Pergamon

Pattern Recognition, Vol. 28, No. 5, pp. 695–709, 1995
Elsevier Science Ltd
Pattern Recognition Society
Printed in Great Britain
0031–3203/95 $9.50+.00

0031–3203(94)00133–2

HIERARCHICAL IMAGE SEGMENTATION BY MULTI-DIMENSIONAL CLUSTERING AND ORIENTATION-ADAPTIVE BOUNDARY REFINEMENT*

P. SCHROETER† and J. BIGÜN
Signal Processing Laboratory, Swiss Federal Institute of Technology, CH-1015 Lausanne, Switzerland

(*Received* 13 *August* 1993; *in revised form* 13 *September* 1994; *received for publication* 13 *October* 1994)

Abstract—In this paper we present a new multi-dimensional segmentation algorithm. We propose an orientation-adaptive boundary estimation process, embedded in a multiresolution pyramidal structure, that allows the use of different clustering procedures without spatial connectivity constraints. The presence of noise in the feature space, mainly produced by modeling errors, causes a class-overlap which can be reduced in a multiresolution pyramid. At the coarsest resolution level, the separation between the different classes is increased and the within-class variance reduced. Thus, at this level, the classes can be obtained with different multi-dimensional clustering algorithms without connectivity constraints. Small and scattered classes as well as isolated class labels are reassigned to their neighborhood by a process which guarantees the spatial connectivity. The resolution is then increased by projecting down the class labels. At each level, the borders are improved by reassigning the boundary pixels to their spatially closest class. However, the class-uncertainty astride the borders has first to be reduced, and we propose to do this by means of orientation-adaptive butterfly-shaped filters. This refinement process further eliminates spatially misclassified pixels produced by the unconstrained clustering. Experimental results show that similarly accurate boundaries are obtained with different clustering algorithms for various test images.

Hierarchical image segmentation Texture segmentation Pyramids
Multi-dimensional clustering Deterministic relaxation Orientation-adaptive filters
Boundary refinement

INTRODUCTION

Image segmentation is an important tool in image analysis, but its general applicability is still a problem, particularly in the case of multi-dimensional features. Progress in texture analysis has led to models which allow us to describe increasingly complex structures, thereby leading to high dimensional feature spaces. In this paper we address unsupervised image segmentation, where "unsupervised" means that no training of the classifier takes place. Below, we will simply assume that the features are given. Detailed examples of applicable feature extraction methods related to cooccurrence matrices and texture energy masks can be found(1–3) while examples of methods related to Gabor decomposition are given.(4–6) Examples for the use of multi-spectral imagery features in classification are discussed.(7,8)

Rosenfeld *et al.*'s relaxation labeling,(9) is an iterative method for grouping together image points into a collection of classes. The classification is achieved by updating class membership probabilities with a compatibility function which must be defined heuristically. However, to choose a good stopping criterion as well as a suitable compatibility function,(10,11) is not obvious in the segmentation problem we deal with.

Clustering techniques without spatial connectivity constraints, e.g.,(12–14) which have found successful applications in many scentific domains, cannot directly be used in image segmentation. The reason for this is that a feature space partitioning does not necessarily yield connected labels in the spatial domain. Thus, when applied to image segmentation, a connectivity constraint becomes necessary.(15,16)

Nagy and Tolaba,(8) treated strips of scan lines as candidates of homogeneous regions. The scan lines are decomposed into sub-lines,(17) whereas the continuity of the sublines is preserved across the scan lines. Haralick and Dinstein,(18) introduced a so called gradient image which reflects the in-homogeneity of the multi-dimensional features. The segmentation is obtained by a thresholding followed by cleaning up in this result. A disadvantage of these boundary based methods is the too many small regions in the segmentation result. In a bottom-up procedure, Gupta and Wintz,(19) proposed to form spatial blobs by merging together subregions with features passing variance and means criteria (*F*- and *t*-tests). The segmentation is obtained when the updating of the blobs has converged. However, the growing directions of the blobs are scanning direction biased.

* This work has been supported by Thomson-CSF, Rennes, France.

† Author to whom all correspondence should be addressed.

Another technique is to combine pyramidal image smoothing with clustering. Horowitz and Pavlidis' split-and-merge technique,(20) builds a quadtree feature pyramid. Starting from a certain resolution level, it ascends (merges) or descends (splits) at possible regions depending on whether the features of the children fulfil a homogeneity criterion or not. The segmentation is obtained when all encountered children and parents fulfil a homogeneity criterion. An inconvenience is that the class boundaries may have a blocky appearance. Another quadtree approach is proposed by Spann and Wilson.(21) The segmentation is done by unconstrained clustering at a given resolution level. The classes are projected down while progressively refining the boundaries. The clustering is performed by successive transformations of the feature space which replace every point with its corresponding local feature space centroid. Due to the convergence instabilities at this stage, it may happen that no classes at all are detected. However, correct classes may be detected at another starting resolution or by using fewer dimensions in the same data.

In the past few years, there has been increasing interest in the use of statistical techniques for texture modeling and segmentation.(22–24) In such approaches, the textures and the regions are modeled by Markov Random Fields (MRF) and the segmentation is obtained by using a maximum *a posteriori* (MAP) criterion. In unsupervised segmentation, a parameter estimation scheme needs first to be applied to estimate the model parameters. The MAP parameters can then be found by means of simulated annealing. Good segmentation results were obtained on synthesized textures and on some natural textures that fit the models. However, the main restriction in such stochastic algorithms is the heavy (sometimes prohibitive) computational cost.

In this paper we present a multi-dimensional image segmentation paradigm based on the framework (quadtree, clustering and boundary estimation) suggested by Spann and Wilson.(25) We propose an orientation-adaptive boundary estimation process, embedded in a multiresolution structure, that allows the use of different clustering procedures without spatial connectivity constraints. The class separation at the coarsest resolution level is significantly increased so as to make possible the detection of the different classes and of their associated prototypes. A prototype is defined as the most representative feature-vector of a class. The classes and the prototypes are obtained by using a clustering technique which in general is allowed to be any clustering algorithm without spatial connectivity constraints. If the image size is too large even at the lowest resolution, the different classes can be found using a reduced data set obtained by random sampling. Small and scattered classes as well as isolated class labels are reassigned to their neighborhood by a process which guarantees the spatial connectivity. The resolution is gradually restored by projecting down the class labels. We use orientation-adaptive butterfly-shaped filters to reduce the class-uncertainty astride the borders. Then, at each level, the borders are improved by reassigning the boundary pixels to their spatially closest class. This refinement process further eliminates misclassified pixels produced by unconstrained clustering.

In the next section, the feature extraction process is briefly explained. Section 4 states the problem of spatial uncertainty in image segmentation and Section 5 gives a description of the segmentation algorithm. In the subsequent sections, the pyramid building (Section 6), the clustering (Section 7), the spatial continuity restoration (Section 8) and the oriented-adaptive boundary refinement processes (Section 9) are explained in details. Experimental results are finally presented in Section 10.

FEATURE EXTRACTION

In order to judge the quality of the clustering and boundary refinement processes, we will use the same feature set throughout this paper. We have selected complex moments of the Gabor power spectrum as features because of their ability to discriminate textures in real images. This method is briefly summarized below and more details as well as a review of different feature extraction techniques are given elsewhere.(26,27)

The complex moments

$$I_{mn} = \int_{-\infty}^{\infty}\int (x+jy)^m (x-jy)^n P(x,y)\,dx\,dy = \int_0^{\infty}\int_0^{2\pi} r^{m+n} e^{j(m-n)\varphi} P(r,\varphi) r\,dr\,d\varphi \qquad (1)$$

effectively represent the symmetry properties of the 2D function P, where $P(x, y)$ is the local power spectrum. Thus x and y as well as r and φ are frequency coordinates. We can discretize (1) by approximating the continuous spectrum P by means of a limited set of Gabor filters. The order of the complex moments is determined by $m + n$ whereas the measured symmetry order is given by $m - n$. For a linear symmetry $m - n = 2$ and this corresponds to structures with one dominant orientation. Rectangular structures have a 4-folded symmetry, triangular and hexagonal structures a 6-folded symmetry, and so on. We use up to the 6th order complex moments applied to octave frequency bands, resulting in 15 real-valued features which represent the real and imaginary parts of I_{mn}. Note that complex moments of order zero are real. The feature set also comprises the DC-band.

Although complex moments can code the power spectrum efficiently, the dimensionality of the feature space can be further reduced by decorrelating the features. The technique used [see(28)] decomposes the multidimensional data set into its principal components by taking into account the spatial arrangement of the features. The transformed features are ordered according to their variances and only those with a variance less than a given threshold value are selected. This procedure results in six features, typically.

THE PROBLEM OF UNCERTAINTY IN IMAGE SEGMENTATION

The problem of uncertainty in image segmentation is analog to the principle of uncertainty in physics and signal processing which states that "a signal cannot be highly concentrated simultaneously in space and frequency". In unsupervised image segmentation, both the prototypes and the class boundaries have to be determined. These two processes are highly interdependent because:

- well defined prototypes are needed to find accurate boundaries, and
- accurate boundaries are needed to compute the prototypes.

Therefore the properties and the boundaries cannot be simultaneously well defined.[29] If these two processes were independent, it would make no difference what is computed first, the prototype or the class boundaries. In that case, edge-based and region-based segmentation methods would give the same results. The presence of noise in the feature space, mainly produced by modeling errors (inappropriate features), is another source of uncertainty which produces a class-overlap.

DESCRIPTION OF THE SEGMENTATION ALGORITHM

The algorithm is embedded in a multiresolution framework using quadtrees.[21,30] Multiple resolutions are very useful in reducing the uncertainty because at lower resolutions the class-prototypes are better defined while higher resolutions are needed to obtain accurate borders. A pyramid is built up to a predefined level in which each coarser level is obtainned by smoothing the preceding finer level. At the coarsest level the amount of noise in the feature space has decreased significantly allowing the prototypes to be determined, at the expense of the spatial resolution. This is illustrated in Fig. 1 [see[21]]. At the finest level the histogram is unimodal even if it is easy to see that two distinct classes are present. At a lower resolution the noise has been smoothed out significantly making possible the detection of the two classes (the histogram is bimodal).

The next step is to find a partition of the feature space into c classes. This operation can be seen as finding c subsets of feature vectors gathered around their respective prototypes (or class centers). This partitioning is obtained by applying a clustering algorithm in the smoothed feature space (obtained at the coarsest level). As the noise has been reduced, algorithms without spatial connectivity constraints can be used. For most of these algorithms, the number of classes c has to be known *a priori*. Our experiments indicate that the choice of a clustering algorithm is not very critical and that similar segmentation results are obtained with different algorithms. Small and scattered classes as well as isolated labels are reassigned to a class nearby using an 8-connected neighborhood assumption that guarantees the spatial connectivity.

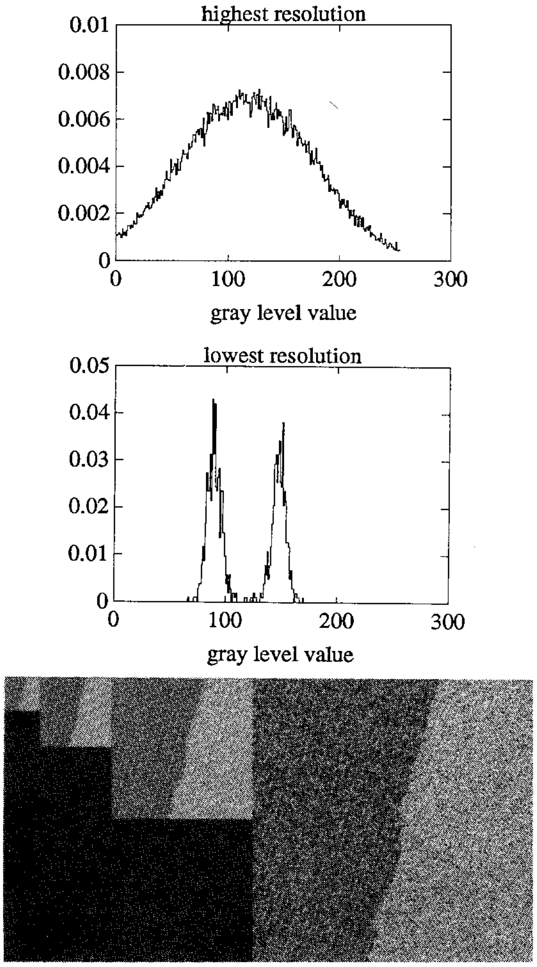


Fig. 1. Reduction of uncertainty using pyramids. We added Gaussian noise to the original image with standard deviation $\sigma = 50$.

The last step is a boundary estimation procedure that gradually improves the class-boundaries. First, at the level where the clustering is performed, the boundaries are identified. The children of the boundary nodes define a boundary region at the next higher resolution. The non-boundary nodes at the children level are given the same labels and properties as their parents. The class-uncertainty within the boundary region is high and has to be reduced before reassignment of the boundary vectors. We propose to do this by means of orientation-adaptive filters. For each dominant local orientation, a butterfly-like filter (see Section 9) is defined. Butterfly-like shape reduces the influence of feature-vectors along the boundary. Then, for each boundary pixel, the filter corresponding to the local orientation is applied (see Fig. 2). The two halves of the filter (applied separately) produce two responses. The distances between these two vectors and the two prototypes, associated to the classes defining the bound-

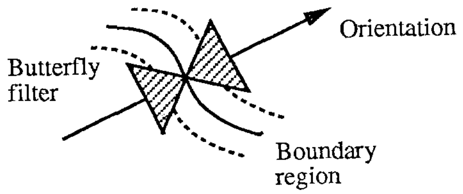


Fig. 2. Illustration of the oriented butterfly filters.

ary, are computed. Then, each boundary vector is reassigned to the closest class.

The algorithm can be summarized as follows:

(1) build up a multiresolution pyramid. The noise in the feature space is reduced, increasing the separation between the classes at the expense of the spatial resolution.

(2) Cluster the data in the smoothed feature space by using algorithms without spatial connectivity constraints. Reassign isolated pixels as well as small and scattered classes using an 8-connected neighborhood assumption.

(3) Gradually improve the spatial resolution by projecting down the labels and refine the boundaries using orientation-adaptive filters.

PYRAMID BUILDING

The noise reduction can be done by means of quadtrees.[30] Lower resolution levels are obtained by projecting the average of non-overlapping squares of size 2 × 2 to the next coarser level. Let $I_p(i, j, l)$ be the value of the p-th feature at location (i, j) and level l of the pyramid. The value of a father node is simply the mean of its four children value.

$$I_p(i,j,l) = \frac{1}{4} \sum_{m=0}^{1} \sum_{n=0}^{1} I_p(2i+m, 2j+n, l-1) \qquad (2)$$

The different levels are computed using (2) in a bottom-up manner starting from level $l = 1$ up to a predefined level ($l = l_{\max}$). The height and width of the image decrease with a factor of 2^{l-1} at level l. A pyramid is constructed for each feature separately. Therefore, the data structure can be seen as a set of pyramids or as a single one composed of vectors of size p. The choice of the number of levels is important. If $l_{\max}$ is too small the uncertainty is not enough reduced, whereas if l is too high, small regions will disappear. Pyramids also reduce the computational cost by progressively reducing the number of feature vectors on which a clustering algorithm will be applied. Consider for instance p feature images of size 256 × 256. If the number of levels is set to five then the number of pixels is reduced from $65536 \times p$ to $256 \times p$. $l_{\max}$ Should be chosen as a function of the noise rather than a function of the size. If the number of feature vectors even at the lowest resolution becomes prohibitive for a clustering algorithm, it still can be reduced by a random sampling of the image sites.

A drawback, related to the simple structure of the quadtree, is its incapacity to keep track of small regions while building up the pyramid. A tradeoff between the size of the smallest detectable region and the amount of noise reduction has to be found. An attempt to eliminate this problem using more complex hierarchical structures, such as a linked pyramid[31] is proposed by Spann and Horne.[32] In a linked pyramid, regions of all sizes can be segmented at the expense of a higher complexity of the structure. The method gives good segmentation results on gray level and color images. However, practical experience learned that it often has problems with multi-dimensional feature spaces (other than color-feature spaces).

CLUSTERING

At the coarsest level of the pyramid, a clustering algorithm is used to find the different classes and their prototypes. We assume that objects with similar properties belong to the same cluster, i.e. are gathered around the same prototype. The problem is to find a partition of the feature space into c homogeneous subsets. In order to solve this problem, a suitable clustering criterion and a similarity (or dissimilarity) measure have to be defined. It can be noticed in the literature that different criteria will lead to different results depending on the shape of the clusters. It is clear that no clustering criterion or measure of dissimilarity is universally applicable. For instance, the single link method which was introduced by Florek *et al.*[33] is more suitable for elongated clusters while WGSS (Whithin Group Sum of Squares) criterion gives good results for ball-shaped structures.[14] However, our method does not critically depend on the choice of a clustering technique. Experiments (Section 10) show that similar segmentation results are obtained with different algorithms. An overview of classical clustering techniques can be found in.[14,34]

The choice of the number of classes c, required by most of the clustering algorithms, is considered to be one of the most fundamental problems in cluster analysis.[12] Usually this information is not known and partitions of the feature space for different values of c are computed. The partition giving the best separation of the classes for a given clustering criterion is chosen. However, this approach can be very expensive in computation time and works mainly in cases where the data set forms compact and well-separated clusters. An overview of cluster validity and studies of this non-trivial problem can be found elsewhere.[35] Hierarchical clustering techniques do not require the number of classes to be known.[14,36] New clusters are formed by reallocation of membership of only one object (or one cluster) at a time, based on some measure of similarity. The result can be seen as a hierarchy of nested clusters represented by a tree called *dendogram*. Sectioning the tree diagram produces a partition of the data set into disjoint clusters. However, the formulation of the problem has only changed from knowing the number of classes to estimating an appropriate thresholding of the tree. In the following, we will consider clustering algorithms in which the number of classes c (or its equivalent for hierarchical clustering techniques) has to be given. We will reduce the problem of an exact estimation of the class number by over-estimating c and reassigning small and scattered classes to their neighborhood (see Section 8). In our experiments, we observed that similar segmentations can be obtained with different initial values of c (see Section 10). Below we will shortly present the fuzzy c-means clustering algorithm proposed by Bezdek(35) which gives good clustering results while being simple. For comparison we will use other multidimensional clustering algorithms as described in references (14, 34) (see Section 10).

Fuzzy c-means clustering algorithm

Let $X = \{x_1, \ldots, x_n\}$ be a finite set of vectors and c an integer $(2 \le c \le n)$ representing the number of classes. The c-partition of the feature-space is represented by the $c \times n$ matrix $U = [u_{ik}]$ where u_{ik} is the membership of object k to the class i, $k = 1, \ldots, n$ and $i = 1, \ldots, c$. In the hard (or crisp) case, the degree of belongingness of object x_k to a class is either 1 or 0, i.e.

$$u_{ik} = \begin{cases} 1 & x_k \in \text{class } i \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

In the fuzzy case, u_{ik} gives the "strength" of the membership of object k to class i $(u_{ik} \in [0, 1])$. The degree of belongingness u_{ik} can be intuitively seen as a distance from object k to the class i normalized by the sum of distances to the c class centers. This representation is in many cases closer to the physical reality in the sense that objects almost never fully belong to one class. The two following conditions have to be respected

$$\sum_{i=1}^{c} u_{ik} = 1, \quad \forall k \tag{4}$$

$$0 < \sum_{k=1}^{n} u_{ik} < n, \quad \forall i \tag{5}$$

In the crisp case, equation (4) simply means that object k belongs to one and only one class. The second condition (5) means that no class is empty and no class is all of X.

The fuzzy c-means algorithm belongs to the class of objective function methods. Such methods minimize a clustering criterion which is in that case, the total within-group sum of squared error (WGSS). The fuzzy c-means is the fuzzy extension of the hard c-mean [basic ISODATA(37)]. It minimizes the family of objective functions given by

$$J_m(U, \mathbf{v}) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m (d_{ik})^2 \tag{6}$$

where $U = [u_{ik}]$, $u_{ik} \in [0, 1]$ is a fuzzy c-partition of X, $\mathbf{v} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c\}$ is the set of cluster centers, $(d_{ik})^2 = \|\mathbf{x}_k - \mathbf{v}_i\|^2$ and $\|\cdot\|$ is any inner product induced norm on $\mathbb{R}^p$. m is the weighting exponent controlling the amount of fuzziness, $m \in [1, \infty[$. Restated, our problem is to find the best pair $(U, \mathbf{v})$ which minimizes J_m. Bezdek(35) showed that J_m may be globally minimal for $(U, \mathbf{v})$ only if

$$u_{ik} = \left[\sum_{j=1}^{c} \left(\frac{d_{ik}}{d_{jk}} \right)^{2/(m-1)} \right]^{-1} \tag{7}$$

$$\mathbf{v}_i = \sum_{k=1}^{n} (u_{ik})^m \mathbf{x}_k \Big/ \sum_{k=1}^{n} (u_{ik})^m, \quad \text{for } 1 \le i \le c \tag{8}$$

The fuzzy c-means algorithm which approximates a solution of the minimization problem can be stated as follows:

Fuzzy c-means:

(1) Fix c, choose any inner product norm metric for $\mathbb{R}^p$ and fix m, $1 < m \le \infty$ then initialize $U^{(0)}$.

(2) Calculate the c fuzzy cluster centers $\{\mathbf{v}_i\}$ with equation (8) and U.

(3) Update U using equation (7) and $\{\mathbf{v}_i\}$.

(4) If $\|U^{(l+1)} - U^{(l)}\| \le \epsilon$ then stop: otherwise goto 2.

As $m \to 1$, the fuzzy c-means converges to a "generalized" hard c-means solution (ISO-DATA). Bezdek demonstrated that this algorithm *always* reaches (in theory) a strict local minimum for different initializations of U.

RESTORATION OF THE SPATIAL CONTINUITY

At the end of the clustering process, each vector belonging to the same class receives the same label γ_i $(1 \le i \le c)$. Thus, at the coarsest resolution level $l_{\max}$, we can define a label image $I_\gamma(i, j, l_{\max})$ in which each pixel receives the label of the corresponding vector in the feature space. At this level, rough boundaries can be observed between the different classes. It will also be noticed (see Section 10) that some pixels are spatially misclassified because no spatial connectivity constraints were present in the clustering algorithm. Let $N_8(i, j)$ be the neighborhood of a pixel at location (i, j) composed of the 8 closest neighbors, i.e. $N_8(i, j) = \{(i + u, j + v)\}$, $-1 \le u, v \le 1$ and $(u, v) \ne (0, 0)$. A pixel (i, j) is considered as spatially misclassified if $I_\gamma(i, j, l_{\max})$ is different from all the labels in $N_8(i, j)$. In that case, it is reassigned to the most represented class in $N_8(i, j)$.

Small and scattered classes ("insignificant classes") are reassigned to their neighborhood. A class in the feature space is distributed in one or more subregions in $I_\gamma(i, j, l_{\max})$ and is considered as "significant" only if its largest subregion contains a sufficient number of connected pixels (nine in our experiments). Thus, a preference is given to classes that are spatially distributed in larger and more compact subregions. It is clear that the meaning of "insignificant classes" is

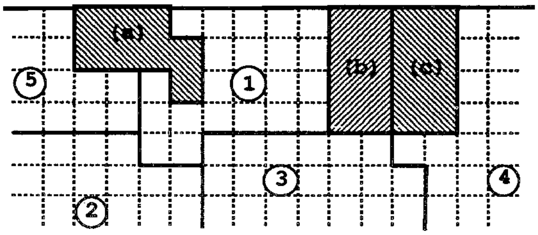


Fig. 3. Example of three classes that have to be reassigned (hatched regions).

closely related to the height of the pyramid, i.e. for a low value of $l_{\max}$, the actual size of an insignificant class is smaller than for higher values of $l_{\max}$.

Consider, for instance, the hatched regions in Fig. 3. These classes fail our test of "significance" and have to be reassigned to either one of the surrounding classes or, in case of (b) and (c), between themselves. The reassignment procedure can be seen as a relaxation problem. In our case we will use a simple deterministic multi-passes approach. We define an isolated class as an "insignificant class" which does not share common boundaries with other "insignificant classes". Such classes can be reassigned in one pass whereas multi-passes are needed if the insignificant classes are touching each other. First, we reassign the isolated classes like C_a, represented by the n_a pixels of region (a) (Fig. 3). A set of candidate classes, among the neighboring classes of C_a is determined for each pixel $(i, j) \in C_a$. If a label $I_y(i+u, j+v, l_{\max}) \in N_8(i, j)$ is different from $I_y(i, j, l_{\max})$ then a new candidate class is obtained. Otherwise, the neighborhood is extended and $I_y(i, j, l_{\max})$ is compared to the next label $I_y(i+2u, j+2v, l_{\max})$ in the same direction. These operations are repeated for the eight possible directions, $-1 \leq u,\ v \leq 1$. For each pixel $(i, j) \in C_a$, the Euclidean distances between the feature-vector corresponding to (i, j) and the prototypes of the different candidate classes are computed. Each pixel is reassigned to the closest candidate class. It is important to extend the neighborhood in which the candidate classes are determined. This can be seen by inspection of Fig. 3. The central pixel of (a) near the border of the image or the pixel of (b) and (c) at the center of the common boundary lack neighbors belonging to significant classes. For these pixels, an extended neighborhood is necessary for obtaining relevant candidate classes.

Finally, the non-isolated classes are reassigned. However, the reassignment order leads to different results and multi-passes are needed to avoid this problem. This process can be illustrated by considering, for instance, the classes (b) and (c). In the first pass, these two classes are temporarily reassigned as if they were isolated classes. The same initial state of $I_y(i, j, l_{\max})$ is used for both classes. The maximum number of pixels $n_{i,\max}$ reassigned to the same candidate class is determined for $i =$ (b), (c). Then a ratio r_i between $n_{i,\max}$ and the respective population n_i of the classes is computed, i.e. $r_i = n_{i,\max}/n_i$, $i =$ (b), (c). A value of $r_i = 1$ means that the class i tends to be reassigned to a single candidate class. The reassignment of the region with the largest ratio r_i is then validated. Suppose that (b) is reassigned first. Before reassigning (c) in a second pass, we need to check if (c) is still considered as an "insignificant class". It might be the case that pixels of (b) were reassigned to (c) making it become a significant class. If this is not the case, (c) is reassigned using the procedure described for isolated classes. The same procedure can be applied for more than 2 non-isolated classes. It only takes more passes until all "insignificant classes" are reassigned.

We could have also used stochastic relaxation and Markov Random Fields (MRF)[22] to restore the spatial connectivity. The principle of such technique is similar to the technique proposed in the preceding paragraphs in the sense that, MRF approaches consist in minimizing an energy function accounting for mutual adjacent sites labeling. At the coarsest resolution level, the image size is smaller making the computational cost acceptable. However, the proposed deterministic approach converges rapidly toward a solution while being simple and works and the results are satisfactory. Moreover, MRF approaches would be very heavy in computational time for low values of $l_{\max}$ and for high values of the number of classes ($c > 4$).

BOUNDARY REFINEMENT

The last step of the algorithm is a boundary refinement procedure that gradually improves the spatial resolution of the label image $I_y(i, j, l_{\max})$. As stated by the spatial principle of uncertainty,[29] accurate prototypes can be obtained only at the expense of the spatial resolution. Therefore, at the coarsest level, the uncertainty astride the boundaries is high. We propose to reduce it by means of orientation-adaptive filters.[38] Another method for edge enhancements that uses non-isotropic filters can be found in reference (39).

The spatial resolution is gradually restored by projecting down the class labels, smoothing around the boundaries and reassigning the boundary pixels to their closest neighboring class. We also assume that the prototypes have constant values across the different levels of the pyramid.

First, at the coarsest level $l_{\max}$, the boundary pixels are determined. Each pixel (i, j) is considered as a boundary pixel if at least one label is $N_8(i, j)$ is different from $I_y(i, j, l_{\max})$. $I_y(i, j, l_{\max}-1)$ is obtained by projecting down the label of each non-boundary father node to its 4 respective children nodes, i.e. $I(i, j, l) = I(i/2, j/2, l+1)$, where / is the integer division. The children of the boundary nodes define a boundary region β (see Fig. 4) in which the oriented smoothing will be performed. For smoothing we need an estimate of local boundary orientations. For its simplicity we use the linear symmetry algorithm[4] which defines the dominant

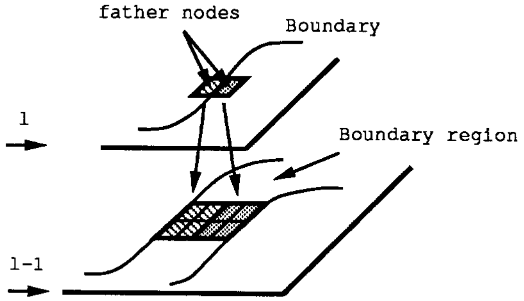
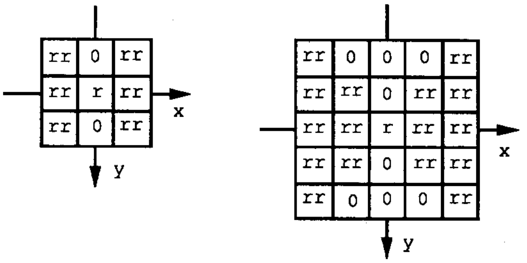


Fig. 4. Boundary region defined by the children nodes corresponding to the boundary father nodes.



rr	0	rr
rr	r	rr
rr	0	rr

rr	0	0	0	rr
rr	rr	0	rr	rr
rr	rr	r	rr	rr
rr	rr	0	rr	rr
rr	0	0	0	rr

Fig. 5. Butterfly-like filter of size 3 × 3 and 5 × 5 defined for a local orientation of $\Theta = 0$.

orientation in the least squares sense. The orientation is determined by the complex number

$$z_1 = (\nabla \mathbf{I}_y)^2 * m \tag{9}$$

where I_y is the image of labels, $\nabla \mathbf{I}_y$ the complex image $(\partial I_y/\partial x) + i(\partial I_y/\partial y)$ and $*m$ the convolution with an averaging filter. The argument of z_1 obtained at every pixel location represents

$$\arg(z_1) = 2\arg(k_{\min}) \tag{10}$$

where $k_{\min}$ is the complex number whose argument is the dominant orientation. In our case, the magnitude of the gradient of I_y is 1 at the transition between two classes, or 0 within a class. The smoothing function m is a Gaussian of size $s \times s$ given by

$$m = \exp\left(-\frac{u^2 + v^2}{2\sigma^2}\right) \tag{11}$$

where $-s/2 \leq u,\ v \leq s/2$. σ is determined assuming that $\exp(-3) \approx 0$ and is obtained using the relation $(s/2 + 1)^2 = 6\sigma^2$. In our experiments, we used a filter size of $s = 7$. The orientation is computed for the boundary pixels at the parent nodes level and is propagated to the children level.

For each dominant local orientation, a butterfly-like filter is defined. Butterfly-like shape reduces the influence of vectors along the boundaries which have high uncertainty. The shape and the weights of a 3 × 3 and 5 × 5 filter for the horizontal orientation ($\Theta = 0$) are given in Fig. 5, where r is a function of the dissimilarity between the two classes that define the boundary and $rr = (1 - r)/n$ with n being the number of weights different from 0 or r. The dissimilarity d is given by

$$d = \frac{|\mu_1 - \mu_2|}{\sqrt{\sigma_1^2 + \sigma_2^2}} \tag{12}$$

where μ and σ^2 are the mean and the variance of the two classes on both sides of the boundary. Then $r = \text{fct}(d)$ is obtained by the function of Fig. 6 which was experimentally found in reference 21. If the dissimilarity d is large then $r = 1$ and no smoothing is applied whereas a stronger smoothing is performed ($r = rr$) for low values of d.

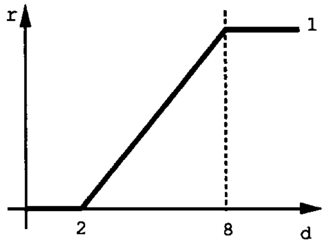


Fig. 6. r As a function of the dissimilarity d between two classes.

We still need to define the filters for different orientations Θ. This is done by rotating the mask defined for the horizontal orientation $\Theta = 0$ and redistributing the weights for matching the grid of an image. New coordinates of the mask are obtained by

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\Theta & \sin\Theta \\ -\sin\Theta & \cos\Theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{13}$$

The weights are then redistributed to the four pixels corresponding to the combination of the lower and upper integer values of x' and y'. Consider the following notation

$$\begin{matrix} x_f = \lfloor x' \rfloor & x_c = \lceil x' \rceil & w_x = x_c - x \\ y_f = \lfloor y' \rfloor & y_c = \lceil y' \rceil & w_y = y_c - y \end{matrix} \tag{14}$$

where $\lfloor \cdot \rfloor$ is the "floor function" that gives the truncated part of a real number and $\lceil \cdot \rceil = \lfloor \cdot \rfloor + 1$ is the "ceiling function". The redistribution of a unit weight is reported in table 1. The new filter is then defined by rotating and summing the contribution of all the weights of value rr (the value of the central weight remains r).

The filters are computed for a fixed number of orientations (eight in our case) and are stored in a look-up table. In our experiments, we noticed that similar segmentation results are obtained with different num-

Table 1. Coordinates and their associated percentage of a unit weight

Coordinates	Weight
(x_f, y_f)	$w_x w_y$
(x_f, y_c)	$w_x(1 - w_y)$
(x_c, y_f)	$(1 - w_x)w_y$
(x_c, y_c)	$(1 - w_x)(1 - w_y)$

bers of orientations (6, 7 or 8). Starting from the coarsest level $l = l_{\max}$, the boundary refinement procedure can be enumerated as follows:

(1) Project down the labels, i.e. $I_\gamma(i, j, l-1) = I_\gamma(i/2, j/2, l)$ and define the boundary region ß at level $l-1$. Compute the mean μ_i and variance σ_i^2 for all classes, $2 \leq i \leq c$.

(2) For each boundary pixel at level l compute the dominant orientation Θ using equations (9), (10) and (11), determine the two classes c_1 and c_2 on both sides of the boundary and compute the weights of the filters using equation (12), $r = \mathrm{fct}(d)$ and $rr = (1-r)/n$. Then, propagate the orientations to the corresponding pixels at level $l-1$.

(3) For each pixel $(i, j) \in$ ß, apply the filter corresponding to the current local orientation to each feature image $I_p(i, j, l-1)$. If a feature-vector is not an element of the boundary region i.e. it already belongs to one class, take the value of its corresponding prototype. The left and right halves of the filters are applied separately eliminating the problem of smoothing across the boundaries. Two responses are obtained, $r_1(i, j, l-1)$ and $r_2(i, j, l-1)$. This smoothing is repeated a certain number of times (four in our case) using a small filter size (3 x 3). This is similar than using a larger filter size in one iteration, but it is computationally faster.

(4) For each pixel $(i, j) \in$ ß, compute the four distances between the two filter responses $r_u(i, j, l-1)$ and the prototypes μ_v corresponding to the classes c_1 and c_2, i.e. $\|\mu_v - r_u(i, j, l-1)\|$, $1 \leq u\ v \leq 2$. Each boundary pixel receives the label of the class that gives the minimum distance.

(5) Decrease the value of l by one and repeat from step 1, until the bottom of the pyramid is reached.

EXPERIMENTAL RESULTS

We used two different "patch" images of size 256 x 256 (see Fig. 7). The first image, p1, is composed of regions taken from aerial images. Seven different textures were selected, consisting of fields, forests and a residential area. Furthermore, each texture was normalized with respect to the mean and variance. The second image, p2, is composed of patches of different shapes and sizes. The constituent textures are Oriental rattan (D65), wood grain (D68), raffia (D84), calf leather (D24), straw screening (D49) and pressed cork (D4) and can be found in Brodatz.[40] The complex moments of the Gabor power spectrum were computed on these two images resulting in 30 features. The six features with the lowest variance were selected, defining a 6-D feature space. Two of the features computed on p1 are given in Fig. 8. It can be first noted that the different patches are corrupted by a significant amount of noise making the discrimination between the classes difficult. This noise is mainly due to modeling errors and is progressively reduced in the pyramid. Experiments showed that a number of levels of 5 is a compromise between the noise reduction and the size of the smallest detectable region (see below). At the coarsest level, the number of vectors is therefore reduced from 256 x 256 to 16 x 16 and the clustering algorithm can rapidly partition the feature space. We used the fuzzy c-means algorithm with $c = 9$ and $m = 1.6$. We will consider $m = 1.6$ as a constant because good clustering results were obtained in all experiments with this value. The only two parameters that need to be specified are the number of levels $l_{\max}$ and the number of classes c. The result of the clustering operation is shown in Fig. (9a) where each class is represented by a different gray level. For convenience, a pixel at the coarsest level is represented by a square of size 16 x 16. It can be seen that some isolated pixels and insignificant classes have to be reassigned to a spatially neighboring class. However, already at this stage, the shape of the different regions defined by the labels corresponds roughly to the shape of the actual regions. The result of the spatial restoration is shown in Fig. (9b). Isolated pixels are removed and insignifi-

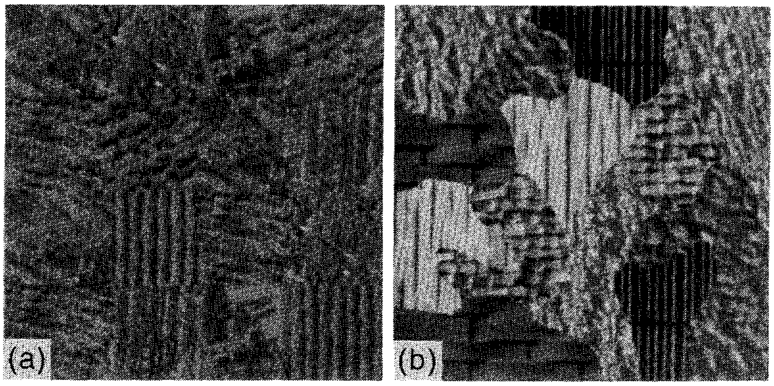


Fig. 7. Two test images. "p1" (left) is an image composed of patches taken from aerial images where each texture is normalized on mean and variance. "p2" (right) is composed of textured-patches of different shapes and sizes with Brodatz textures.

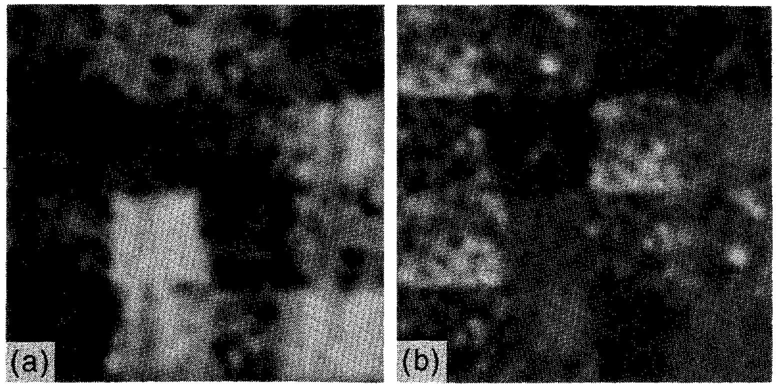


Fig. 8. Two features of "p1" computed by using the complex moments of the Gabor power spectrum.

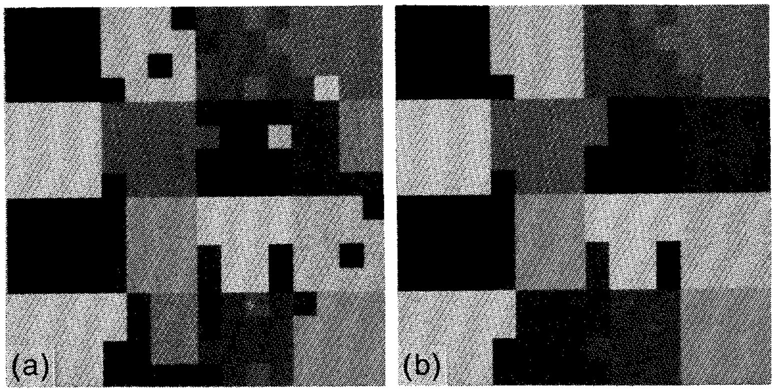


Fig. 9. (a) Results of the fuzzy c-means clustering algorithm applied at the coarsest resolution level $l = 5$: each class is represented by a different label value. (b) Results of the spatial restoration: isolated pixels and insignificant classes are reassigned.

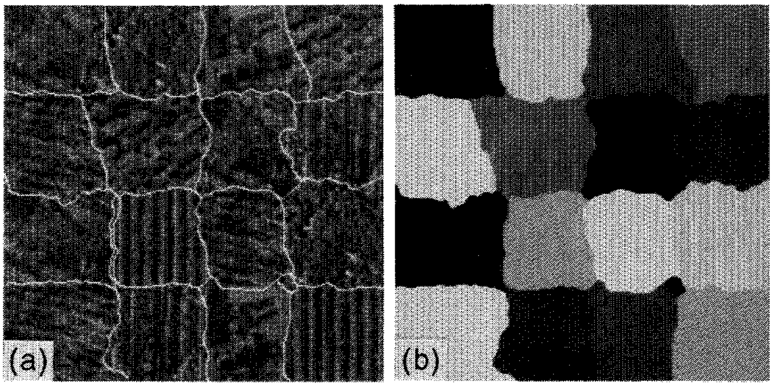


Fig. 10. Segmentation results of "p1" superimposed on the original image (a) and the corresponding class assignments (b).

cant classes reassigned, reducing the number of classes from 9 to 7. The boundary refinement procedure is then performed and the final segmentation can be seen in Fig. 10. All seven classes are found and the borders correspond well to those that can be seen by a human observer. The same operations with unchanged parameters are applied to p2 and the segmentation result is given in Fig. 11. Again, all six classes are correctly found and the borders correspond well to the "reality".

Figure 12 shows the segmentation results for different initial number of classes c. In all cases, the boundaries are very similar and the final number of classes

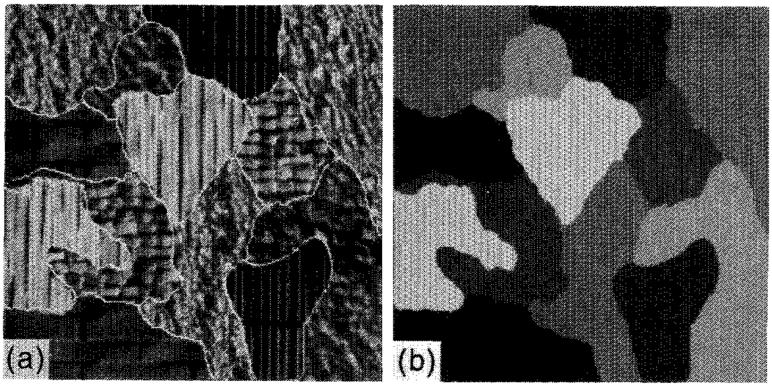


Fig. 11. Segmentation results of "p2" superimposed on the original image (a) and the corresponding class assignments (b).

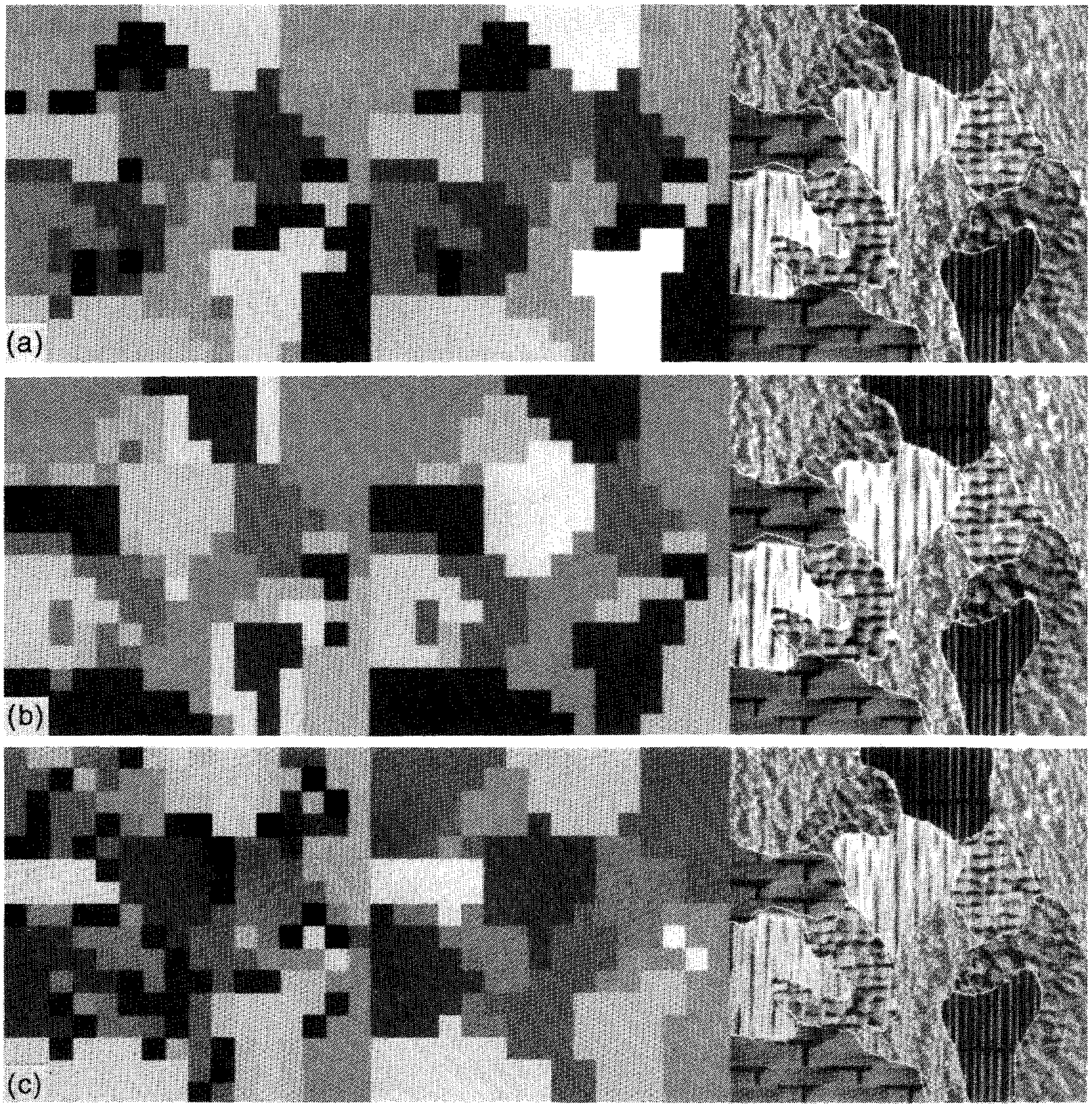


Fig. 12. Segmentation results obtained with an initial value of c of 6, 7 and 10 classes. The first column shows the results of the fuzzy c-means algorithm with $m = 1.6$ and the second, the results of the spatial restoration. In the last column is presented the segmentation superimposed on the original image. Six classes and similar boundaries are obtained in the three cases.

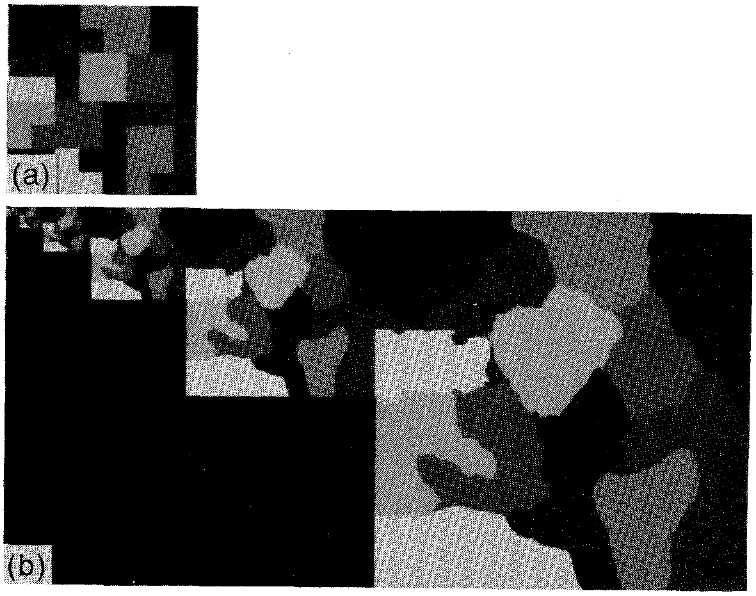


Fig. 13. (a) Spatial distribution of the labels computed on "p2" at level $l_{\max} = 6$ (one pixel at that level is represented by a square of size 16 × 16). (b) Levels of the pyramid of labels obtained by the boundary refinement procedure.

has been reduced to the true number, six. Even if c is set to 10, the restoration of the spatial connectivity produces a spatial distribution of labels very close to the one obtained with $c = 6$. Small changes can be observed and are further eliminated by the boundary refinement. Consider, for instance, the raffia textured region with the U-shape ($c = 10$). The spatial restoration procedure leaves part of the pixels misclassified and almost one half of the U is missing. However, at the end of the boundary refinement procedure, the correct segmentation is obtained and the missing half of the U is recovered. It is also worth noting that small misclassified subregions of two or three pixels disappear during the boundary refinement process. Figure 13 shows the labeling at the top of the pyramid ($l_{\max} = 6$) and the different levels obtained by the boundary refinement procedure. It is interesting to see how this procedure is able to recover the actual shapes of the regions even starting from a very rough description of the regions.

As already mentioned, the choice of the number of levels is important. Figure 14 shows the segmentation results for different values of $l_{\max}$. As one could expect, the noise at level $l_{\max} = 3$ is not enough reduced and too many subregions are detected. As the number of pyramid levels increases, these extra subregions progressively disappear and the expected segmentation is obtained for $l_{\max} = 5$ and 6. For higher values ($l_{\max} > 6$), the classes merge together because the sizes of the regions are too small. The same behavior can be observed on "p1".

Similar segmentation results can be obtained with a large variety of clustering algorithms as long as the clustering criterion matches correctly the shape of the clusters in the feature space. The best segmentation results were obtained by using the fuzzy c-means. For comparison, we tried three different agglomerative algorithms described in Gordon.[14] Agglomerative algorithms belong to hierarchical methods. They start when all objects are apart (the clusters contain a single pixel) and at each step, the two clusters with the smallest dissimilarity measure are merged. These algorithms are thus step-wise optimal and the clusters are hierarchically nested and can be represented by a dendogram. There exists many agglomerative algorithms, which only differ in the definition of the between-cluster dissimilarity. We have selected the general algorithm proposed by Lance and Williams[41,42] with the following criteria: (1) sum of squares, (2) centroid and (3) complete link (farthest neighbor). At each step of the algorithm, the two "closest" clusters i, j are merged and the dissimilarities between the newly formed (i, j) and and the k others are updated using the following recursive relation

$$d_{k(i,j)} = \alpha_i d_{ki} + \alpha_j d_{kj} + \beta d_{ij} + \gamma |d_{ki} - d_{kj}| \qquad (15)$$

where α_i, α_j, β and γ are parameters specifying the particular strategy employed (see Table 2), and d_{ij} is the squared Euclidean distance between the ith and jth cluster.

The sum of squares criterion minimizes the within-group variance, i.e. minimizes equation (6) with $u_{ij} \in \{0, 1\}$ (crisp case). The centroid algorithm amalgamates at each stage a pair of groups whose centroids are closest and the complete link method merges two clusters with the smallest dissimilarity, where the dissimilarity is defined as the largest distance between objects of these two clusters. In Fig.15, the segmenta-

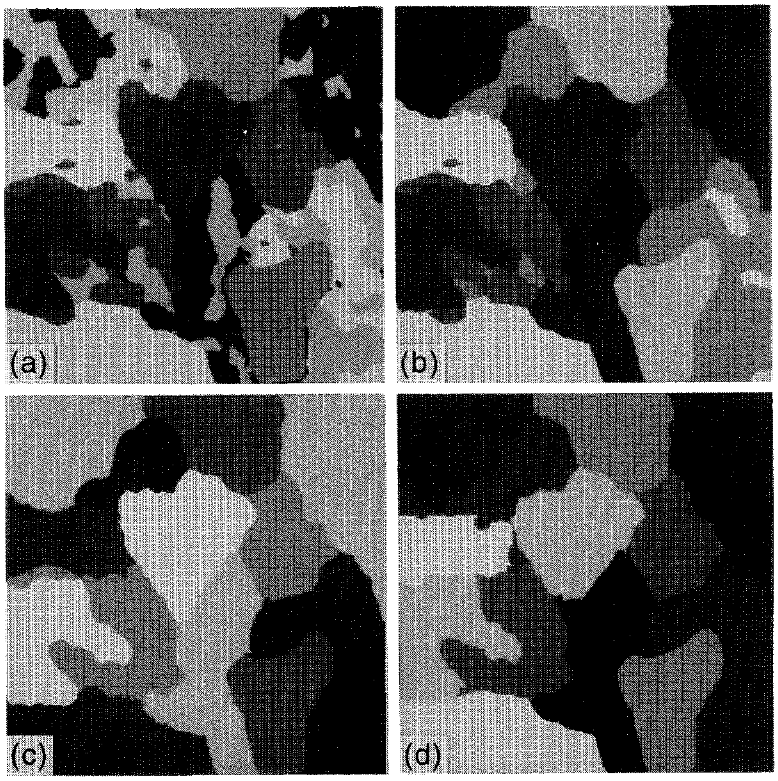


Fig. 14. Segmentation results obtained on "p2" for different number of pyramid levels l_{max}. We used the fuzzy c-means algorithm with $c = 6$. (a) $l_{max} = 3$, (b) $l_{max} = 4$, (c) $l_{max} = 5$, (d) $l_{max} = 6$.

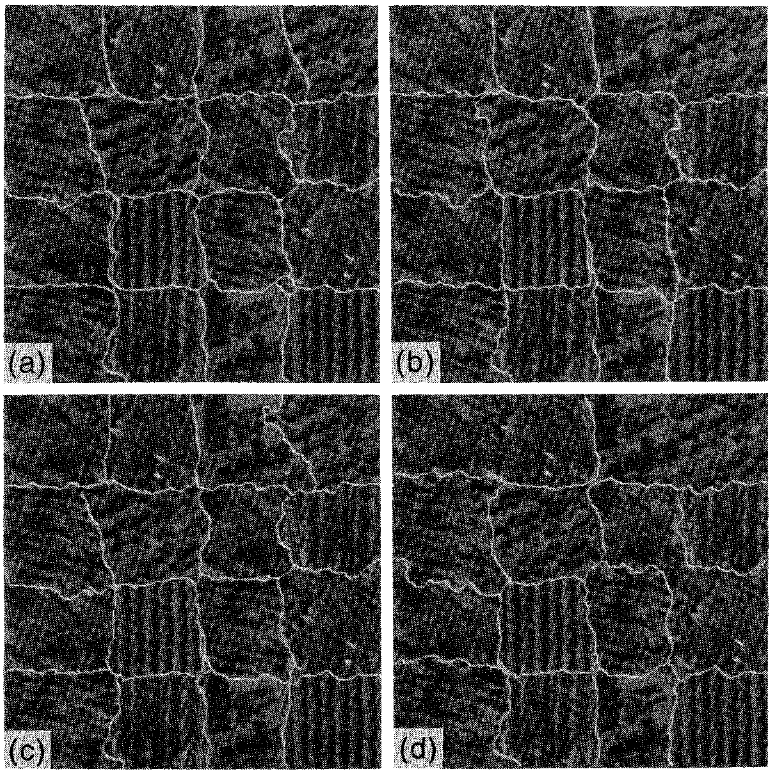


Fig. 15. Segmentation results of "p1" obtained with different clustering algorithms. (a) fuzzy c-means, (b) sum of squares, (c) centroid and (d) complete link (farthest neighbor).

Table 2. Some clustering strategies obtainable from Lance and Williams' general agglomerative algorithm

Name	α_i	β	γ
(1) Complete link	1/2	0	1/2
(2) Centroid	$n_i/(n_i+n_j)$	$-n_in_j/(n_i+n_j)^2$	0
(3) Sum of squares	$(n_i+n_k)/(n_i+n_j+n_k)$	$-n_k/(n_i+n_j+n_k)$	0

n_i Denotes the number of objects in the ith group.

Table 3. Cpu time (s) for "p1" ($c = 10$, $l_{\max} = 5$) and "p2" ($c = 6$, $l_{\max} = 3$) on a Sparc 10 and Silicon Graphics Power Series machines

Image	Machine	Pyramid	Clustering	Restoration	Refinement	Total
"p1"	Sparc 10	1.7	44.0	0.2	37.8	83.7
"p2"		1.5	19.7	0.5	67.4	89.1
"p1"	Silicon Graphics	1.6	39.8	0.5	31.9	73.8
"p2"	Power series	1.5	14.8	0.6	61.5	78.4

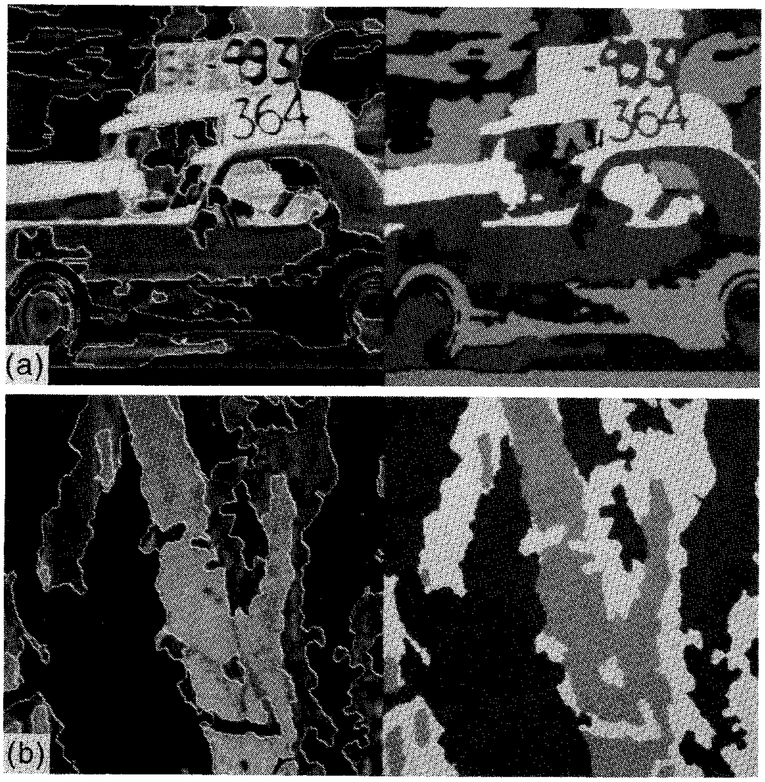


Fig. 16. Segmentation of images based on the gray level information. The segmentation and the class assignments are shown for (a) the vintage cars race and (b) a piece of bark. In the first case, we used the fuzzy c-means algorithm with $c = 4$ and $m = 1.6$ and in the second case with $c = 3$ and $m = 1.6$.

tion results obtained with the different clustering algorithms can be visually compared. The boundaries are very similar. However, the detected number of classes changes. With the fuzzy c-means and centroid algorithms all seven classes are found, while one class (resp. two) is missing when using the sum of squares criterion (resp. complete link). We also tested our algorithm on 1D features, being the gray levels of the original images. Two examples are given in Fig. 16. The first image represents a vintage cars race and the second, a piece of bark. Mostly accurate borders are obtained and even fine details are preserved (the number "364" for example). However, we can notice a few false contours. They can be due to a cross-boundary smoothing while building the pyramid or to low variations of the gray-level gradient.

We can finally mention the cpu time (in seconds) for the segmentation of "p1" ($c = 10$, $l_{\max} = 5$) and "p2" ($c = 6$, $l_{\max} = 3$). This time depends mainly on the number of classes c, on the number of pyramid levels and on the image size at the highest resolution level. In general, it will take longer for higher values of c and lower values of $l_{\max}$ (because the image size at level $l_{\max}$ is larger and there are more boundaries). We used the fuzzy c-means as clustering algorithm and tested our segmentation algorithm on images of size 256 × 256. For lower values of $l_{\max}$, a reduced set of feature vectors is chosen by means of random sampling. The different times are given in Table 3.

CONCLUSIONS

Detecting image boundaries often requires multidimensional segmentation algorithms. The presence of noise in the feature space causes a class-overlap and this can be reduced in a multidimensional pyramid. At the coarsest resolution level, the classes are obtained by means of a clustering algorithm. We proposed an orientation-adaptive boundary refinement process along with a relaxation labeling that allows the use of different clustering procedures without spatial connectivity constraints. Experiments on real images showed that accurate boundaries can be obtained.

Although the problem of knowing *a priori* the exact number of classes is not solved, it has been reduced by over-estimating the number of initial classes and reassigning small and scattered classes. A multi-passes deterministic relaxation algorithm is proposed to reassign the insignificant classes. Experiments showed that similar classifications are obtained for different initial number of classes c. Small differences in the spatial distribution of the labels are further eliminated by the boundary refinement procedure. The way of selecting a clustering criterion is always a problem because the shape of the clusters is usually unknown. Our paradigm is relatively insensitive to this problem and experiments showed that similar segmentation results can be obtained with different clustering algorithms. It seems also that the paradigm.is not sensible to different within-class variances at the level where the clustering algorithm is applied.

In our approach we separated the clustering and the spatial restoration processes into two steps. These two processes could be merged into one step by using the clustering algorithms with spatial connectivity constraints. At the coarsest level of the pyramid, it could be interesting to try, for instance, the MRF k-means segmentation algorithm described by Pappas[43] or some other algorithms using MRF and stochastic relaxation.[22] Due to the small image size at this level, it would take acceptable computational cost. However, we showed that good segmentation results can also be obtained with unconstrained clustering algorithms that are widely addressed in the literature.

A drawback of this algorithm, related to the simple structure of the quadtrees, is its incapacity of keeping track of small regions while building up the pyramid. A tradeoff between the size of the smallest detectable region and the amount of noise reduction has to be found. From the experiments we can define a lower and upper bound for $l_{\max}$. Under the lower bound, the noise is not enough reduced and some extra subregions can be observed. Above the upper bound the classes merge together. In between ($l_{\max} = 5$ and 6), the expected segmentation is obtained. Therefore, it seems that the choice of the number of levels has a certain degree of freedom as more than one possibility produces good segmentation results. Thus, it would be interesting to find a way to compute the "optimal" number of levels. However, the proposed algorithm does not solve the problem of images with regions of all sizes and more complicated data structures such as the linked pyramid[31] should be used.

Acknowledgments—The authors would like to thank Dr Hans Du Buf and Dr Jean-Marc Vesin for helpful comments and for the proof-reading of the manuscript.

REFERENCES

1. R. Haralick and I. Dinstein, Textural features for image classification, *IEEE Trans. Systems, Man Cybernet.* **3**, 610–621 (1973).
2. K. Laws, *Textured Image Segmentation*, Ph.D. thesis, Dept of Elec. Eng., Univ. Southern California, California U.S.A. (1980).
3. M. Unser, Sum and difference histograms for texture classification, *IEEE Trans. Pattern Analysis Mach. Intell.* **8**, 118–125 (1986).
4. J. Bigün, G. Granlund and J. Wiklund, Multidimensional orientation estimation with applications to texture analysis and optical flow, *IEEE Trans. Pattern Analysis Mach. Intell.* **13**, 775–790 (1991).
5. A. Bovik, M. Clark, and W. Geisler, Multichannel texture analysis using localized spatial filters, *IEEE Trans. Pattern Analysis Mach. Intell.* **12**, 55–73 (1990).
6. A. Jain and F. Farrokhnia, Unsupervised texture segmentation using gabor filters, *Pattern Recognition* **24**, 1167–1186 (1991).
7. J. Duvernoy and J. Leger, Karhunen-loeve analysis of multispectral data from landscapes, *Optics Commun.* **32**, 39–43 (1980).
8. G. Nagy and J. Tolaba, Nonsupervised crop classification through airborne multispectral observations, *IBM J. Res. Dev.* 138–153 (March 1972).
9. R. Rosenfeld, R. Hummel and S. Zucker, Scene labeling by relaxation operations, *IEEE Trans. Systems, Man Cybernet.* **6**, 420–433 (1976).
10. E. Riseman and M. Arbib, Survey, computational techniques in the visual segmentation of static scenes, *Comput. Graphics Image Process.* **6**, 221–276 (1977).
11. J. Kittler and J. Foglein, On compatibility and support functions in probabilistic relaxation, *Comput. Vision Pattern Recognition* **34**, 257–267 (1986).
12. B. Everitt, Unresolved problems in cluster analysis, *Biometrics* **35**, 169–181 (1979).
13. J. Broffitt, P. Krishnaiah and L. Kanal, Nonparametric classification, *Handbook of Statistics*, **2**, 139–168 North-Holland (1982).
14. A. Gordon, *Classification.* Chapman and Hall, New York and London (1981).
15. G. Coleman and H. Andrews, Image segmentation by clustering, *Proc. IEEE* **67**, 773–785 (1979).
16. J. Jolion, P. Meer and S. Batauche, Robust clustering with applications in computer vision, *IEEE Trans. Pattern Analysis Mach. Intell.* **13**, 791–802 (1991).
17. R. Webster and P. Burrough, Computer based soil mapping of small areas from samples data. i. multivariate

classification and ordination, *J. Soil Sci.* **23**, 210–221 (1972).

18. R. Haralick and I. Dinstein, A spatial clustering procedure for multi-image data, *IEEE Trans. Circuits Syst.* **22**, 440–450 (1975).
19. J. Gupta and P. Wintz, A boundary finding algorithm and its applications, *Trans. Circuits Syst.* **22**, 351–363 (1975).
20. S. Horowitz and T. Pavlidis, Picture segmentation by a directed split and merge procedure, in *Proc. Int. Join Conf. Pattern Recognition*, pp. 424–433 (1974).
21. R. Wilson and M. Spann, *Image Segmentation and Uncertainty*. Research Studies Press Ltd, Letchworth, England (1988).
22. S. Geman and D. Geman, Stochastic relaxation, gibbs distributions, and the bayesian restoration of images, *IEEE Trans. Pattern Analysis Mach. Intell.* **6**, 721–741 (1984).
23. H. Derin and H. Elliott, Modeling and segmentation of noisy and textured images using gibbs random fields, *IEEE Trans. Pattern Analysis Mach. Intell.* **9**, 39–55 (1987).
24. S. Laskshmanan and H. Derin, Simultaneous parameter estimation and segmentation of gibbs random fields using simulated annealing, *IEEE Trans. Pattern Analysis Mach. Intell.* **11**, 799–813 (1989).
25. M. Spann and R. Wilson, A quadtree approach to image segmentation which combines statistical and spatial information, *Pattern Recognition* **18**, 257–269 (1985).
26. J. Bigün, Frequency and orientation sensitive texture measures using linear symmetry, *Signal Processing* **29**, 1–16 (1992).
27. J. Bigün and J. Du Buf, N-folded symmetries by complex moments in gabor space and their application to unsuperivsed texture segmentation, *IEEE Trans. Pattern Analysis Mach. Intell.* **1**, 80–87 (1994).
28. J. Bigün, Unsupervised feature reduction in image segmentation by local transforms, *Pattern Recognition Lett.* **14**, 573–583 (1993).
29. R. Wilson and G. Granlund, The uncertainty principle in image processing, *IEEE Trans. Pattern Analysis Mach. Intell.* **6**, 758–767 (1984).
30. W. Grosky and R. Jain, Optimal quadtrees for image segments, *IEEE Trans. Pattern Analysis Mach. Intell.* **5**, 77–83 (1983).
31. J. Burt, T. Hong and A. Rosenfeld, Segmentation and estimation of image region properties through cooperative hierarchical computation, *IEEE Trans. Systems, Man Cybernet.* **11**, 802–809 (1981).
32. M. Spann and C. Horne, Image segmentation using dynamic thresholding pyramid, *Pattern Recognition* **22**, 719–732 (1989).
33. K. Florek, Lukasewicz, J. Perkal, H. Steinhaus and S. Zubrzycki, Sur la liaison et la division des points d'un ensemble fini, in *Colloq. Math.* pp. 282–285 (1951).
34. L. Kaufman and P. Rousseeuw, *Finding Groups in Data*. John Wiley and Sons, Chichester, U.K. (1990).
35. J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithm*. Plenum Press, New York and London (1981).
36. H. Romesburg, *Cluster Analysis for Researchers*. Lifetime Learning Publications, Belmont, California (1984).
37. G. Ball and D. Hall, Isodata, a novel method of data analysis and pattern classification, Technical Report NTIS AD699616, Standford Research Institute, Standford, California (1965).
38. P. Schroeter and J. Bigün, Image segmentation by multidimensional clustering and boundary refinement with oriented filters, in *Gretsi Fourteenth symposium*, pp. 663–666, Juan les Pins, France (1993).
39. A. Grace and M. Spann, Edge enhancement and fine feature restoration of segmented objects using pyramid based adaptive filters, in *BMWC*, Guildford (1993).
40. P. Brodatz, *Textures: a Photographic Album for Artists and Designers*, Dover Publications, New York (1966).
41. H. Lance and W. Williams, A general theory of classificatory sorting strategies i hierarchical systems, *Computer J.* **9**, 373–380 (1967).
42. J. Banfield and A. Raftery, Model-based gaussian and non-gaussian clustering, *Biometrics* (accepted) (1993).
43. T. Pappas, An adaptive clustering algorithm for image segmentation, *IEEE Acoustics, Speech Signal Process.* **40**, 901–914 (1992).

About the Author—PHILIPPE SCHROETER was born in Fribourg, Switzerland on October 30th, 1966. He received his diploma in electrical engineering from the Swiss Federal Institute of Technology EPFL, Lausanne, Switzerland, in January 1991. In April 1991, he joined the Signal Processing Laboratory at the EPFL as a Ph.D. student. His research interests include image segmentation, computer vision and motion analysis.

About the Author—JOSEF BIGÜN received his M.Sc. (applied mathematics and physics) and Ph.D. degrees (computer vision) from Linköping University, (Sweden) in 1983 and 1988, respectively. During 1988 he was, as secretary and treasurer, a board member of the Swedish Society for Automated Image Processing. In 1988, he joined the department of electrical engineering at Swiss Federal Institute of Technology in Lausanne (EPFL) where he is conducting research on pattern recognition and computer vision. His general research interests are computer image analysis in its broad sense, and understanding of the biological recognition mechanisms. In particular he is interested in texture, volumetric and motion image analysis.