



# Colour and texture segmentation using wavelet frame analysis, deterministic relaxation, and fast marching algorithms

S. Liapis, E. Sifakis, and G. Tziritas\*

*Department of Computer Science, University of Crete, P.O. Box 2208, Heraklion, Greece*

Received 16 February 2001

---

## Abstract

Luminance, colour, and/or texture features may be used, either alone or in combination, for segmentation. In this paper luminance and colour classes are described using the corresponding empirical probability distributions. For texture analysis and characterisation a multichannel scale/orientation decomposition is performed using wavelet frame analysis. Knowing only the number of the different classes of the image, regions of homogeneous patterns are identified. On these regions the features characterising and describing the different classes are estimated. Two labelling algorithms are proposed. The first, a deterministic relaxation algorithm using a quadratic distance measure, yields the labelling of pixels to the different colour–texture classes. The second is a new Multi-label Fast Marching algorithm utilising a level set boundary determination.

© 2003 Elsevier Inc. All rights reserved.

---

## 1. Introduction

Often in image analysis and interpretation tasks, including multimedia applications and visual inspection, colour textured images must be segmented for recognition purposes. Most approaches to texture analysis proposed during the past decades fall into one of three broad categories: statistical, model-based, and signal

---

\* Corresponding author. Fax: +30-81-39-3501.

*E-mail addresses:* liapis@csd.uoc.gr (S. Liapis), sifakis@csd.uoc.gr (E. Sifakis), tziritas@csd.uoc.gr (G. Tziritas).

processing techniques (Randen and Husoy, 1999; Reed and du Buf, 1993). The statistical approaches, such as (Chen and Pavlidis, 1983), were primarily investigated in the 1980s. Model-based methods include, among others, Markov random field (Kashyap et al., 1982) and simultaneous autoregressive models (Mao and Jain, 1992). Our approach could be identified as a signal processing method, and in particular it is based on a joint spatial/spatial-frequency representation of the texture patterns. A characteristic common to most signal processing methods is that the image is submitted to a filter bank, and then from the filters' outputs some energy measure is computed. In this context the Gabor filter analysis could be used (Bovic et al., 1990; Jain and Farrokhnia, 1991; Porat and Zeevi, 1989; Raghu and Yegnanarayana, 1996). Having similar properties, but a simpler implementation, wavelet transform representations could also be used.

In (Reed and du Buf, 1993) a concise survey of many recent texture segmentation and feature extraction techniques are presented. We focus here on some of these, which are more relevant to our work. Mao and Jain (1992) use a multiresolution simultaneous autoregressive model for extracting texture features. These features are filtered separately and eventually weighted. Then a clustering algorithm determines the features for a known number of texture classes. The MH index, a modification of Hubert's statistic, is used for evaluating the segmentation and for validating the number of clusters. For the same task the ratio of the within-class variance to the total variance is also used. Comer and Delp (1999) use a multiresolution Gaussian autoregressive model for the pyramid representation of the observed image, and assume a Markov random field for the label image. They propose a Multiresolution Maximisation of Posterior Marginals algorithm for solving the labelling problem, while an expectation/maximisation method is used for estimating the model's parameters. A similar model is proposed in (Krishnamachari and Chellappa, 1997), where the *Iterated Conditional Modes* algorithm is used for labelling pixels.

The multi-channel filtering approach is also adopted by many authors for texture analysis and segmentation. Jain and Farrokhnia (1991) used a bank of Gabor filters and each filtered image is subjected to a nonlinear transformation. Using a local energy computation they obtain feature images, and then a square-error clustering algorithm is used to identify the texture classes. A modification of the Hubert's statistic is used as an index to validate the number of texture categories. Raghu et al. (1997) use also a Gabor wavelet filter-bank. A vector quantizer divides the entire feature space to a limited finite number of clusters. Then based on the modelisation of two processes, called partition and competition processes, an energy function is defined and optimised by a deterministic relaxation neural network. The modified Hubert index is again used for validating the number of clusters. Laine and Fan (1996) use a wavelet packet frame decomposition for texture description. Then they obtain a segmentation map relying on 2D envelope detection. For that, two algorithms were proposed: Hilbert transform and zero-crossings localisation.

Many different multi-channel filtering approaches have been compared in (Randen and Husoy, 1999). The global segmentation scheme is similar to that of (Jain and Farrokhnia, 1991), while the final stage is implemented using a classifier on the features images. Among the best filters were the wavelet frame approaches and

the quadrature mirror filter-bank (QMF). In addition, Randen and Husoy (1999) addressed the problem of filter optimisation for best discriminant capability.

In this paper, the problem of texture segmentation is approached with algorithms based on the concept of wavelet frames. The aim of the analysis is to determine characteristics corresponding to each texture type, so that each texture pattern is uniquely defined. Such a distinction takes place in the frequency domain, where the input image is equivalently decomposed to different scale levels using the discrete wavelet frames (DWF). Once these characteristics are deduced, statistical properties are applied to extract the features which are necessary to describe the texture pattern. Although this texture analysis approach has been introduced in the past (Laine and Fan, 1996; Unser, 1995), our scheme differs in the segmentation methodology. The main advantages of the Wavelet frames representation are that it focuses on scale and orientation texture features, it decomposes the image into orthogonal components, and it is translation invariant. The last property is quite desirable for precise boundary localisation in the segmentation problem.

Methods for combining colour and texture for segmentation have been proposed in the past (Belongie et al., 1998), using *HSV* colour space and primitive texture features combined in pointwise expectation maximisation clustering.

In this paper two new approaches for textured colour image segmentation are introduced, the first using a Markov random field model for the label field, and the second using a level set method. In both cases at the first stage of the algorithm the texture and colour feature parameters are determined. For this task an unsupervised method of feature extraction is proposed. The number of different classes present in the image is assumed to be known, and the depth of wavelet analysis is given by the user. However in (Jain and Farrokhnia, 1991) a method is proposed for determining the number of different categories, which could also be useful in our approach. The depth of scale analysis is content dependent, and could be determined using the texture similarity criterion introduced here for reliable classification. The analysis could be stopped when a minimum discrimination distance is achieved for any pair of colour–texture patterns. Knowing the number of classes, blocks with homogeneous colour and texture pattern are extracted and grouped, when they are similar. After the hierarchical clustering of these blocks, the feature vector is estimated for each label. At each image point, distance measures are defined from the different labels using a logarithmic function of the likelihood of the corresponding feature vector. When a Markov random model for the label field is adopted, a deterministic relaxation algorithm is applied. In order to obtain more robust initial measures, the distances of the content vector at each image point from the feature vectors could be smoothed by a median filter. When a level set method is used, a propagation speed is defined for each label at each point according to the a posteriori probability. A new multi-label fast marching algorithm is introduced and applied for obtaining the final segmentation map. The first work on level sets for texture segmentation was presented in (Paragios and Deriche, 1999), where a non-orthogonal filter-bank is used, texture features are user-provided, and the curve propagation is guided by boundary and region properties.

The presented work is organized as follows: in Section 2, the underlying theory of the basic filters, the necessary decomposition by upsampling and the use of DWF, as

applied to 2D signals, are described. The feature extraction algorithm is presented in Section 3. Two segmentation approaches are then introduced: a Markov random field model which leads to the use of the deterministic relaxation algorithm of *Iterated Conditional Modes* (Section 4), and a level set boundary method (Section 5). This last was presented in a general framework of Bayesian level sets in (Sifakis et al., 2002), where the *Multi-label Fast Marching* algorithm was first introduced. Various results are shown on synthetic images containing textures from the Brodatz album (Brodatz, 1966) and from the Vistex database, and on various natural scenes.

## 2. Texture and colour characterisation

### 2.1. Texture analysis

The fundamental tools used for processing the texture images are a filter bank and the concept of wavelet frames. Such an analysis is translation invariant, which is a quite desirable property, in particular for the accurate localisation of region boundaries. In addition, the filter bank decomposes the image into orthogonal components. A low-pass filter  $H(z)$  and its conjugate high-pass filter  $G(z)$  are taken as a pair of prototype filters for generating the whole filter bank by upsampling, so that the whole range of bands is covered. Cubic splines having suitable scaling properties could be used for designing the pair of prototype filters (Unser, 1995). Here the fourth-order binomial filter and its conjugate filter are used,

$$H(z) = \frac{3}{8} + \frac{1}{4}(z + z^{-1}) + \frac{1}{16}(z^2 + z^{-2}), \quad (1)$$

$$G(z) = zH(-z^{-1}), \quad (2)$$

in the frequency domain, and

$$h(n) = \begin{cases} 6/16, & n = 0, \\ 4/16, & |n| = 1, \\ 1/16, & |n| = 2, \\ 0, & |n| > 2, \end{cases} \quad g(n) = (-1)^{1-n}h(1-n) \quad (3)$$

in the spatial domain. The filter bank is obtained iteratively, indexed by the scale factor  $i = 0, \dots, I$ ,

$$H_{i+1}(z) = H(z^{2^i})H_i(z), \quad (4)$$

$$G_{i+1}(z) = G(z^{2^i})H_i(z), \quad (5)$$

where  $H_0(z) = 1$ .

Therefore, the input signal can be decomposed into wavelet coefficients corresponding to different scales. Thus the DWF can handle important texture characteristics, such as periodicity and translational invariance. The resulting representation is overcomplete because the filtered images are not subsampled. This approach places

fewer restrictions on the choice of the prototype filters, and furthermore generates orthonormal basis functions, a very important feature for texture classification. The above low-pass filter is zero-phase, guaranting good localisation in boundary determination. It also ensures a good orthogonalisation, as it can be measured by the correlation coefficient between the two output signals, as follows:

$$\gamma_{\text{LH}}(m) = \frac{\sum_k \sum_l h(k)g(l)\gamma_s(m+k-l)}{\sqrt{\sum_k \sum_l h(k)h(l)\gamma_s(k-l)}\sqrt{\sum_k \sum_l g(k)g(l)\gamma_s(k-l)}}, \quad (6)$$

where  $\gamma_s(\cdot)$  is the autocovariance function of the input signal. For a fully correlated input signal,  $\gamma_{\text{LH}}(m)$  is zero-valued for any  $m$ , because the sum of all  $g(\cdot)$  coefficients is zero. An interesting and important indication of the output correlation is given by the maximum value of  $\gamma_{\text{LH}}(m)$  under the assumption of a first-order Markov process for the input signal. As at each layer the input signal processed is an approximation signal with strong correlation, we provide here the maximum  $\gamma_{\text{LH}}(m)$  value for  $\rho = 0.9$ ,

$$\max_m \gamma_{\text{LH}}(m) = 0.027,$$

which is sufficiently small for considering the interband decorrelation hypothesis to be valid in practice. Other filter choices are possible, and a study of the best filter bank for separating specific texture classes is open for future investigations.

All of the above may be extended into 2D, thus becoming applicable to textured images whose features must be extracted. This can be accomplished by taking as wavelet bases the cross product of separable bases in each direction. Thus, the analysis is computationally less complicated, since rows and columns of the image are processed separately, as if they were 1D signals. Finally a set of  $N$  components is obtained,  $\{y_i(m, n) : i = 1, \dots, N\}$ , with  $N = 3I + 1$ , where  $I$  is the number of analysis levels. The  $N$ th component is the approximation at the  $I$ th scale level, while the other  $N - 1$  are the details in the different orientations and scales.

Of course, other filter prototypes could be used in the same processing scheme, including non-dyadic decompositions. In addition to wavelet frames decomposition, Gabor analysis could also be used in the following sections considering the segmentation problem. We preferred the Discrete Wavelet Frames analysis because of the separability, which makes it simpler.

## 2.2. Colour description

*Lab* colour space, designed to be perceptually uniform, was used for colour feature extraction. Because the luminance component  $L$  is used in texture analysis and is contained in the approximation component, only the chromaticity components  $(a, b)$  are used. In our work the local 2D histograms of the  $(a, b)$  components were used as features. When some model of the distribution of the  $(a, b)$  histograms is fit (e.g., Gauss or Laplace) the parameters of the model are used as the features. Often though no such modeling is feasible, in which case local histogram estimation is required, making the procedure time consuming. The histograms are smoothed with a Gauss kernel to improve statistical robustness.

### 2.3. Dissimilarity measure

In the feature extraction procedure and for some clustering operations dissimilarity measures are needed. The Bhattacharya distance was proven suitable for the tasks of feature extraction in our work. The Bhattacharya distance is strongly linked to the minimum classification error for the two-classes case. It is in fact a special case of the Chernoff bound for the Bayes error (Young and Fu, 1986) and it is well known that the Chernoff information gives the highest achievable exponent for the error probability. The definition of the Bhattacharya distance is

$$d_B(p_1, p_2) = -\ln \left( \int_x \sqrt{p_1(x)p_2(x)} dx \right), \quad (7)$$

where  $p_1, p_2$  are probability density functions of vector  $x$  of any dimension under two hypotheses indexed by 1 or 2. The Bhattacharya distance satisfies the symmetric property,

$$d_B(p_1, p_2) = d_B(p_2, p_1),$$

while the triangle property is satisfied only in specific configurations.

In our case this distance could be defined on empirical probability distributions. The discrete expression is

$$d_B(h_1, h_2) = -\ln \left( \sum_i \sqrt{h_1(i)h_2(i)} \right), \quad (8)$$

where  $i$  is an index into the bins of the normalized histograms  $h_1, h_2$ .

In the case that we have a model for the feature's distribution, a simpler expression for the Bhattacharya distance can be deduced. Statistical analysis of experimental results have shown that the probability distribution of the high-frequency components could be the generalised Gaussian (Mallat, 1989)

$$p(x) = \frac{c}{2\sigma\Gamma(1/c)} e^{-(|x|/\sigma)^c}, \quad (9)$$

where the parameter  $\sigma$  is related to the variance and  $c$  reflects the sharpness of the probability density function. For  $c = 2$ , we obtain the Gaussian function, and for  $c = 1$ , the Laplacian function. Taking into account that we have in practice an interband decorrelation, the Bhattacharya distance will be the sum of the corresponding distances on all the components

$$d_{1,2}^B(y) = \frac{1}{c} \sum_{i=1}^N \ln \frac{\sigma_{i,1}^c + \sigma_{i,2}^c}{2\sqrt{\sigma_{i,1}^c \sigma_{i,2}^c}}, \quad (10)$$

where  $N$  is the dimension of the feature vector and  $\sigma_{i,1}^2$  and  $\sigma_{i,2}^2$  the feature variances.

When texture features and colour histogram features need to be combined, the last model-based expression is used for texture features and the initial expression is used for histograms. Each term for each feature is summed independently of the expression of the distance, because the decomposition is approximately orthogonal.

### 3. Feature extraction

In this work, only the number of different colour–texture classes is assumed to be known. The segmentation can be based on any combination of luminance, colour, and texture, either fixed or user-selected. A method for determining the number of texture categories is proposed in (Jain and Farrokhnia, 1991). The empirical probability distributions for luminance and colour, and the variances of the texture components are estimated using a learning process, which is described below. The mean value of the approximation component is also estimated and taken into consideration, if it is sufficiently discriminating. The option of two different approaches is allowed. Fig. 1 illustrates the whole procedure for texture segmentation, and shows at which point the two approaches diverge. The feature parameter estimation stage is the same for both. The determination of the pixel’s label differs, even though it is based, in both cases, on the a posteriori probability of the label. The first one adopts a Markov random field model for the labels, while the other assumes that the region boundaries are suitably defined level sets. In the following paragraphs the various feature extraction modules are described in detail, while the two segmentation algorithms are treated in the two following sections.

#### 3.1. Rejection of heterogeneous blocks

Crucial to the segmentation process is the identification of image regions containing the best representative colour–texture characteristics for all the classes. For this

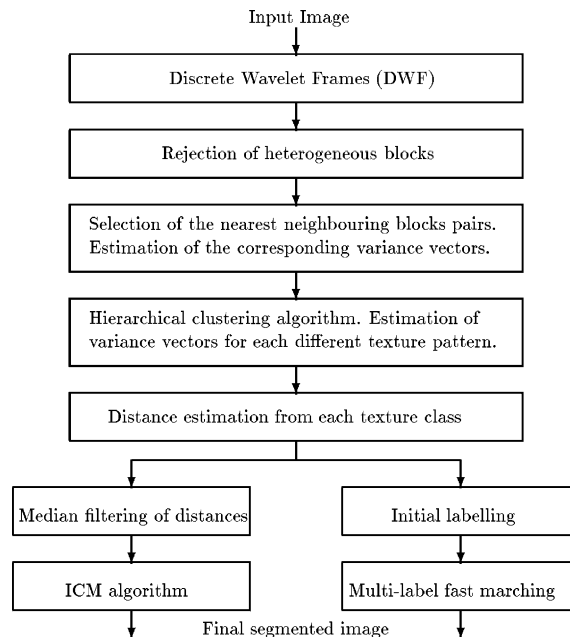


Fig. 1. Texture feature extraction and segmentation algorithms.

purpose, the image is divided into blocks. The block size should be large enough to be statistically reliable, but not so large as to prohibit the segmentation of small regions. Typically a block should be about 1–2% of the image, that is for a  $256 \times 256$  image a suitable block size would be  $32 \times 32$ , giving 64 sub-images. It is evident that blocks containing two or more different classes (heterogeneous blocks) should be not used for estimating the characteristics of any class. If a block is homogeneous, with high probability it will be similar to the majority of its neighbouring blocks. The similarity is measured by the Bhattacharya distance defined in the previous section. As the absolute distance depends on the texture pattern it will be more efficient to compare the median dissimilarity to the minimal one, thus defining an homogeneity criterion

$$M(B_0) = \text{med}_n d_B(B_0, B_n) - \min_n d_B(B_0, B_n), \quad (11)$$

where  $B_0$  is the block under examination and the median and the minimum are taken over its eight neighbouring blocks  $B_n$ . The next step is to find a threshold for distinguishing the homogeneous blocks from the heterogeneous ones. We can assume a bimodal distribution for  $M(B_0)$  corresponding to the two categories of homogeneous and heterogeneous blocks. The problem is then to identify the breaking point between the two categories. We sort the blocks according to the homogeneity criterion  $M$ , expecting an abrupt change between the two categories. As an indicator of the change we use a measure deduced from the numerical approximation of the second derivative as calculated from the sorted data

$$M(B_{n+1}) - 2M(B_n) + M(B_{n-1}).$$

We take the maximum of the indicator, in the interval between 10 and 50% of the blocks, as the breaking point. Fig. 2 illustrates the thresholding procedure applied to a textured image. On the left is the input mosaic textured image; on the right are blocks detected as heterogeneous, shown in white. The threshold is automatically set as shown on the plot of the sorted homogeneity criterion. On the vertical axis the homogeneity measure is given, while on the horizontal axis the index of the block, according to the order in the sorted list is given.

### 3.2. Hierarchical clustering for feature estimation

The procedure described in the previous subsection rejects all heterogeneous blocks. Following this, to obtain the best representative blocks from the remaining homogeneous ones, the homogeneous blocks are sequentially grouped into pairs

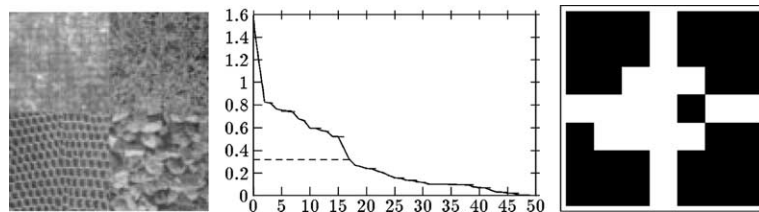


Fig. 2. Rejection of heterogeneous blocks.



using again the Bhattacharya distance and starting by the closest ones. A pair is formed only if the blocks are neighbouring and their distance is smaller than a threshold set according to the homogeneity screening explained in the previous paragraph

$$T_p = \frac{\max_{B_0 \in B_M} \text{med}_n d_B(B_0, B_n) + \min_{B_0 \in B_M} \text{med}_n d_B(B_0, B_n)}{2}, \quad (12)$$

where the *maximum* and the *minimum* are taken over all homogeneous blocks  $B_0$  retained in the previous phase ( $B_M$  being the set of homogeneous blocks).

Then, the variance vectors for texture and colour distributions are estimated for each pair of blocks thus created. In order to estimate the features of the different classes in the image, a hierarchical clustering algorithm (Duda and Hart, 1973) is applied to the features of the grouped pairs. This algorithm, in each step, merges the two nearest clusters by estimating the new features from the corresponding blocks. Each step thus reduces the number of clusters by one. The procedure terminates when the number of clusters becomes equal to the number of the different classes in the image to be segmented. The texture parameters are estimated at the end of this procedure, as well as the colour histograms. Results illustrating the effectiveness of the estimation algorithm are given for the synthetic image of Fig. 2 containing four different textures (top-left: D19, top-right: D9, bottom-left: D3, and bottom-right: D5), which was analysed to four scale levels yielding 12 detail coefficients. We present in Fig. 3 the blocks retained for the feature vectors estimation (left-hand image), as well as the corresponding result for another mosaic image containing five different texture patterns (Fig. 7a). The estimated variance vectors of each texture pattern are given, both for the supervised (Table 1) and unsupervised (Table 2) procedures. The unsupervised procedure uses the automatic clustering technique described above, while the supervised procedure estimates the features from the known texture patterns.

### 3.3. Distance estimation and filtering

Taking into consideration all parameters characterising a colour/texture pattern, a point belongs to a given class if its distance from the class is minimal. The distance is

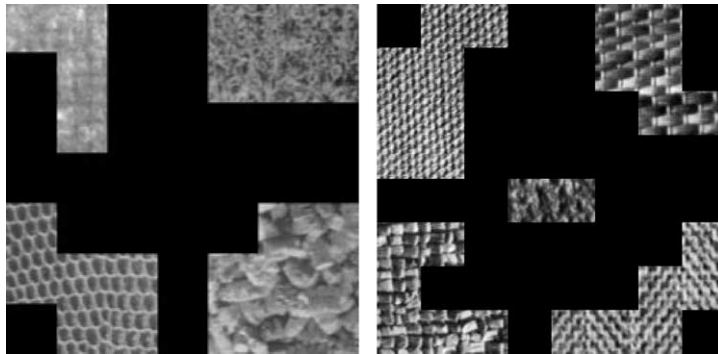


Fig. 3. Blocks automatically retained for feature vectors estimation for two mosaic textured images.

Table 1  
Variance vectors for texture pattern supervised feature extraction

Subband	D19	D9	D3	D5
3H	5.96	2.03	2.94	0.95
3V	0.08	0.37	0.96	0.12
3D	0.03	0.06	0.07	0.05
2H	0.89	3.42	6.00	2.29
2V	0.61	1.86	1.98	0.39
2D	0.03	0.05	0.05	0.02
1H	1.24	3.62	7.70	4.62
1V	1.46	3.12	7.60	1.62
1D	0.06	0.19	0.17	0.10
0H	2.61	2.24	12.43	8.47
0V	1.78	2.33	3.30	2.68
0D	0.07	0.19	0.44	0.22

Table 2  
Variance vectors for texture pattern by unsupervised feature extraction

Subband	D19	D9	D3	D5
3H	6.13	1.75	2.92	0.90
3V	0.08	0.41	0.41	0.14
3D	0.02	0.05	0.06	0.04
2H	0.91	3.42	5.84	2.23
2V	0.61	1.77	1.76	0.34
2D	0.02	0.05	0.04	0.01
1H	1.32	3.42	8.19	4.30
1V	1.47	3.26	7.31	1.26
1D	0.06	0.20	0.16	0.10
0H	3.13	2.01	13.48	9.20
0V	1.76	2.42	2.61	2.67
0D	0.07	0.19	0.42	0.23

defined using the likelihood function of the class label. The more probable a class is, the less the distance is for a given point, according to the inverse of the logarithm of the likelihood function. Assuming that the probability density function of the texture features is Gaussian, the distance of a point  $(m, n)$  represented by the vector  $\mathbf{y}(m, n) = [y_1(m, n) \dots y_N(m, n)]^T$  from a class with variances  $\sigma_{i,j}^2$  and mean value  $\mu_j$  is determined as follows:

$$d_j(\mathbf{y}(m, n)) = \sum_{i=1}^{N-1} \left( \frac{y_i^2(m, n)}{\sigma_{i,j}^2} + \log \sigma_{i,j}^2 \right) + \frac{(y_N(m, n) - \mu_j)^2}{\sigma_{N,j}^2} + \log \sigma_{N,j}^2. \quad (13)$$

Indeed, all the components of vector  $\mathbf{y}(m, n)$  as used in the above equation are zero-mean independently of their class. The discrimination results from the difference in the variance of the components. If another assumption is adopted, e.g., the generalized Gaussian, the above equation must be suitably adapted. If the probability density function of the colour features is also Gaussian, the two corresponding

components are added in Eq. (13). Otherwise, the empirical probability distribution is used for the colour features

$$d_j(y_C(m, n)) = -\ln p_j(y_C(m, n)), \quad (14)$$

where  $y_C(m, n)$  is the vector of  $(a, b)$  colour components. Following these assignments, due to statistical errors on the distance measure, a 2D median filter (typically of size  $9 \times 9$ ) could be applied to each distance array of pixels from each class, as a pointwise distance measure is highly noisy. Averaging the distances provides a more reliable distance measure, and in the presence of region boundaries the median filter should be preferred over a linear filter. The median filter is applied separately for each class (index  $j$ ) and yields smoothed distance arrays, thus compensating for statistical errors.

#### 4. Pixel labelling using deterministic relaxation

Minimising the distance of an image point from the candidate texture classes gives the most probable texture for the given point. As the point classification could give many erroneous classifications, a relaxation algorithm for pixel labelling may be used. A Bayesian approach is adopted based on a Markov random field (MRF) model of the colour/texture labels.

Indeed, many problems in image analysis can be formulated as a scene labelling with contextual information. In such a statistical framework, there are

- a set of sites  $S = \{(m, n)\}$ ;
- a set of possible labels for each site  $L = \{l_1, l_2, \dots, l_q\}$ ;
- a set of observations  $Y = \{y(m, n)\}$ , associated with  $S$ ;
- a neighbourhood relation,  $G$ , over the sites.

The problem is to assign to each site a label in such a way that the solution is consistent with the constraints and the available observations set.

Let  $c$  denote a clique of the neighbourhood  $G$ , that is, a set of sites in which all pairs are mutual neighbours. In Fig. 4 is shown an 8-connected neighbourhood and the retained cliques. A global discrete labelling  $\omega$  assigns one label  $\omega(m, n) \in L$  at each site. A Gibbs distribution (Geman and Geman, 1984), is defined by the so-called clique potentials  $V_c(\omega)$ , for every possible  $c$  and every possible labelling. Following the *Hammersley–Clifford* theorem and the equivalence between MRF

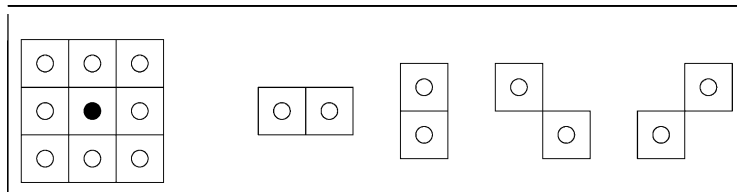


Fig. 4. Second-order neighbourhood: a possible choice of effective cliques.

and Gibbs distributions (Besag, 1974), the probability of a global labelling  $\omega$  is given by the following formula (Geman and Geman, 1984):

$$P(\omega) \triangleq \frac{1}{Z} e^{-(1/T)U_1(\omega)}, \quad (15)$$

where  $T$  is a regularising constant, and

$$U_1(\omega) \triangleq \sum_{c \in C} V_c(\omega), \quad Z = \sum_{\omega} e^{-(1/T)U_1(\omega)}. \quad (16)$$

In the above formula  $C$  denotes the set of totally connected cliques with respect to the neighbourhood definition  $G$  and  $Z$  is a normalising constant, called partition function. In statistical terms,  $U_1(\cdot)$  is the *energy (cost)* function of the system, while  $V_c(\cdot)$  is called potential function and corresponds to the contribution of the local interactions to the global energy. A very crucial issue in this process is to incorporate the prior knowledge with the available observations, in order to create a new form for the energy function. This form is a combination between the expected spatial properties (*homogeneity*) of the label field and adequacy between observations and labelling decisions. Under the above hypotheses the joint probability density function of the label field and the field of texture features is proportional to the following:

$$P(\omega) \prod_{(m,n) \in S} e^{-\delta(y(m,n); \omega(m,n))},$$

where the adequacy of a feature vector to a texture class is given in Eq. (13). The maximisation of the a posteriori probability is equivalent to the maximisation of this joint probability density function, and it leads to the minimisation of the energy function given by

$$U(\omega, y) \triangleq \frac{1}{T} \sum_{c \in C} V_c(\omega) + \sum_{(m,n) \in S} \delta(y(m,n); \omega(m,n)) \triangleq U_1(\omega) + U_2(\omega, y), \quad (17)$$

where the term  $\delta(\cdot)$  expresses this adequacy demand.

For the labelling according to the colour/texture pattern, following the signal decomposition presented above, the adequacy between observations and labelling decisions can be expressed using the distances of Eq. (13)

$$U_2(\omega, y) = \sum_{(m,n) \in S} d_j(y(m,n)), \quad \text{where } \omega(m,n) = l_j. \quad (18)$$

The maximisation of the *maximum* a posteriori probability (MAP), or equivalently the minimisation of the cost function may be performed using either stochastic relaxation algorithms (Geman and Geman, 1984), or deterministic relaxation algorithms as the *Highest Confidence First* (HCF) algorithm (Chou and Brown, 1990), or the *Iterated Conditional Modes* (ICM) algorithm (Besag, 1986). In our work we have used the ICM algorithm.

We have applied our method to three synthetic images containing two, four, and five different textures, all derived from the Brodatz Album (Brodatz, 1966). The first image to be segmented (composed of textures D3 and D9) and the result of the

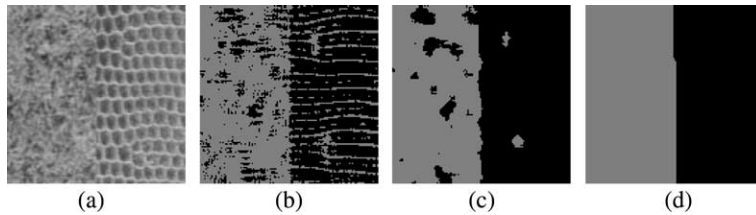


Fig. 5. Segmentation for the synthetic image (a) which contains two textures D3 and D9. (b) Labelling with distance only. (c) Assignments after smoothing with median filter. (d) Final segmented image after applying the ICM algorithm.

segmentation are shown in Fig. 5. The classification error percentage is 18.2% based on likelihood only, 6.5% after median filtering, and 0.6% after ICM relaxation. The above algorithm was applied to a synthetic image containing four different textures (Fig. 6a). Fig. 6b illustrates the segmented image with assignments to labels deduced using only the distance measure in Eq. (13) with 36.0% classification errors. In Fig. 6c, the label assignments resulting from distance smoothing using the median filter are shown reducing the classification errors to 11.4%. The final segmented image after applying the ICM algorithm is presented in Fig. 6d, with 4.8% misclassified pixels. The initial image was analysed to four frequency levels yielding 12 detail coefficients. An additional example was considered, where the segmentation algorithm was applied to the image illustrated in Fig. 7a which contains five different texture types. As in the previous example, the label assignments based on the distance measure are shown in Fig. 7b (35.2% classification errors), the result after smoothing of the distance values is presented in Fig. 7c (5.5% classification error) and the result following application of the ICM algorithm is illustrated in Fig. 7d, with 2.9% misclassified pixels. The decomposed frequency layers in this example were 3, thus producing nine detail coefficients.

Results of this algorithm are shown in Fig. 8 where the *SeaStones* image is segmented in three classes using three levels in DWF decomposition and colour modeled as a Gaussian distribution, yielding 11 distinct features. Substantial improvement was observed when colour features were added, since the used model

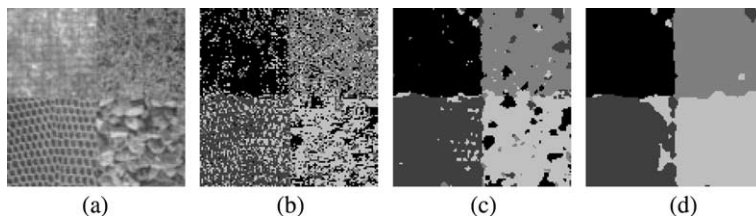


Fig. 6. (a) Initial synthetic image composed of D19, D9, D3, and D5 of the Brodatz album. (b) Labelling with distances only. (c) Assignments after smoothing with median filter. (d) Final segmented image after applying the ICM algorithm.

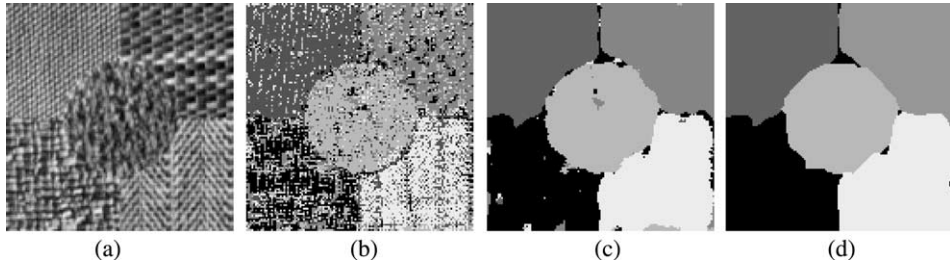


Fig. 7. (a) Initial synthetic image composed of D77, D55, D84, D17, and D24 of the Brodatz album. (b) Labelling with distances only. (c) Assignments after smoothing with median filter. (d) Final segmented image after applying ICM.

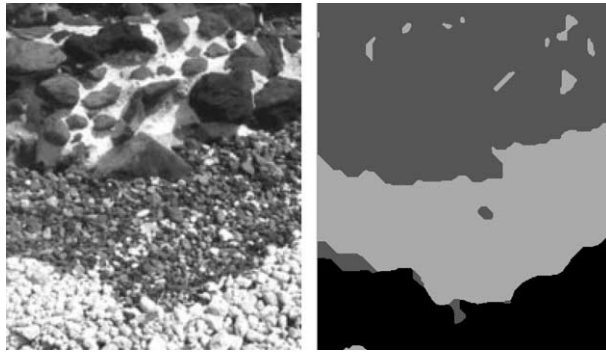


Fig. 8. Segmentation of the *SeaStones* image using the ICM algorithm.

is an acceptable approximation of the actual data distribution. The inherent difficulty in segmenting the current image should be noted, which is augmented by the heterogeneity of the upper class which occurs in both texture and colour.

## 5. Pixel labelling using level sets

We now introduce a new labelling algorithm based on level sets and on the fast marching algorithm. In a subsequent paragraph we describe how to use the new algorithm for labelling according to the texture features, and their distances from the estimated class features.

### 5.1. Multi-label Fast Marching Algorithm

The fast marching level-set algorithm introduced by Sethian (1996) computes a constructive solution to the stationary level set equation

$$|\nabla T(x, y)| = \frac{1}{v(x, y)}, \quad (19)$$

where  $v(x, y)$  corresponds to the velocity of the moving front, while  $T(x, y)$  is a map of crossing times. At a given time the location of the evolving active contour is determined. The resulting segmentation is interpreted by means of the velocity field used. Given the limitation of a constantly positive velocity function and a suitable discrete gradient definition, the fast marching algorithm can construct a solution  $T(x, y)$  which satisfies Eq. (19) over the entire image without resorting to iterative methods. The running time of the fast marching algorithm is of order  $n \log n$  over the image size, classifying it as a very efficient segmentation technique. A thorough presentation of the level set method is given in (Sethian, 1996), and a review of the fast marching algorithm appears in (Sethian, 1999).

The proposed algorithm is a variant of the fast marching algorithm, which, in addition to maintaining the properties of the original, is able to cope with multiple classes (or labels). Execution time is effectively made independent of the number of such classes by handling all the propagations simultaneously and dynamically limiting the range of action for each label to the continuously shrinking set of pixels for which a final decision has not been made. The propagation velocity may also be formulated differently for each class.

The algorithm described below assumes the existence of an initialisation for  $T(x, y)$ , specifically its zero level set. There are three possible states for each pixel. An “alive” pixel represents a fixed arrival time value, A “trial” pixel constitutes a candidacy for a specific label with an arrival time value subject to change. “Far away” pixels have not yet been processed. When no more trial pixels remain the alive pixel with the smallest arrival time is used to label each pixel. The result of the algorithm is not only the resulting  $T(x, y)$  but the corresponding labelling as well. The symbolic description of the algorithm follows:

```

InitTValueMap()
InitTrialLists()
while(ExistTrialPixels()) {
    pxl = FindLeastTValue()
    MarkPixel Alive(pxl)
    UpdateLabelMap(pxl)
    AddNeighborsToTrialLists(pxl)
    UpdateNeighborTValues(pxl)
}

```

The algorithm is supplied with a label map partially filled with decisions. The arrival time for the pixels containing decisions is set to zero while for all other pixels it is set to infinity. A map with pointers to linked lists of trial pixels is also maintained. Those lists are initially empty except for the sites being neighbours to initial decisions. For those sites a trial pixel is added to the corresponding list for each different label of neighbouring decisions and an initial arrival time is assigned. All trial pixels are held in a common priority queue, based on their arrival time.

Until no more trial pixels exist, the trial pixel with the smallest arrival time is selected and turned alive. If there is no other alive pixel for this site, its label is copied

to the final label map. For all non-alive neighbours of this site a trial pixel of the same label is added, if it does not already exist, in the corresponding trial lists. Finally all neighbouring trial pixels of the same label update their arrival times according to the new data.

Although it might seem that trial pixels can exist per site for all different labels, in fact these can be no more than four since a trial pixel is only introduced by a finalised neighbouring decision. In practice trial pixels of different labels coexist only in region boundaries, giving an average number of label candidacies per pixel of at most two. Even in the worst case it is evident that the time and space complexity of the algorithm is independent of the number of different labels. Experiments have illustrated a running time no more than twice the time required by the single contour fast marching algorithm.

## 5.2. Label initialisation and propagation

An initial map of labelled sites is obtained using statistical tests which classify points with high confidence. The probability of classification error is set to a small value. At first all pixels are classified according to their distance from the different labels. The distance of a point from a label is measured using the Bhattacharya distance of the distribution of the data of a block centered at the given point and the features of the label. Then a square block of dimension  $(2w + 1) \times (2w + 1)$  centered again at the given point is considered. The mean distance is used for classifying it to one of the possible labels. This classification gives an estimate of the number of pixels of the sets for all labels. Decision with highest confidence provide the initial sets of labelled points. A small percentage is generally sufficient. The confidence criterion results from the comparison of the distance of the candidate label to the distance of the best label among all the others

$$\sum_{i=-w}^w \sum_{k=-w}^w \min_{l(m+i, n+k) \neq j} d_{l(m+i, n+k)}(m+i, n+k) - d_j(m+i, n+k),$$

where the *minimum* is taken among the labels  $l(m+i, n+k) \neq j$ , and the confidence is higher if the above quantity is greater. The multi-label fast marching level set algorithm, presented in the previous section, is then applied for all sets of points initially labelled. The contour of each region propagates according to a motion field which depends on the label and on the distance of the considered point from the candidate label. The exact propagation velocity for a given label is

$$\begin{aligned} v_j(m, n) &= \Pr(j | \mathbf{y}(m, n)) = \frac{p(\mathbf{y}(m, n) | j) \Pr(j)}{\sum_{l=1}^L p(\mathbf{y}(m, n) | l) \Pr(l)} \\ &= \frac{\Pr(j)}{\sum_{l=1}^L \Pr(l) e^{d_j(\mathbf{y}(m, n)) - d_l(\mathbf{y}(m, n))}}. \end{aligned} \quad (20)$$

The expression of the propagation speed is motivated by the maximum a posteriori probability criterion. The candidate label is propagated according to the a posteriori probability, which is expressed using the likelihood function of each label. Indeed, a



label should be propagated faster towards pixels which could be classified to the candidate label with high probability. The propagation should be rather stopped in case of low probability.

We use the fast marching algorithm for advancing the contours towards the unlabelled space. The dependence of the propagation speed only on the pixel properties, and not on contour curvature measures, is not a strong disadvantage here.

### 5.3. Experimental results

In Fig. 9 the segmentation result for the *zebra* image is presented. Only the luminance histograms are used for the segmentation. In Fig. 10 the segmentation result on the *GrassPlants* natural scene (MIT Media Lab Vistex data set) is shown, where either colour (Fig. 10b) or texture (Fig. 10c) alone is used. In the case of colour, only the histograms of the chromaticity components ( $a, b$ ) are used. For texture-based segmentation the variances of the DWF analysis are used. In Fig. 11 the segmentation result on the natural scene of a *Corridor* (MIT Media Lab Vistex data set) is presented, where only the chromaticity histogram is used. In Fig. 12 the segmentation result on the natural scene of a *City* (MIT Media Lab Vistex data set) is shown, where the histograms of intensity and chromaticity are used.



Fig. 9. Segmentation of the *zebra* image: the original image, the initial labelling, and the final segmentation map.

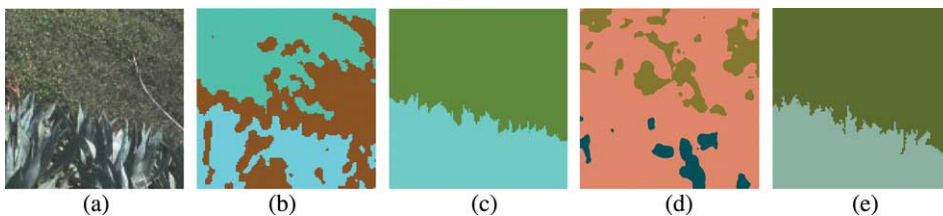


Fig. 10. Segmentation of *GrassPlants*: (a) original image, (b) initial sets for colour-based segmentation, (c) final colour-based segmented image, (d) initial sets for initial sets for segmentation, and (e) final texture-based segmented image.



Fig. 11. Segmentation result on the *Corridor* image: the original image, the initial labelling, and the final segmentation map.



Fig. 12. Segmentation result on the *City* image: the original image, the initial labelling, and the final segmentation map.

In Fig. 13 the result of segmentation is shown on two-texture images presented in (Randen and Husoy, 1999). The percentage of classification errors is 1.3% for the *D004D084* image, and 1.0% for the *D005D092* image. The respective errors

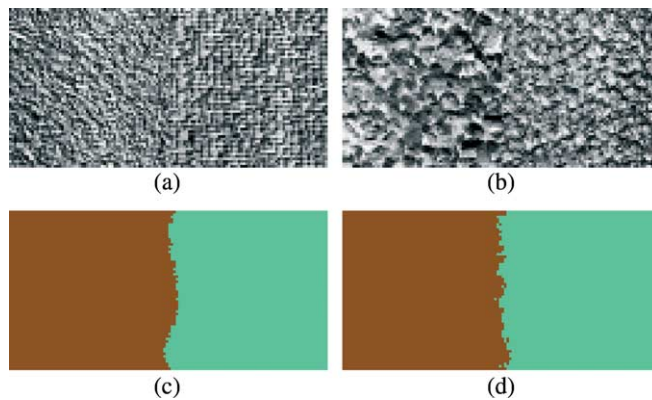


Fig. 13. Segmentation of two-texture images: (a) *D004D084* from Brodatz album, (b) *D005D092* from Brodatz album, (c) segmentation of *D004D084*, and (d) segmentation of *D005D092*.

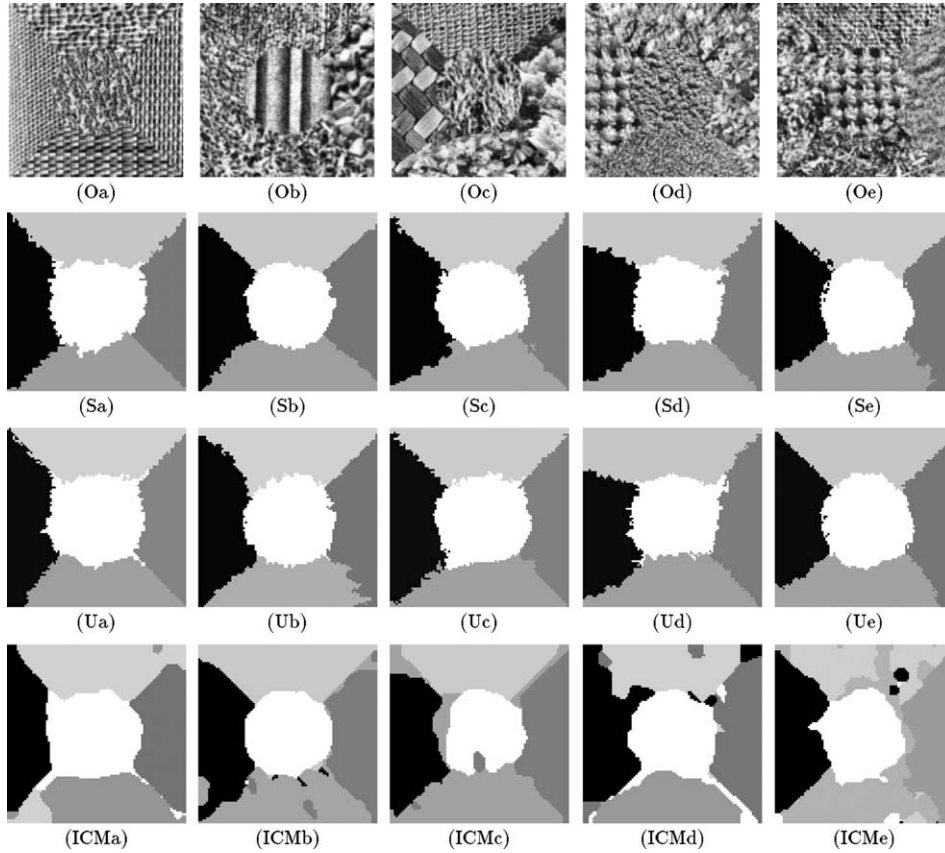


Fig. 14. Segmentation of five-texture images: (Ox) original image, (Sx) segmentation using the multi-label fast marching algorithm with supervised feature extraction, (Ux) segmentation using the multi-label fast marching algorithm with unsupervised feature extraction, and (ICMx) segmentation using the ICM algorithm with unsupervised feature extraction.

for the best filter reported in (Randen and Husoy, 1999) are: 1.9 and 2.5%. In Fig. 14 the result of segmentation is shown on the five-texture images presented in (Randen and Husoy, 1999). In Table 3 we give the classification error results of our methods compared to the results reported in (Randen and Husoy, 1999) for the best filter. We give results for both the supervised (SMLFM) and the unsupervised (UMLFM) feature extraction algorithms, and for the ICM algorithm described in the previous section. In the case of supervised feature extraction we use the known segmentation map for estimating the unknown parameters. In the cases of unsupervised feature extraction and of the ICM algorithm we consider the number of labels to be known and we apply the method of Section 3 for estimating the parameters. In (Randen and Husoy, 1999) the features of the classes have not been extracted from the image to be classified, but from disjoint patches of the same texture.

Table 3  
Comparison of the classification errors.

Image	SMLFM	UMLFM	ICM	Randen and Husoy (1999)
(Oa)	5.8%	5.7%	9.4%	8.2%
(Ob)	4.9%	5.4%	6.5%	17.2%
(Oc)	5.4%	5.5%	9.3%	18.9%
(Od)	8.8%	9.1%	10.6%	22.6%
(Oe)	8.3%	5.7%	13.7%	18.4%

In Fig. 15 the result of segmentation is shown on the 10-texture images presented in (Randen and Husoy, 1999). Here the feature extraction procedure was supervised. The classification error is 6.5% for (Oa) and 5.6% for (Ob). The respective errors given in (Randen and Husoy, 1999) using the best filter (QMF) are 32.3% for (Oa) and 28.5% for (Ob).

In Fig. 16 the result of segmentation is shown on the 16-texture images presented (Randen and Husoy, 1999). Again the feature extraction procedure was supervised. The classification error is 8.3% for (Oa) and 10% for (Ob). The respective errors given in (Randen and Husoy, 1999) using the best filter (QMF) are 36.4% for (Oa) and 41.7% for (Ob). Fig. 17 illustrates the segmentation results of the multi-label fast marching algorithm on the same set of synthetic images presented in the previous section. For the first mosaic with the two Brodatz textures the percentage classification error is 0.7%. For the mosaic image with four (resp. five) Brodatz textures the percentage classification error is about 2.8% (resp. 2.5%).

In Fig. 18 results of combined use of both texture and colour in the *SeaStones* images (Fig. 8) are presented. In Figs. 19–23 we show results on natural scenes from the Corel Photo Gallery. In general, a three-level wavelet frame analysis was used, luminance was quantized into 32 values and chrominance was quantized into 64 values.

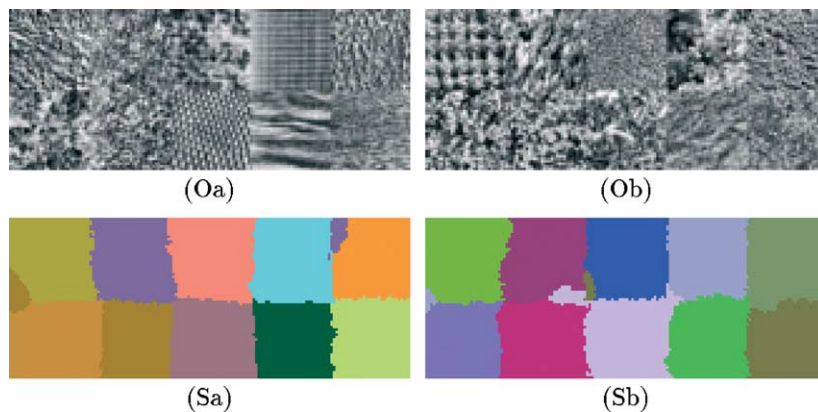


Fig. 15. Segmentation of 10-texture images: (Ox) original image, (Sx) segmentation using the multi-label fast marching algorithm with supervised feature extraction.

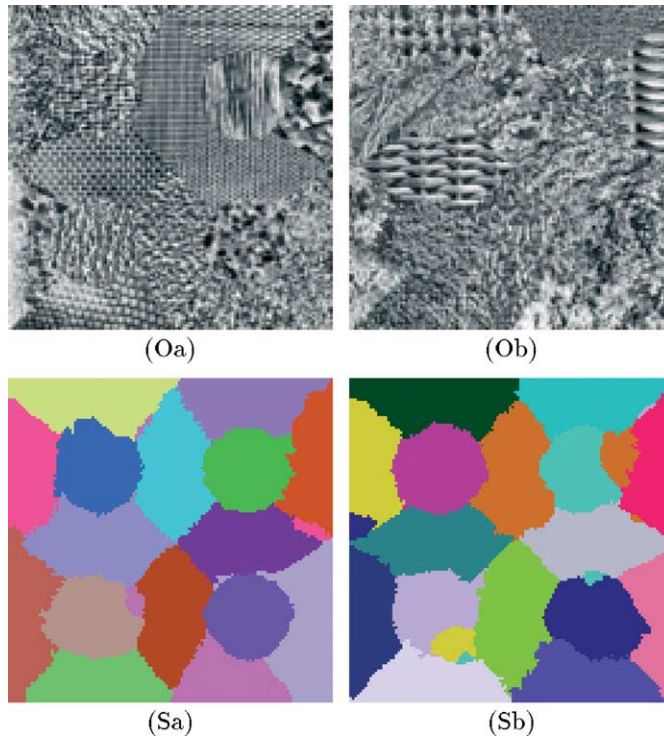


Fig. 16. Segmentation of 16-texture images: (Ox) original image and (Sx) segmentation using the multi-label fast marching algorithm with supervised feature extraction.



Fig. 17. Segmentation results of the multi-label fast marching algorithm on the synthetic images.

Initially 30–40% points were labelled according to the technique of Section 5.2. In Fig. 23 we show the final segmentation result for three different numbers of classes. The result is semantically acceptable for the three cases, while for a lower number of labels the classes are not simply merged, but mixed, resulting to false segmentation maps.

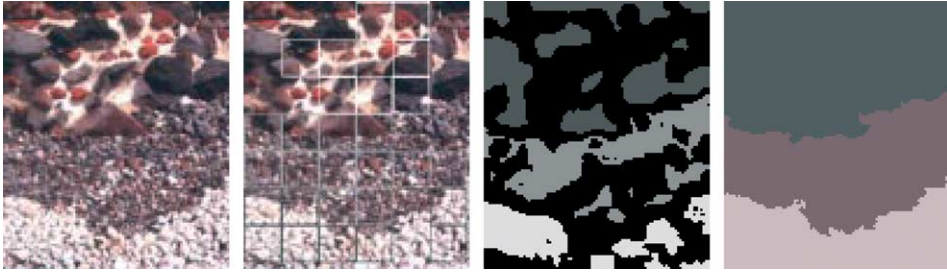


Fig. 18. Original *SeaStones* image, blocks selected for feature extraction, initial map, and final result of level set segmentation.



Fig. 19. Original *Village* image, blocks selected for feature extraction, initial map, and final result of level set segmentation.

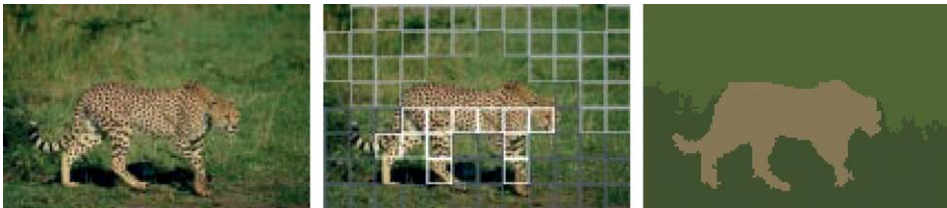


Fig. 20. Original *Leo* image, blocks selected for feature extraction, and final segmentation.



Fig. 21. Original *Tree* image, initial map, and final segmentation.





Fig. 22. Original *Road* image and final segmentation.

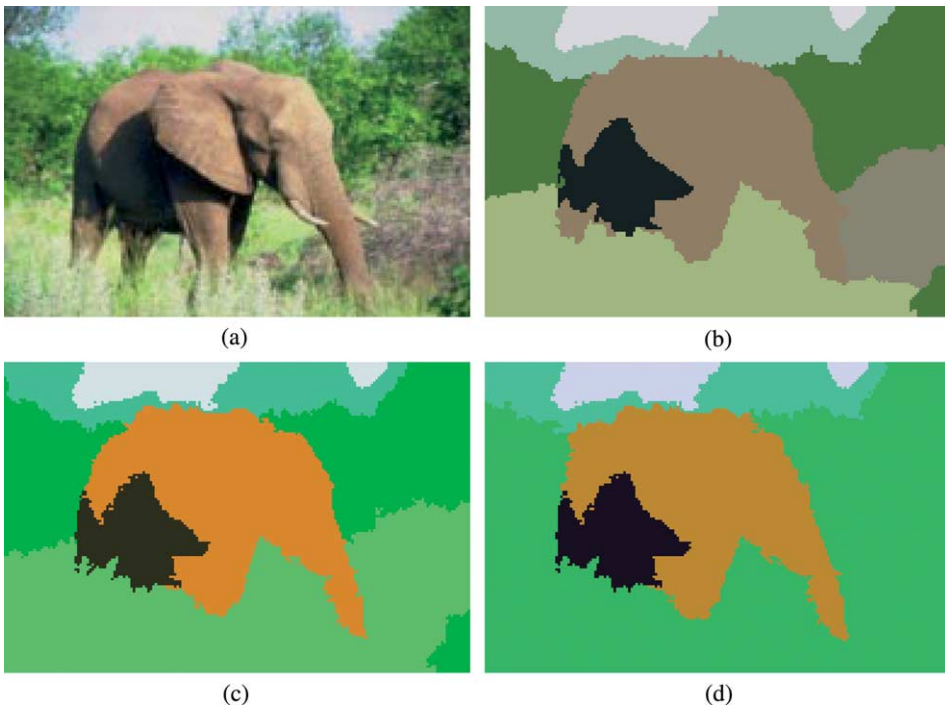


Fig. 23. Original *Elephant* image (a), result of level set segmentation with seven labels (b), six labels (c), and five labels (d).

## 6. Discussion and conclusions

In this paper we have addressed the problem of image segmentation based on colour and texture features. For texture description DWF are used for decomposing the image into different frequency bands. The components of the wavelet frames analysis

were shown to be particularly effective in exploiting periodicities of the texture patterns, as these patterns are expressed in different scales and different orientations. The wavelet frames decomposition used in our work is extremely simple, implemented using 5-tap 1D filters. In the future it will be interesting to find better filter-banks for segmenting given textured images. Colour was described in the *Lab* coordinate system by 2D histograms of the  $(a, b)$  components; the intensity histogram can be also used, if relevant for the segmentation.

New segmentation methodologies have been proposed, using both the wavelet frame analysis and colour histograms. In the first stage the parameters of the texture patterns are automatically extracted. This procedure starts by subdividing the image into equally sized blocks and then rejecting those blocks found to be heterogeneous with respect to their colour–texture content. The texture parameters are estimated after applying a hierarchical clustering procedure in the remaining blocks. The proposed scheme assumes only that the number of different colour–texture classes is known. Often this is a small number. In some cases a hierarchy of relevant segmentation maps could be obtained for different numbers of classes (Fig. 23).

The problem of automatically determining the number of labels is an important open issue. Among various approaches we have investigated a cluster validation test and a confidence test of the segmentation result. For cluster confidence we search for an indicator of the clustering performance. The validity of the segmentation can be measured by the *minimum description length* criterion, which takes into account the likelihood distance and the complexity of the resulting segmentation map. The last approach might be considered computationally expensive, but it has the advantage of global assessment for all parameters, not only the number of labels. Experiments we have made until now, show that this approach could be used for validating the result of the segmentation procedure.

Two new segmentation algorithms have been introduced. The first one is based on the deterministic relaxation algorithm; the second is called multi-label fast marching, an extension of the classical fast marching algorithm, and is based on a level set method, where the propagation speed is determined by the a posteriori probability of the labels. Both algorithms gave very good segmentation results on synthetic texture images and on natural scenes, where texture and colour are often combined for obtaining the best label field. Globally the multi-label fast marching algorithm was more performant on segmenting natural scenes.

As expected, in the absence of an objective criterion, a number of parameters must be selected by the user. Our method has the advantage of combining in a simple way intensity, colour and texture features. Still the user must decide which feature is most relevant for segmentation. We show results (Fig. 10) demonstrating that different features lead to similar segmentation maps. In other cases, different features have additional contributions for semantic segmentation. In general, if the wavelet frame decomposition is used, a maximum number of three or four levels could suffice, depending on the image size. In any case the human operator has a primordial role in image segmentation useful for semantic interpretation.



## References

- Belongie, S., Carson, C., Greenspan, H., Malik, J., 1998. Color- and texture-based image segmentation using em and its application to content based image retrieval. *Int. Conf. Comput. Vision*.
- Besag, J., 1974. Spatial interaction and the statistical analysis of lattice systems (with discussion). *J. Roy. Statist. Soc. B* 36, 192–326.
- Besag, J., 1986. On the statistical analysis of dirty images. *J. Roy. Statist. Soc.* 48, 259–302.
- Bovic, A.C., Clark, M., Geisler, W.S., 1990. Multichannel texture analysis using localized spatial filters. *IEEE Trans. Pattern Anal. Machine Intell.* 12 (January), 55–73.
- Brodatz, P., 1966. *A Photographic Album for Artists and Designers*. Dover, New York.
- Chen, P.C., Pavlidis, T., 1983. Segmentation by texture using correlation. *IEEE Trans. Pattern Anal. Machine Intell.* 5 (January), 64–69.
- Chou, P., Brown, C., 1990. The theory and practice of Bayesian image labeling. *Int. J. Comput. Vision* 4, 185–210.
- Comer, M., Delp, E., 1999. Segmentation of textured images using a multiresolution Gaussian autoregressive model. *IEEE Trans. Image Process.* 8 (March), 408–420.
- Duda, R., Hart, P., 1973. *Pattern Classification and Scene Analysis*. Wiley, New York.
- Geman, S., Geman, D., 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.* 6, 721–741.
- Jain, A., Farrokhnia, F., 1991. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition* 24, 1167–1186.
- Kashyap, R.L., Chellappa, R., Khotanzad, A., 1982. Texture classification using features derived from random field models. *Pattern Recognition Lett.* 1, 43–50.
- Krishnamachari, S., Chellappa, R., 1997. Multiresolution Gauss–Markov random field models for texture segmentation. *IEEE Trans. Image Process.* 6 (Feb.), 251–267.
- Laine, A., Fan, J., 1996. Frame representation for texture segmentation. *IEEE Trans. Image Process.* 5 (May), 771–780.
- Mallat, S.G., 1989. A theory of multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Anal. Machine Intell.* 11 (Jan.), 674–693.
- Mao, J., Jain, A., 1992. Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern Recognition* 25, 173–188.
- Paragios, N., Deriche, R., 1999. Geodesic active regions for supervised texture segmentation. *Int. Conf. Comput. Vision*.
- Porat, M., Zeevi, Y.Y., 1989. Localized texture processing in vision: analysis and synthesis in Gaborian space. *IEEE Trans. Biomed. Eng.* 36, 115–129.
- Raghu, P.P., Yegnanarayana, B., 1996. Segmentation of Gabor-filtered textures using deterministic relaxation. *IEEE Trans. Image Process.* 5 (December), 1625–1636.
- Raghu, P.P., Poongodi, R., Yegnanarayana, B., 1997. Unsupervised texture classification using vector quantization and deterministic relaxation neural network. *IEEE Trans. Image Process.* 6 (October), 1376–1387.
- Randen, T., Husoy, J.H., 1999. Filtering for texture classification: a comparative study. *IEEE Trans. Pattern Anal. Machine Intell.* 21 (April), 291–310.
- Randen, T., Husoy, J.H., 1999. Texture segmentation using filters with optimized energy separation. *IEEE Trans. Image Process.* 8 (April), 571–582.
- Reed, T., du Buf, J.M.H., 1993. A review of recent texture segmentation and feature extraction techniques. *CVGIP: Image Understanding* 57 (May), 359–372.
- Sethian, J., 1996. A marching level set method for monotonically advancing fronts. *Proc. Natl. Acad. Sci.* 93, 1591–1595.
- Sethian, J., 1996. Theory, algorithms, and applications of level set methods for propagating interfaces. *Acta Numer.*, 309–395.
- Sethian, J., 1999. Fast marching methods. *SIAM Rev.* 41, 199–235.
- Sifakis, E., Garcia, C., Tziritas, G., 2002. Bayesian level sets for image segmentation. *J. Visual Commun. Image Representation* 13, 44–64.

- Unser, M., 1995. Texture classification and segmentation using wavelet frames. *IEEE Trans. Image Process.* 4 (November), 1549–1560.
- Young, T., Fu, K.-S. (Eds.), 1986. *Handbook of Pattern Recognition and Image Processing*. Academic Press, New York.