Thus, the classification results cannot be attributed to average gray level differences even though the images were not preprocessed or normalized.

Two computational simplifications of the channel filtering procedure were tested. One possible simplification is to use fewer channels. Since texture is most clearly perceived in regions containing only small areas of constant gray level, the low frequency channels, which respond to gray level variations over large areas, might be eliminated without degrading performance. In addition, the highest frequency channel does not capture any spectral information which is not available in other channels. Thus, the two lowest spatial frequency channels and the highest spatial frequency channel were eliminated [4]. Using the remaining eight channels on the 64×64 subimages resulted in 98% correct classification, the same as with all the channels present. The second simplification involved using channels with an ideal band-pass characteristic [4]. The ideal band-pass filters do not seem to degrade texture analysis performance; feature F_1 provides 98% correct results.

5. Texture segmentation

In more realistic situation, the images being analyzed contain an unknown number of textured

Table 1
Classification of eight natural image classes using 64×64 subimages with feature F_1

	TILE	ROCK	SAND	PAPE	CORK	GRAS	WOOD	SCRE	
TILE	22	2	1	0	0	0	0	0	
ROCK	1	24	0	0	0	0	0	0	
SAND	0	0	25	0	0	0	0	0	
PAPE	0	0	0	25	0	0	0	0	98% accuracy
CORK	0	0	0	0	25	0	0	0	
GRAS	0	0	0	0	0	25	0	0	
WOOD	0	0	0	0	0	0	25	0	
SCRE	0	0	0	0	0	0	0	25	

regions which must be identified, thus segmenting the image. The segmentation procedure will require a determination of the texture in a neighborhood about each pixel. A feature vector will be computed for each pixel in the image. The number of texture classes present will be determined from the feature vectors, and all of the pixels in the image will be classified.

5.1. Computing texture features for segmentation

By applying two operations to the filtered images, the gray level at each pixel of the filtered images can be replaced by a value which is related to feature F_1. In the first step, we replace each gray level as follows:

$$I_k'(x, y) = 2|I_k(x, y) - \bar{G}|$$

where $I_k(\cdot, \cdot)$ is the kth filtered image and G is the mean gray level of the filtered images. Simply taking the absolute deviation would result in an image with about half the original number of gray levels. The absolute deviation is scaled to increase the precision of the averaging step which occurs next.

The second step involves computing a moving average over the $I_k'(\cdot, \cdot)$ images. For segmentation experiments, the averaging is performed by replacing each pixel in $I_k'(\cdot, \cdot)$ with the average gray level in a small neighborhood about the pixel. The result of the averaging is a set of 'feature images' in which the gray level at each pixel is a measure of the texture present at the corresponding location of the original image. In [4], both 8×8 and 16×16 square windows were used for computing the feature images. In this paper, only the results of the 16×16 window will be reported. Note that the computation of the feature images does not require each neighborhood to be filtered separately.

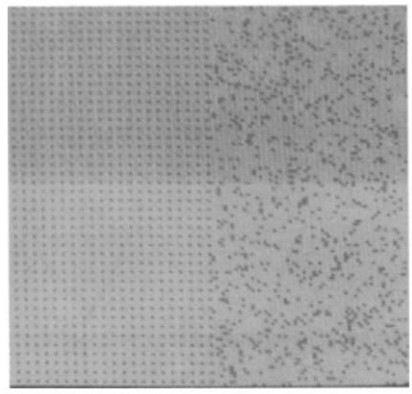

Figure 3. Dots image for segmentation experiment 1.

5.2. Segmentation using feature images

Each pixel in the original image is represented by a feature vector composed of the gray levels at corresponding locations in the sequence of filtered images. A partitional clustering algorithm called CLUSTER (5) provides partitionings of the data with the number of clusters going from 2 to user-specified bound (8 in this study). Each partitioning of the data corresponds to a segmentation of the given image. It will be necessary to evaluate these clusterings to determine which ones are appropriate representations of the data.

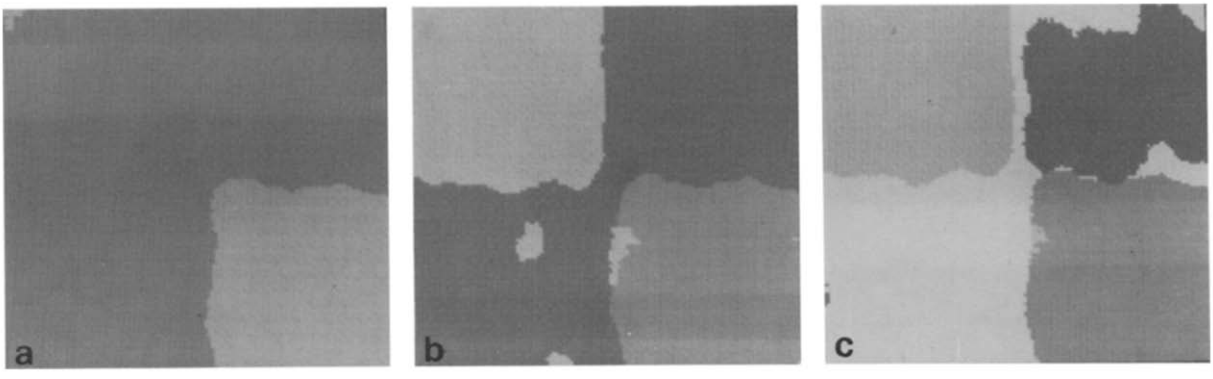

Figure 4. Segmented images for dots using 16×16 averaging windows. (a) 2-cluster solution; (b) 3-cluster solution; (c) 4-cluster solution.

To reduce computational requirements, only 64 pixels spaced 16 rows and 16 columns apart will be clustered. This simplification assumes that the textured regions to be segmented will not be irregularly shaped or very small in area. Additionally, only eight channels, the four orientation channels and spatial frequency channels 3–6 (center frequencies 4, 8, 16, and 32 cycles per image) will be used. These eight channels were found earlier to provide good classification results. The cluster centers will be used to define a minimum-distance classifier to classify the remaining pixels. The classification results will be displayed as a segmented image in which gray levels denote the cluster labels assigned to each pixel.

5.3. *Evaluating the segmentations*

A measure of the 'validity' of cluster k which takes into account its compactness and isolation is defined as

$$S(k) = \frac{\min\limits_{l \neq k} \left[\sum\limits_{j=1}^{D} [C(k,j) - C(l,j)]^2 \right]^{1/2}}{\left[\frac{1}{N(k)} \sum\limits_{i=1}^{N(k)} \sum\limits_{j=1}^{D} [X(i,j) - C(k,j)]^2 \right]^{1/2}}$$

where $N(k)$ is the number of points in cluster k, $C(k,\cdot)$ is the cluster center for cluster k, D is the dimensionality of the feature space, and $X(i,j)$ is the value of the jth feature for the ith pattern. Large values of $S(k)$ indicate compact, well-isolated clusters. An acceptable clustering is one where the value of $S(k)$ for all clusters exceeds a threshold. We have empirically determined a threshold of 1.70 for $S(k)$ by tuning the threshold in preliminary experiments to yield approximately the number of perceived clusters. The threshold value requires that for each cluster the minimum distance to another cluster center be at least 1.70 times the average within-cluster distance.

5.4. *Segmentation experiment 1: Regular and random dots*

The data for this experiment consists of a 128×128 binary image (Figure 3). The left half of the image is a regular dot pattern in which the dots are separated by 3 pixels from their horizontal and vertical neighbors. The right half of the image is a random dot pattern in which the probability that each pixel is black is independent of the other pixels and is approximately equal to 1/9.

The segmented images are shown in Figure 4. The figure shows only the segmentations for 2, 3 and 4 clusters. The two-cluster segmentation accurately distinguishes the regions of different texture; over 98% of the pixels are correctly labelled. The segmentation results corresponding to more than two clusters break up the random texture into irregularly shaped regions. The regular texture is not subdivided. None of the segmented regions lie across the boundary between the textures. Table 2 shows the number of points in each cluster, the values of $S(k)$ for each cluster, and the weighted average values of $s(k)$. Only the two-cluster solution is accepted. The table also shows a very high value for cluster 1, which corresponds to the regular texture.

5.5. *Segmentation experiment 2: Gaussian white noise*

We will apply the segmentation procedure to a 128×128 image which is visually perceived to contain a single homogeneous texture. The gray level

Table 2
Evaluation of clustering on dots image using 16×16 averaging windows, * indicates accepted clustering solution

k	$N(k)$	$S(k)$
2 clusters		
1	32	9.39
2	32	3.79
AVG		6.59*
3 clusters		
1	32	8.86
2	22	1.33
3	10	1.62
AVG		5.14
4 clusters		
1	32	8.27
2	11	1.22
3	9	1.39
4	12	1.28
AVG		4.78

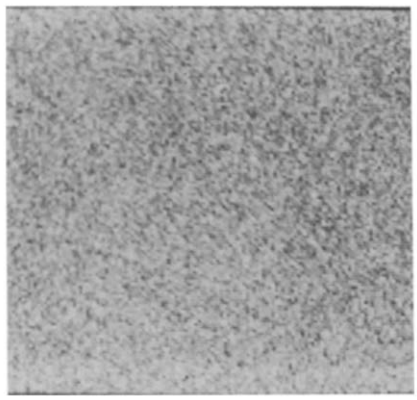

Figure 5. Gaussian white noise image for segmentation experiment 2.

at each pixel is generated independently using a Gaussian distribution with mean 128 and standard deviation of 30. The image is illustrated in Figure 5. The same filtering and segmentation procedure as in the previous experiment was applied. The Gaussian white noise image is segmented into irregular patches as shown in Figure 6.

None of the clusterings pass the threshold test. In fact, none of the individual clusters are valid. Since all multiple-class solutions are rejected, we conclude that the image contains a single texture.

5.6. *Segmentation experiment 3: Composite of natural images*

Figure 7 shows the 128×128 image composed of four natural image classes: WOOD, PAPE, SCRE, and SAND. The results of segmentation using 16×16 windows are shown in Figure 8.

The solutions for 3, 4 and 5 clusters are found to be acceptable based on $S(k)$ values. Using the weighted average of $S(k)$ to rank the solutions, we find the three-cluster solution to be the best, followed in order by the 4- and 5-cluster solutions. In the three-cluster solution, the SAND and PAPE regions are merged in a single cluster. The irregularity and the contrast of these areas are similar, so the preference for this solution is plausible. The four-cluster solution, which is the second choice, reasonably captures the four image types used; 91% of the pixels are correctly labelled.

6. Conclusions

A theory of human visual processing which postulates the existence of spatial frequency and orientation channels was implemented using a filtering procedure. A texture feature was developed which measures the average local energy in an image region. The texture feature was successfully applied in classification problems. An algorithm was presented for computing the texture of a neighborhood about each pixel. The resulting textural transform was evaluated by applying a texture segmentation procedure in three experiments. The segmented images indicate that the feature space is a reasonable representation of the perceived image texture.

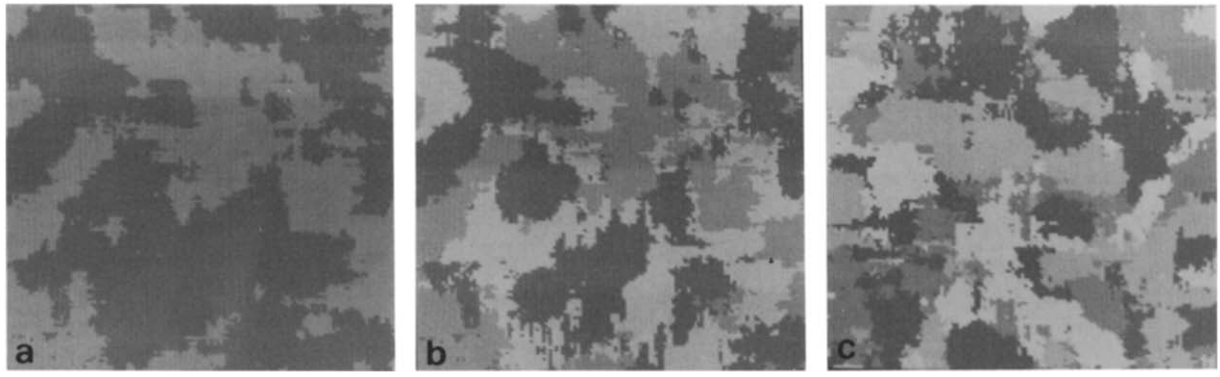

Figure 6. Segmented images for Gaussian white noise using 16×16 averaging windows. (a) 2-cluster solution; (b) 3-cluster solution; (c) 4-cluster solution.

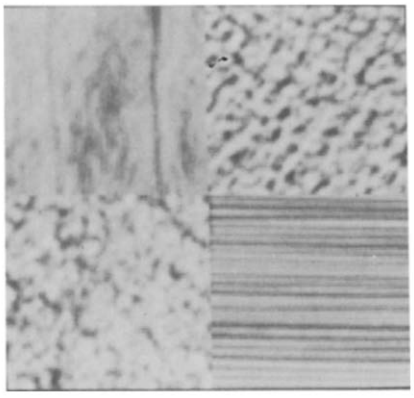

Figure 7. Natural image composite for segmentation experiment 3.

Appendix. Filter definitions

The transfer function of a $2N \times 2N$ spatial frequency filter $F_k(u, v)$, $-N+1 \le u, v \le N$, is defined as follows:

$$\text{Imag}[F_k(u, v)] = 0 \quad \text{for all } u, v, k,$$

$$\text{Real}[F_k(0, 0)] = 1 \quad \text{for all } k,$$

and for $(u, v) \neq (0, 0)$,

$$\text{Real}[F_k(u, v)] = \exp\left(-0.5 * \frac{[\ln(D(u, v)) - \ln(\mu_k)]^2}{\sigma^2} \right),$$

where $D(u, v) = (u^2 + v^2)^{1/2}$, $\sigma = 0.275$, and $\mu_k = 2^{k-1}$.

For a $2N \times 2N$ image, we will use spatial frequency filters for $k = 1, \ldots, (\log_2 N + 1)$. This definition yields a series of filters one octave apart whose widths are slightly more than one octave on each side of the center frequency. The value of σ was selected to produce filters whose widths are within the constraints for human visual filters given in [7].

Table 3
Values of μ_k

Value of k	Value of μ_k (*degrees*)
1	0 (*horizontal*)
2	45
3	90 (*vertical*)
4	135

The transfer function of a $2N \times 2N$ orientation filter $G_k(u, v)$, $-N+1 \le u, v \le N$, is defined as follows:

$$\text{Imag}[G_k(u, v)] = 0 \quad \text{for all } u, v, k,$$

$$\text{Real}[G_k(0, 0)] = \tfrac{1}{2} \quad \text{for all } k,$$

and for $(u, v) \neq (0, 0)$,

$$\text{Real}(G_k(u, v)) = \exp\left(-0.5 * \frac{A_k^2}{\sigma^2} \right),$$

where

$$A_k = \min[|\mu_k - \arctan(v/u)|, |(\mu_k - 180) - \arctan(v/u)|],$$

$\sigma = 17.8533$, $1 \le k \le 4$, and the values of μ_k are given in Table 3.

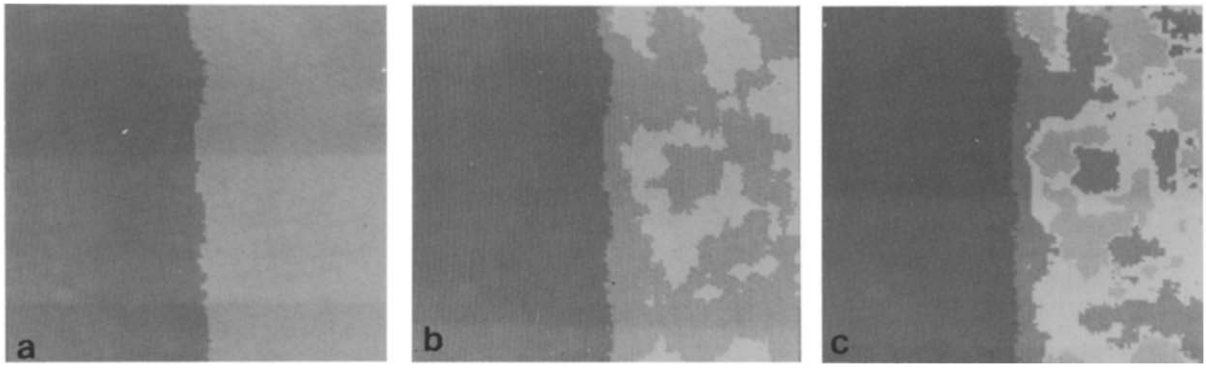

Figure 8. Segmented images for natural image composite using 16×16 averaging windows. (a) 2-cluster solution; (b) 3-cluster solution; (c) 4-cluster solution.

The value of σ is chosen to produce the same amount of overlap between channels as in the spatial frequency channels. The orientation channels here are somewhat wider than the 30 degree wide channels proposed for human vision in [7].

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

[2] Campbell, F.W. and L. Maffei (1970). Electrophysiological evidence for the existence of orientation and size detectors in the human visual system. *J. of Physiology* 207, 635–652.

[3] Campbell, F.W. and J.G. Robson (1968). Application of Fourier analysis to the visibility of gratings. *J. of Physiology* 197, 551–566.

[4] Coggins, J.M. (1982). A framework for texture analysis based on spatial filtering. Ph.D. dissertation, Department of Computer Science, Michigan State University.

[5] Dubes, R.C. and A.K. Jain (1976). Clustering techniques: The user's dilemma. *Pattern Recognition* 8, 247–260.

[6] Faugeras, O.D. (1978). Texture analysis and classification using a human visual model. *Proc. Fourth Intern. Joint Conf. Patt. Recognition*, 549–552.

[7] Ginsburg, A.P. (1977). Visual information processing based on Spatial filters constrained by biological data. Ph.D. dissertation, University of Cambridge, England. Also published as Air Force Aerospace Medical Research Laboratory Technical Report AMRL-TR-78-129, December, 1978.

[8] Ginsburg, A.P. and J.M. Coggins (1981). Texture analysis based on filtering properties of the human visual system. *Proc. Intern. Conf. Cybernet. and Society*, Atlanta, 112–117.

[9] Granlund, G.H. (1981). GOP: A fast and flexible processor for image analysis. In: M.J.B. Duff and S. Levialdi, Eds., *Languages and Architectures for Image Processing*, Academic Press, New York.

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

[11] Ikonomopoulos, A. and M. Unser (1984). A directional filtering approach to texture discrimination. *Proc. Seventh Int. Conf. Patt. Recognition*, Montreal, 87–89.

[12] Knutsson, H. et al. (1981). Anisotropic filtering by image content. *Proc. Second Scandinavian Conf. Image Anal.*, Helsinki, 146.

[13] Laws, K.I. (1980). Textured image segmentation. University of Southern California Image Processing Institute, TR-940.

[14] Unser, M. (1984). Local linear transforms for texture analysis. *Proc. Seventh Int. Conf. Patt. Recognition*, Montreal, 1206–1208.

[15] Verbeek, P.W. (1982). Space invariant image operations constructed from local direction detection and shift invariant filters. Delft University of Technology, Applied Physics Patroonherkennen TR-820324.

[16] Verbeek, P.W. and D.J de Jong (1984). Edge preserving texture analysis. *Proc. Seventh Int. Conf. Patt. Recognition*, Montreal, 1030–1032.

[17] Wermser, D. and C.E. Leidtke (1982). Texture analysis using a model of the visual System. *Proc. Sixth Int. Conf. Patt. Recognition*, Munich, 1078–1080.

[18] Weszka, J., C. Dyer and A. Rosenfeld (1976). A comparative study of texture measures for terrain classification. *IEEE Trans. Systems Man Cybernet.* 6, 269–285.