Unsupervised Segmentation of Texture Images†

Xavier Michel, Riccardo Leonardi and Allen Gersho
Communications Research Laboratory
Department of Electrical & Computer Engineering
University of California, Santa Barbara, CA 93106

ABSTRACT

Past work on unsupervised segmentation of a texture image has been based on several restrictive assumptions to reduce the difficulty of this challenging segmentation task. Typically, a fixed number of different texture regions is assumed and each region is assumed to be generated by a simple model. Also, different first order statistics are used to facilitate discrimination between different textures. This paper introduces an approach to unsupervised segmentation that offers promise for handling unrestricted natural scenes containing textural regions. A simple but effective feature set and a novel measure of dissimilarity are used to accurately generate boundaries between an unknown number of regions without using first order statistics or texture models.

A two stage approach is used to partition a texture image. In the first stage, a set of sliding windows scans the image to generate a sequence of feature vectors. The windowed regions providing the highest inhomogeneity in their textural characteristics determine a crude first-stage boundary, separating textured areas that are unambiguously homogeneous from one another. These regions are used to estimate a set of prototype feature vectors. In the second stage, supervised segmentation is performed to obtain an accurate boundary between different textured regions by means of a constrained hierarchical clustering technique. Each inhomogeneous window obtained in the first stage is split into four identical subwindows for which the feature vectors are estimated. Each of the subwindows is assigned to a homogeneous region to which it is connected. This region is chosen according to the closest prototype vector in the feature space. Any two adjacent subwindows that are assigned to different regions will in turn be considered as inhomogeneous windows and each is then split into four subwindows. The classification scheme is repeated in this hierarchical manner until the desired boundary resolution is achieved. The technique has been tested on several multi-texture images yielding accurate segmentation results comparable or superior to the performance obtained by human visual segmentation.

1. INTRODUCTION

Human beings have the natural skill of dividing a complex scene into the different objects that it contains. This ability is based not only on our sophisticated visual system but also on our knowledge of the appearance of familiar objects in natural images and their likely shapes, locations, and spatial relationships. Nevertheless, each texture region in an image tends to have a statistically distinct character so that it is reasonable to expect that a substantial part of our image segmentation ability is contributed by processing that performs statistical discrimination between texture regions. This is confirmed by our effectiveness in segmenting synthetic texture images that contain no semantic clues. Such a segmentation of course belongs to the category of ***unsupervised*** pattern recognition since their is no prior information regarding the number of classes or the characteristics of the individual texture types that may be present.

Thus, it is an exciting challenge in the field of computer vision to develop algorithms that perform unsupervised segmentation of images on the basis of their textural content alone with a performance that matches the human visual system. Since texture is an important primitive that must be considered in in the analysis of images, such algorithms can provide useful tools for more complex vision tasks. There are also important applications in industrial and medical imaging where segmentation uniquely based on textural information is essential.

This paper introduces an approach to texture segmentation that first identifies homogeneous subregions of each identifiable texture region thereby obtaining a set of prototypes as well as a first-order approximation to the segmentation task. Then, in a second stage, a supervised segmentation procedure is used to extend these subregions and identify accurate boundaries with the help of a hierarchical clustering technique, ridge riding, and a carefully designed dissimilarity measure. The method is applicable to a variety of transform based methods for generating local feature vectors. Section 2 reviews the concept and meaning of texture and examines the characterization of texture features using transforms. Section 3 deals with the problem of extracting the inhomogeneous areas, while Section 4 discusses the boundary refinement process. Finally, Section 5 presents

† This work was supported by the Swiss National Foundation for Scientific Research, the University of California MICRO Program, and Bell Communications Research, Inc.

the experimental results obtained with our segmentation method.

2. MEASURING TEXTURE PROPERTIES

As often pointed out in the literature [1-3], the "notion of texture admits to no rigid description" (see [1], p. 166). Haralick [2] suggests that texture may be regarded as "an organized area phenomena". This organization may be chaotic (clouds) or perfect (chessboard), but in any case it implies a spatial arrangement of certain patterns. The problem of texture characterization lies in the fact that the patterns are not clearly or uniquely extractable, and that they may slowly vary in space. Given a specified texture region, certain descriptive patterns or statistical features may be extracted, but these may not correspond to those used by the human visual system. This is especially true for natural images, considering the complexity and diversity of real textures. In some sense, however, it is reasonable to regard texture as a "neighborhood property" of an image point [4]. For a homogeneous texture region, we expect to have some degree of statistical regularity of this neighborhood property as we scan through all points in the region. In the context of texture segmentation however, it is not necessary to completely describe the texture characteristics of each region; it is sufficient to be able to discriminate between two or more different textures. Therefore, only a partial but reasonably well-chosen description of the neighborhood property of each pixel may be sufficient.

In this study, we focus specifically on texture images. Such images are defined as mosaics of natural image textures, such as grass, paper, sand, cork, reptile skin, etc. If a texture image is synthetically generated, it is possible to know the exact boundary location for each region and therefore to assess the performance of the suggested algorithm. It is also possible to avoid the ambiguity in real situations where a region cannot be discriminated from its neighbors as no neighborhood property of any pixel in that region can be found to describe that specific texture and where the human observer is also unable to identify a boundary. With synthetic texture images it is also possible to verify if a segmentation algorithm performs ***better*** than a human observer since we have an absolute knowledge of the true boundaries. The individual textures used in this work have been obtained by digitizing images from the Brodatz ***Photographic Album for Artists & Designers*** [5] over a 512x512 sampling grid with an 8 bit quantization of the luminance signal.

The goal of a segmentation algorithm is to split an image into disjoint regions such that within each region there is a homogeneity of properties, but between regions there are sharp discontinuities of these properties. There is a fundamental ***uncertainty*** rising in the texture segmentation problem. Identification of the correct region membership of a point requires observation of an adequate size neighborhood of that point; however, as the point approaches a boundary between two region, the neighborhood may overlap the adjoining region. Reducing the neighborhood size tends to allows sharper boundary discrimination but weakens the within-texture feature measures. The prodigious ability of a human observer when performing this task suggests that the uncertainty is overcome by use of sequential processing with different levels of resolution. The simplest such model of this type of processing is a two stage segmentation. In the first stage, a course resolution is used to identify inhomogeneous areas located along one-dimensional paths that separate the most predominant uniform regions of the texture image. A finer resolution analysis is then performed within these inhomogeneous areas in order to refine the boundaries and obtain sharp transitions of the textural properties across these boundaries. This analysis requires a matching between regions that have been previously defined as homogeneous and the adjacent portions of the inhomogeneous areas. In other words, once homogeneous regions have been detected, they are extended outwards as far as possible, on the basis of local texture properties, to delineate an accurate boundary.

The problem of describing local texture properties is equivalent to find an adequate set of parameter values that describe of a local neighborhood. It is reasonable to consider an individual texture region as a "cookie-cutter" subset of an infinite two-dimensional stationary ergodic random process. To describe this process, it is convenient to define for each pixel (k,l) an $M \times N$ neighborhood vector, $\mathbf{x}_{k,l}$ of dimension MN whose components are the luminance values of the $M \times N$ pixels centered on (k,l) when M and N are odd. In the event M or N is even, the window is centered at $(k+1/2,l)$ or $(k,l+1/2)$, respectively. The original texture may then be viewed as a two-dimensional multivariate sequence $\{\mathbf{x}_{k,l}\}$.

Given such a texture region, statistical characteristics of the neighborhood vector can then be performed on that image by spatial averaging of some function of the neighborhood vector. A texture model can in principle be defined by the probability distribution of the neighborhood vector over an ensemble of possible realizations of this texture. In any recognition process, we assume that humans compare this "ideal" representation to some kind of spatial averaging of the local data to be analyzed.

Julesz [6] and more recently Gagalowicz [7] have conjectured that most textures are perceptually indistinguishable if they have the same first and second order statistics and many experiments with synthetically generated textures seem to corroborate this claim. This hypothesis justifies the use of spatial grey-level correlations or co-occurence matrices [8] to estimate second-order spatial statistics. However, large neighborhoods often make this approach very cumbersome since a very large number of features are needed to describe the texture content, making the segmentation task much more complex. In fact, many of these neighborhood parameters are highly correlated and therefore contain considerable redundancy. Furthermore, it is more difficult, in the absence of *a priori* knowledge, to define a satisfactory clustering criterion in a high dimensional feature space, due to the significant interdependence among the features. It is therefore advisable to work with a low-order feature space.

The Julesz hypothesis implies that an adequate feature set can be based on the second order moments of linear combinations of the components of the neighborhood vector. This is consistent with the work of different authors [4,9,10] who have suggested that local neighborhood information can be extracted by linear filtering operators, using energy measures computed at the output of a filter bank. More recently, Cano et al. [11] suggested the use of hierarchical linear transforms to derive sets of features that are compact and visually complete, for a large class of textures. Compactness means that a modest number of parameters is used to describe the texture characteristics. Completeness means that any two textures that have the same set of parameter values are visually indistinguishable. The measurements can be viewed as estimates of certain local features of the texture. First-order statistics of these features yield second-order statistics of $\{\mathbf{x}_{k,l}\}$. An optimal set of filters may be derived via the Karhunen-Loeve expansion. The transformed space then yields a compact and uncorrelated set of features from the neighborhood vector. It should be noted that any transform on an $M \times N$ neighborhood vector produces MN transformed values, each of which is obtained by an $M \times N$ mask operating on the $M \times N$ neighborhood of a given point. The effect of each such mask as the neighborhood is moved across an image is a filtering operation that generates a sequence of feature values, one for each point. The output of each filter is sometimes called a *channel*.

The segmentation problem does not require the optimal character of the the Karhunen-Loeve transform to adequately describe each texture. In fact, a transform should ideally be designed to achieve optimal discrimination between different image regions. To improve clustering performances, it is more appropriate to use the generalized Fisher's linear discriminant functions (GFLD), which are optimal for a large variety of classification problems [12]. A multi-resolution Karhunen-Loeve transform has been recently suggested by Unser et al. [13] to approximate the GFLD, and gives satisfactory results in texture segmentation problems. Its computation is however very demanding because it requires the simultaneous diagonalization of the covariance matrices evaluated at two different spatial resolutions. The most delicate issue in that approach is the estimation of the covariance matrices, when no supervision of the segmentation process is possible.

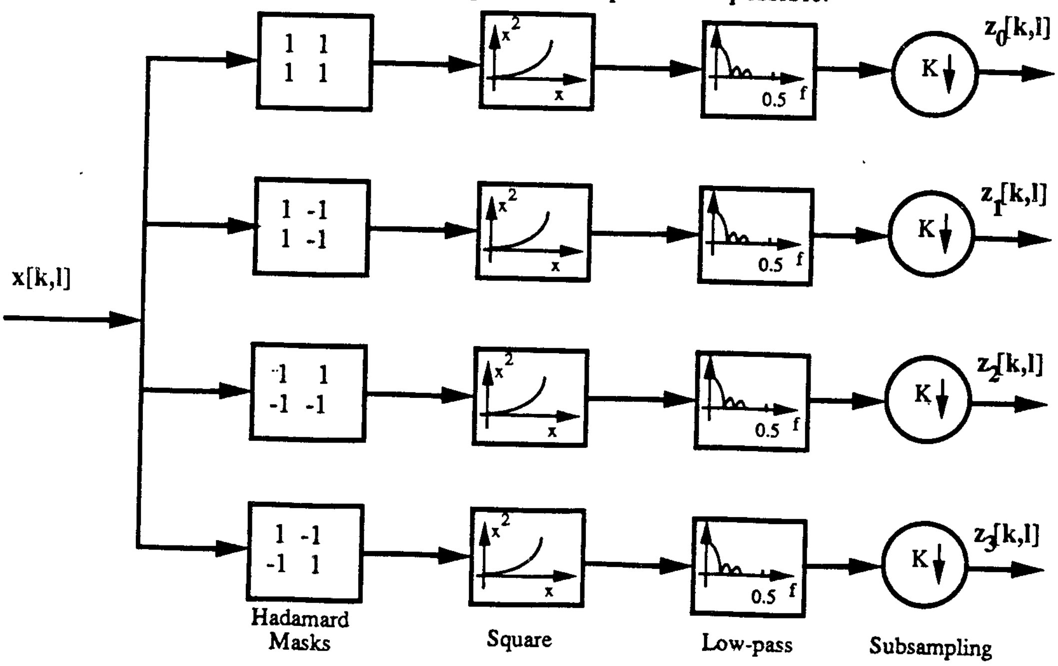


Figure 1: Feature extraction process (Subsampling and averaging are shown in one dimension for simplicity).

In this paper, we describe a segmentation technique that is applicable to any choice of transform, but for simplicity we focus on a single computationally simple linear transform, the Hadamard transform of a 2×2 neighborhood vector, to describe or discriminate each texture. Figure 1 outlines the feature extraction process.

Each channel of the filter bank picks up specific frequency information. In the first channel, the signal is low-pass filtered whereas in the other channels a simple set of horizontal, vertical and diagonal edge detectors are obtained. The texture characterization is represented by the different channel histograms. If the stationarity assumption is valid, first-order moments of each transformed coefficient will be identically equal to zero for the high frequency channels. More useful features (related to the relative variance of different directional components) may be estimated from the second-order moments of the transformed coefficients. A complete description of the channel histograms is difficult to obtain since estimation of third or higher order moments is very sensitive to round-off errors and the variance of the estimation remains high for small size windows [11].

In order to guarantee that the segmentation is achieved using high order texture properties, each texture in the texture images used in this work has been preprocessed to have uniform first-order statistics. This will assure that the algorithm will remain efficient in handling real scenes containing regions that can only be discriminated on the basis of structural information. Since the variance and mean are kept constant throughout the image, the second-order moments of the first transform coefficient may be disregarded. The texture features that will be used to describe the texture characteristics in the segmentation process correspond to the three energy components of the different high-frequency channels.

Figure 2 shows a texture image made of five different textures (specifically, D9, D19, D24, D29, and D38 in [5]). Figure 3 shows the corresponding image with preprocessed textures (with mean and standard deviation of the pixel values set to $m_d = 127$ and $\sigma_d = 35$ respectively).

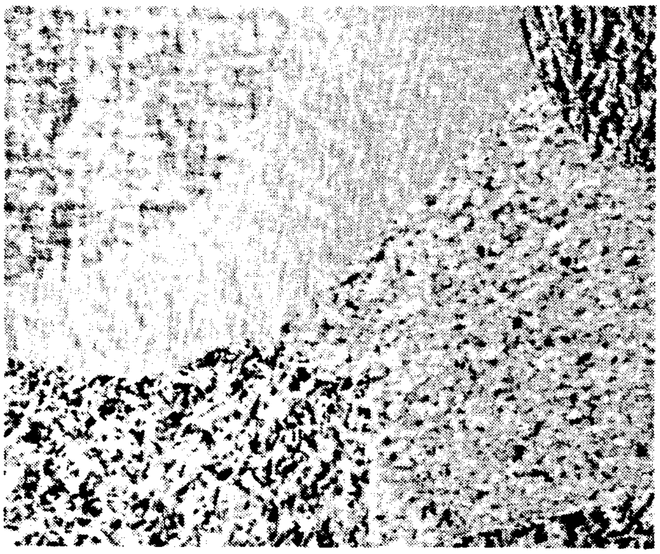


Figure 2: Texture image (D9, D19, D24, D29 and D38 in [5]).

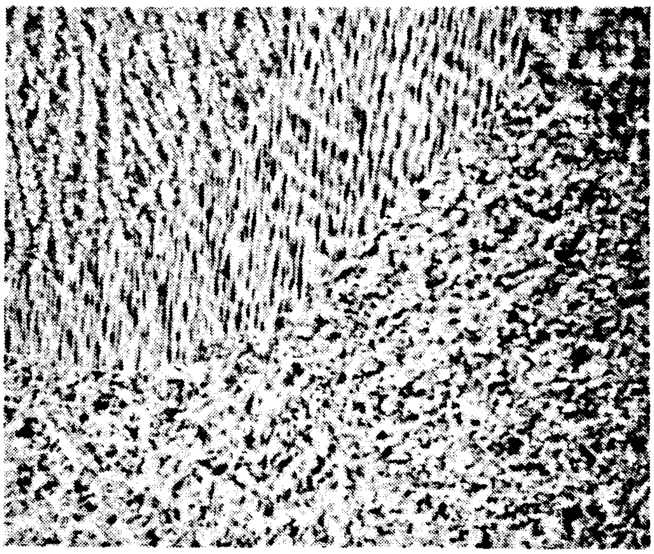


Figure 3: Preprocessed texture image (D9, D19, D24, D29 and D38 spatial mean and standard deiation has been set to 127 and 35, respectively).

A sequence of feature images may be extracted taking one of the feature values (a transform coefficient) at every pixel location for each feature used in the texture description model, each image corresponding to one feature defined as $z_i[k,l]$ where in our case, i ranges from 1 to 3.

For practical reasons, it is desirable to subsample the feature images to avoid unnecessary computations. In fact, a coarse estimate of the boundary location is sufficient (at least in the first stage of the segmentation algorithm). However, it is essential to protect the subsampling process from any aliasing effects. Estimation of the different features is done by averaging the square of the outputs of the 2x2 Hadamard filter bank. A low-pass averaging filter of size KxL can be used to perform this estimation. The values of K and L depend on the size of the texture patterns. Assuming the averaging operation is performed over a uniform area of the texture image, K and L may be chosen so as to be the smallest integers that would guarantee a near-constant measure of the different feature components; however, it is important to keep them large enough to reduce the variance of the feature estimates. It is reasonable to assume that nearly-constant values for the features would be computed for each texture with a 64x64 window (a reasonable amount of texture "patterns" should be present in such a window).

We next consider how much the different averaging windows of size $K \times L$ (i.e. 64x64) should overlap in order to avoid aliasing effects. The averaging operation has a frequency response given by

$$H(f,g) = \frac{1}{K\,L}\,\frac{\sin(K\pi f)\,\sin(L\pi g)}{\sin(\pi f)\,\sin(\pi g)} \tag{1}$$

The Hadamard filter bank reduces the bandwidth of the original signal within each of its channels. We disregard this effect compared to the effect of the low-pass behavior introduced by the averaging process. Assuming that the square of the output of each channel is a full band signal (which is not the case), only 5% of its energy would produce aliasing effects if the different averaging windows would overlap by 75%, i.e. $K \times L$.

Unless otherwise specified, we shall assume that texture information would be represented by the second-order moments of the high-frequency 2x2 Hadamard masks. Estimation of these second-order moments is performed over 64x64 windows overlapping by 16x16. Figure 4 shows the three different feature images of size 32x32 relative to the texture image of Figure 3.

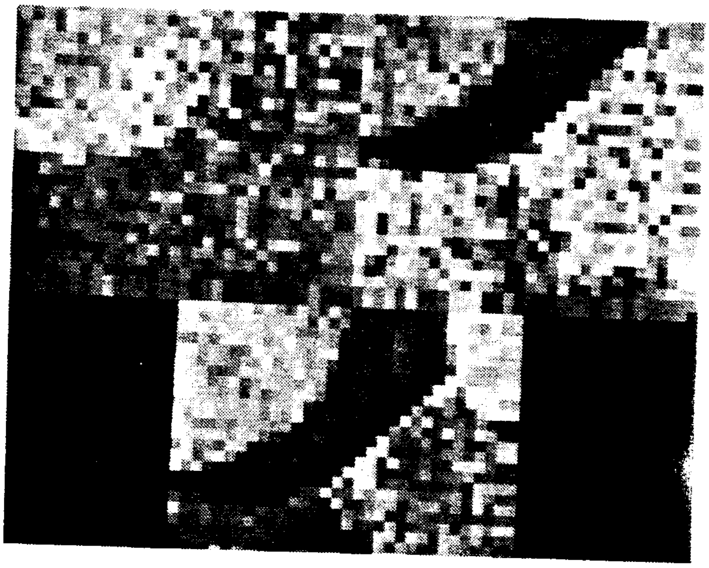


Figure 4: Feature images (clockwise from top left: $z_1[k,l]$, $z_2[k,l]$ and $z_3[k,l]$).

3. INHOMOGENEOUS AREA EXTRACTION AND PROTOTYPE TEXTURE

The segmentation of the texture image implies a partitioning of the corresponding image into uniform textured regions. Since this task is unsupervised, no prior information is available on the image. We know only that the image is composed of textures.

Boundaries between different textures define one-dimensional paths through the image. The areas along these paths can be viewed as inhomogeneous texture areas. Since texture information is described by spatial averaging over a certain neighborhood, inhomogeneity may be detected by comparing how the different texture features vary spatially. By integrating the information available at every (k,l) location over the different feature images and comparing it to the relative role of its neighbors by an adequate inhomogeneity measure (see 3.1), an inhomogeneity image may be extracted. Areas that are nonuniform in their texture characteristics would appear as edge information in the inhomogeneity image. Local irregularities within natural textures may also be regarded as inhomogeneous areas of the texture image. A simple threshold, even if adaptively adjusted, may not be sufficient to discriminate between these local irregularities and the inhomogeneous areas along texture boundaries. A specific ridge riding strategy, described later, is needed to track the edges that these inhomogeneous areas define in the inhomogeneity image.

Once the inhomogeneous paths is extracted, all connected sets of pixels that do not correspond to them may be merged together. They should correspond to uniform texture areas. These uniform portions of the image will provide training areas or cores for extracting prior information about the texture model that might be used to refine the boundaries between textures. Prototype feature vectors may be estimated as well as specific statistical parameters that are necessary in any subsequent clustering process.

3.1. Inhomogeneity Measure

Two distinct aspects need to be considered for the design of an inhomogeneity measure: robustness to the inherent variations of each feature; one-dimensional characteristic of region boundaries. The one-dimensional characteristic of the region boundaries (see edge detection) suggests to design an inhomogeneity measure based on some gradient information of the feature images $z_i[k,l]$. The corresponding inhomogeneity image could be estimated by:

$$E[k,l] = d^2(\mathbf{z}[k,l],\mathbf{z}[k+1,l]) + d^2(\mathbf{z}[k,l],\mathbf{z}[k,l+1]) \tag{2}$$

where $\mathbf{z}[k,l]$ represents the vector $[z_1[k,l] \; z_2[k,l] \; \cdots \; z_{N_f}[k,l]]^T$, and d is an appropriate distance measure in the feature space of dimension N_f, that discriminate between two feature vectors. Another way of designing an appropriate inhomogeneity measure is to base it on the region uniformity condition suggested by Chen and Pavlidis [14]. Given a uniformity predicate [15], an area is said to be uniform if all the connected subsets of the region satisfy the predicate, i.e. each subregion's property equals the region's property. A simple case may be obtained by considering a square and its four subsquares. The corresponding inhomogeneity measure is given by:

$$E[k,l] = \sum_{i=1}^{4} d(\mathbf{z}[k,l],\mathbf{z}_{S_i}[k,l]) \tag{3}$$

where $\mathbf{z}_{S_i}[k,l]$ defines the feature vector for each of the four subsquares that form the square of size $K \times L$ (i.e., 64x64) located at (k,l).

Even if this criterion is not error-free (see Figure 5), the probability of occurence of such errors is relatively small. Furthermore, such a criterion may be more robust to local irregularities of a given texture than a gradient estimate.

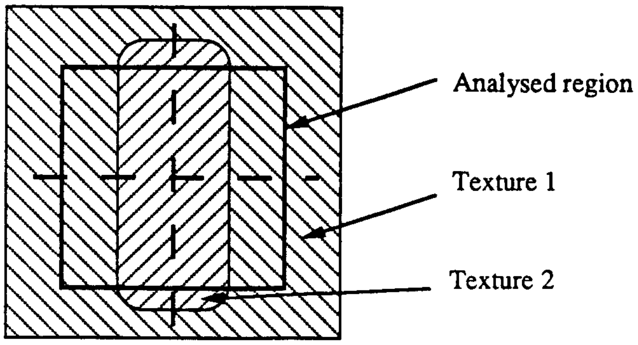


Figure 5: Error induced by inhomogeneity measure (3). Even though 2 textures are present in the window, the feature values will not differ significantly between the big square and the four subsquares.

The inherent variations of each feature may be taken into account by the use of a specific distance measure d in the feature space. Unfortunately statistical estimates of the different processes related to each texture cannot be obtained at this level of the segmentation. It is therefore difficult to weight each feature component so as to minimize the intraset distance of each texture. Rather, the differences in order of magnitude between the various feature components may be reduced by the choice of a logarithmic distance measure. Given two feature vectors $\mathbf{z}$ and $\mathbf{z}'$, their distance in the feature space may be expressed as:

$$d(\mathbf{z},\mathbf{z}') = \sum_{i=1}^{N_f} \log\left(\frac{z_i}{z'_i}\right) \tag{4}$$

Figure 6 shows the 32x32 inhomogeneity images associated to texture image of Figure 3. Figure 6a shows the result obtained for a Euclidean distance, whereas Figure 6b shows it for a logarithmic distance measure.

3.2. Ridge Riding

Due to the local irregularities of natural textures, the inhomogeneity images present noisy information. The edge detection that allows to track the inhomogeneous regions cannot be based on a simple threshold of the inhomogeneity measure. It is better to provide a ridge riding scheme that would strengthen inhomogeneous areas that are extended along a certain direction.

The following ridge riding algorithm has been designed.

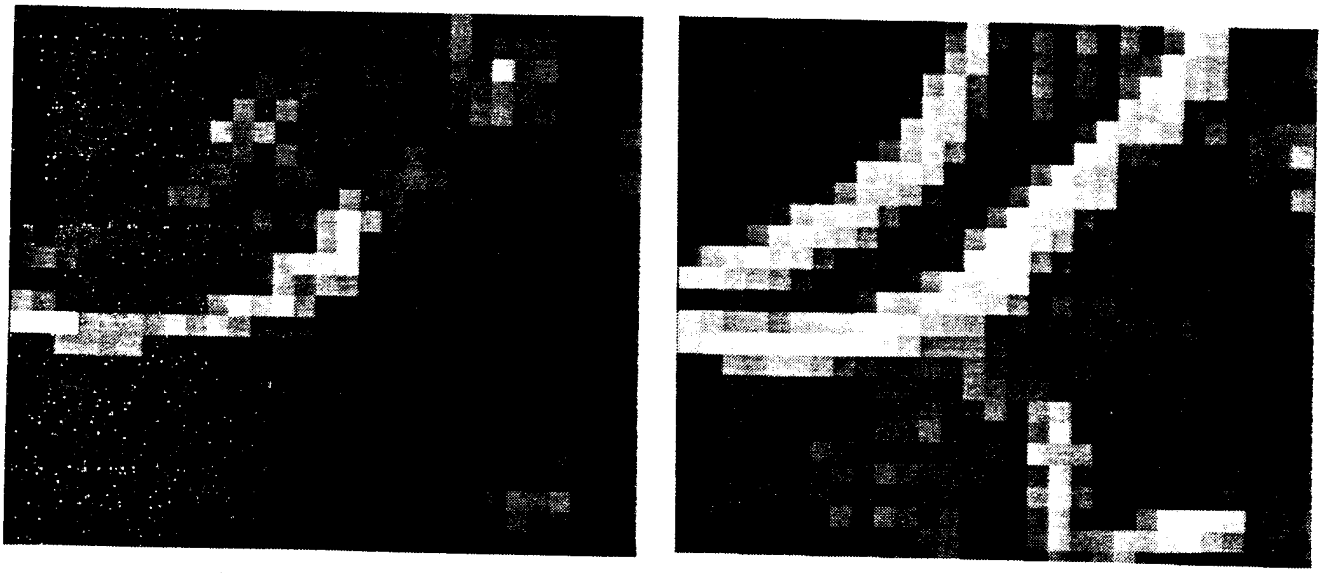
1) Start from the highest inhomogeneity measure above a certain threshold T_1 at location $P_0 = P_1$, which does not correspond yet to an edge point (every time a point is considered on a ridge, it defines an edge point). If no point left in the inhomogeneity image has a value larger than T_1, the algorithm terminates.

2) Among all P_1's neighbors in a 3x3 window, find the largest inhomogeneity value above another threshold T_2 and denote its location by $P'_0 = P_2$, that does not define yet an edge point or image boundary.

3) Track the next edge point (point with inhomogeneity measure above T_2) P_3 in a $-45° + 45°$ range of the direction P_1P_2. Redefine P_1 as P_2 and P_2 as P_3 and repeat the same step as long as no edge point or image boundary has been reached.

4) Search in the opposite direction of the designed path starting at P_0 by redefining P_1 as P'_0 and P_2 as P_0 and then going back to step 3). If such a search has already been performed return to step 1).

Since a requirement of segmentation is that all regions must be closed, any isolated edge segment may be removed. An automatic strategy for choosing the threshold values T_1 and T_2 is essential to guarantee an unsupervised segmentation scheme. If the set inhomogeneity values are plotted in increasing order, the corresponding curve is seen to have a maximum curvature point. It is reasonable to assume that the associated value represents a good threshold between homogeneous and nonhomogeneous areas. It defines an appropriate choice for T_1, at which a ridge may start. Finding a value for T_2 is a much more delicate problem. The value should be set according to the intraset variation of the feature vector representing each texture. For simplicity, it has been set to $T_1/2$.



a) Euclidean distance

b) Logarithmic distance (4)

Figure 6: Inhomogeneity images.

4. FINE BOUNDARY DETECTION

Once the homogeneous area of the image have been separated from each other, the segmentation problem becomes supervised. Each homogeneous area can serve as a core to refine the boundaries. Several different schemes can be found to refine the segmentation [14,16,17]. Most suggest the use of Markov Random Fields (MRF) or autocorrelation models. The corresponding algorithms are often iterative and of high computational cost. These previous works have studied the general disadvantage of methods applied to the feature space, where spatial relations between connected areas are lost. This drawback may to a certain extent be overcome by using the region locations as additional features.

Another way suggested here is to constraint the classification of parts of the inhomogeneous areas to only their potential neighbors. In other words, the feature vector of a certain area is compared only to a reduce number of classes (i.e. textures) in the feature space. These classes represent the set of textures in the image plane that are connected to the area under consideration.

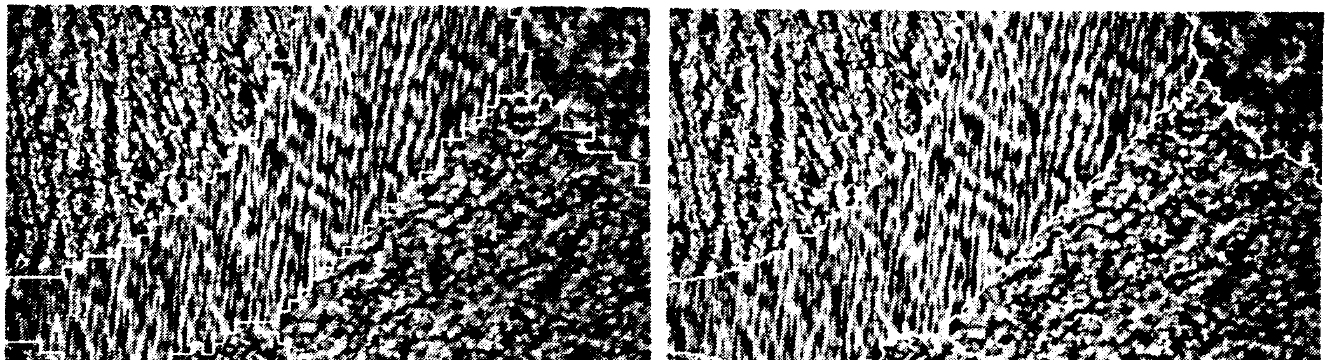
In order to provide a boundary with a very fine precision, a hierarchical classification has been used. The set of 64×64 blocks that correspond to inhomogeneous regions are first separated into 16×16 blocks. Each 16×16 block is then classified as one of its four most neighboring texture in the image plane which it is the closest to in the feature space. If as a result of this classification, such a block does not share the same texture class of its connected neighbors, the classification process is repeated among the set of its connected neighbors. After this first step, the boundary between different areas is obtained with a 16 image point precision. Assuming the clustering operation has been performed correctly, the uncertainty in the boundary location has been reduced from a set of 64x64 blocks to the set of neighboring 16x16 blocks that belong to different texture classes. It is now possible to repeat the same constrainted classification procedure to the set of 8x8 blocks that form these 16x16 blocks. The same idea is carried out down to 1x1 blocks.

The distance measure used to classify each feature vector to one of the prototypes in the feature space is the inhomogeneity measure suggested in Section 3. Prototype vectors are obtained by estimating the second-order moments of the Hadamard filtered 2x2 neighborhoods over the different core areas.

In order to reduce the risk of combining two different textures when estimating the different feature vectors for the various size blocks, it is desirable to compute the different second-order moments over the size of such blocks. This may tend to increase the variance of the feature estimation. The constrainted classification procedure should compensate for this drawback.

5. RESULTS AND DISCUSSION

Figures 7a and 7b show the result of the segmentation refinement procedure for the texture image of Figure 3, when an 8×8 and a 1×1 precision are required, respectively. The results are very encouraging, even at very fine resolution levels (1×1). For 1×1 blocks, the estimation of the second-order moments is not very meaningful. This shows the potential of a hierarchical clustering procedure that uses region connectivity as a constraint. It is interesting to notice, that most errors are related to misclassifications of larger size blocks. It is also interesting to notice that our segmentation scheme fails where the human eye has difficulty tracking the various texture boundaries. The boundary refinement procedure suggested here should be compared to different classical supervised segmentation schemes.



a) 8x8 boundary precision b) 1x1 boundary precision

Figure 7: Segmentation results for Figure 3.

Different improvements can be suggested to increase the performances of this segmentation method. It is desirable to find a better criterion for setting threshold T_2 in the ridge riding of the inhomogeneity image. Another ridge-riding strategy is actually under study, in which only one design parameter has to be set: the number of different textures that one would like to discriminate.

A better choice for the boundary refinement procedure could have been provided by using the Mahalanobis distance [19] or an optimal pattern classification procedure such as the GFLD [12-13]. The Mahalanobis distance should perform properly if the feature vectors for each texture are relatively well modeled with a jointly Gaussian distribution. This is indeed a reasonable assumption since they are the result of a linear filtering operation applied to a stationary signal.

If the criteria given in [18] to judge the efficiency of unsupervised texture segmentation algorithms are applied here, the proposed scheme is in the category of *high resolution*, it has a *low computational cost*, and it requires a relatively *low memory cost.*

Three main ideas have been introduced in this work:

1) A sequential approach in which homogeneous texture area are first separated from each other by the detection of inhomogeneities in the texture image and then these areas are extended as far as possible so as to obtain accurarate texture boundaries.
2) The one-dimensional nature of boundaries between different textures is utilized which allows the discrimination between local texture irregularities and true boundaries.
3) Connectivity and neighborhood constraints are used to achieve accurate segmentation results.

References

[1] D.H. Ballard and C.M. Brown, *Computer Vision*, Prentice-Hall, 1982.

[2] R.M. Haralick, "Statistical and Structural Approaches to Texture", *Proceedings of the IEEE*, Vol. 67, No. 5, pp. 786-804, May 1979.

[3] N. Ahuja and A. Rosenfeld, "Mosaic Models for Textures", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-3, No. 1, pp. 1-11, Jan. 1981.

[4] M. Unser, "Local Linear Transforms for Texture Measurements", *Signal Processing*, Vol. 11, pp. 61-79, 1986.

[5] P. Brodatz, *Textures - A Photographic Album for Artists & Designers*, Dover Publications, 1966.

[6] B. Julesz et al., "Inability of Humans to Discriminate Between Visual Textures That Agree in Second-Order Statistics", revisited, *Perception*, Vol. 2, pp. 391-405, 1973.

[7] A. Gagalowicz, *Vers un Modèle de Texture*, Université Pierre et Marie Curie, Paris VI, Thèse, 1983.

[8] R.M. Haralick, K. Shammugan and I. Dinstein, "Textural Features for Image Classification", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-8, No. 6, pp. 610-621, Nov. 1973.

[9] F. Ade, "Characterization of Texture by Eigenfilter", *Signal Processing*, Vol. 5, No. 5, pp. 451-457, Sept. 1983.

[10] G.H. Granlund, "Description of Texture Using the General Operator Approach", *Proc. of the 5th Intern. Conf. on Pattern Recognition*, pp. 776-779, 1980.

[11] D. Cano and T. Ha Minh, "Texture Synthesis Using Hierarchical Linear Transforms", *Signal Processing*, Vol. 15, No. 2, pp. 1??-1??, Sept. 1988.

[12] D.W. Peterson and R.L. Mattson, "A Method for Finding Linear Discriminant Functions for a Class of Performance Criteria", *IEEE Transactions on Information Theory*, Vol. IT-12, pp. 380-387, Jul. 1966.

[13] M. Unser and M. Eden, "Multi-Resolution Feature Extraction and Selection for Texture Segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, to be published.

[14] P.C. Chen and T. Pavlidis, "Segmentation by Texture Using Correlation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-5, No. 1, pp. 64-69, Jan. 1983.

[15] T. Pavlidis, *Structural Pattern Recognition*, Springer-Verlag (New York), 1977.

[16] P.C. Chen and T. Pavlidis, "Image Segmentation as an Estimation Problem", *Computer Graphics and Image Processing*, Vol. 12, pp. 153-172, 1980.

[17] F.S. Cohen and D.B. Cooper, "Simple Parallel, Hierarchical and Relaxation Algorithms for Segmenting Non Causal Markovian Random Fields", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 2, pp. 195-217, Mar. 1987.

[18] D. Wermser and N. Lissel, "Comparison of Algorithms for Unsupervised Segmentation of Images by the Use of Texture Information", *Proceedings of the 2nd Eusipco Conference*, Erlangen, pp. 287-290, 1983.

[19] J.T. Tou and R.C.Gonzalez, *Pattern Recognition Principles*, Addison-Weley, 1974.