Abstract—We propose a general purpose image segmentation framework, which involves feature extraction and classification in feature space, followed by flooding and merging in spatial domain. Region growing is based on the computed local measurements and distances from the distribution of features describing the different classes. Using the properties of the label dependent distances spatial coherence is ensured, since the image features are described globally. The distribution of the features for the different classes are obtained by blockwise unsupervised clustering based on the construction of the minimum spanning tree of the blocks’ grid using the Mallows distance and the equipartition of the resulting tree. The final clustering is obtained by using the k-centroids algorithm. With high probability and under topological constraints, connected components of the maximum likelihood classification map are used to compute a map of initially labelled pixels. An efficient flooding algorithm is introduced, namely, Priority Multi-Class Flooding Algorithm (PMCFA), that assigns pixels to labels using Bayesian dissimilarity criteria. A new region merging method, which incorporates boundary information, is introduced for obtaining the final segmentation map. Therefore, the merging stage is based on region features and edge localization. Segmentation results on the Berkeley benchmark data set demonstrate the effectiveness of the proposed methods.

Index Terms—Image Segmentation, Bayesian Statistical Analysis, Minimum Spanning Tree, Tree Partition, Region Growing, Flooding Algorithm, Region Merging

I. INTRODUCTION

Image segmentation is a key step in many image analysis and interpretation tasks. Segmentation of color textured images has become a necessity for many applications, such as content based image retrieval and object recognition. Finding semantic regions is the ultimate goal of segmentation for image understanding.

Despite the plethora of methodologies for image segmentation, a general-purpose, admitting some user-tuned parameters, algorithm that addresses the whole range of segmentation problems and applications does not exist. This is due to the complexity and ill-posedness of segmentation problems and the fact that often an unambiguous ground truth does not exist, instead the evaluation of the result is somehow dependent on subjective interpretation.

Considering image segmentation as an optimization problem we should introduce four kinds of constraints: boundary, shape, region and topology constraints. Boundary constraints refer to cues presenting significant differences between neighboring sites belonging to different regions. The objective could be limited to detect boundaries, without labelling regions. Soft shape constraints lead to boundary regularization, while hard shape conditions, assuming prior knowledge, could address specific rigid or deformable object localization. In any case, an accurate spatial localization is required. Region constraints refer to pixel grouping according to class properties and similarity criteria. The objective is to obtain regions that are uniform and homogeneous with respect to the selected features. Finally, topology constraints should limit the number of connected components. Incorporating the last constraints, proximity and continuation become relevant properties for grouping. The various meaningful constraints are not totally independent and are often implicitly related. For example, shape constraints generally affect also the topology, for which it is difficult to apply directly quantitative criteria.

A. Related work

We refer to very few well-known algorithms for illustrating the role of different constraints, either in region boundaries localization, or in image segmentation. In edge detection [1] only boundary constraints are taken into account, while contrast constraints can be integrated in region growing techniques [2]. The last category of techniques incorporates soft topology constraints in the propagation process. Boundary constraints combined with geometric shape constraints lead to geodesic active contours [3], where a global optimization method is applied. This approach is generalized in [4] giving a powerful method, because it introduces and deals with boundary, shape and region constraints.

While a lot of work has been done with regard to the first three constraints, topology constraints have received less attention and are taken into account only implicitly, mainly because it is difficult to handle them explicitly. However, topology constraints could aid the segmentation process for obtaining semantic regions, without sacrificing topology flexibility. One of the objectives of our work is to incorporate topology constraints in different stages of the grouping process. Incorporating topology constraints in the segmentation process allows for limiting the effect of visual appearance imperfections.

In our approach, region growing and graph theoretic techniques are used for enforcing topology constraints and spatial
The visual properties are mapped on a graph, where the weight of vertices measures the distance of local visual appearance from global class prototypes. Effectively, in the last years graph theoretic approaches revealed suitable for efficiently handling the spatial coherence. Visual groupings can be captured by mapping the perceptual features and coherence constraints to a graph. Shi and Malik [5] considered image segmentation as a graph partitioning problem in order to obtain perceptual groupings by focusing on local features and their consistencies in the image data. They used a normalized cut criterion for measuring both the dissimilarity between the different groups, as well as the similarity within the groups. Felzenszwalb and Huttenlocher [6] proposed an efficient graph-based method using a minimum spanning tree (MST) algorithm taking simple greedy decisions for region merging, while respecting global properties for image segmentation. Zabih and Kolmogorov [7] proposed a segmentation algorithm that operates simultaneously in feature space and in image space. An energy function is defined on a graph and the min cut algorithm provides the optimal spatially coherent grouping. Falcao et al. [8] considered a minimum-cost path forest in a graph for designing image processing algorithms taking into account the connectivity with possible use in image segmentation.

As our framework is applied to natural image segmentation we also refer here to recent related work and later we compare our results to those obtained by these successful approaches. In [9], texture features are modelled using mixture of Gaussian distributions. The mixture distribution is segmented by an agglomerative clustering algorithm derived from a lossy data compression approach. In [10], an energy function, which expresses the local smoothness of an image area, is derived by exploiting an intermediate step of modal analysis that is utilized in order to describe and analyze the deformations of a 3-D deformable surface model. This energy function is used as a criterion for a region growing algorithm.

B. The proposed approach

Our framework may be roughly separated in three main components, namely, feature extraction and classification in the visual space, using image data information, region growing (RG) or flooding in spatial domain, where the connectivity is based on the computed features of classes, and region merging. That way, our method belongs to techniques that involve global feature space analysis under spatial coherence constraints, such as that in [11], while our framework achieves these objectives sequentially, in order to regularize the segmentation map by topological constraints. The feature space analysis aims to achieve a classification rate as accurate as possible, in order to produce reliable massive connections. The spatial coherence uses the similarity and proximity principles in order to obtain a final labelling with a restricted number of connected components, and eventually with regularized boundaries. Therefore, the topological constraints in the adopted framework are implicitly related to the global characteristics of the different regions.

The proposed segmentation framework can operate completely unsupervised. It is not excluded however to incorporate prior knowledge concerning relevant visual cues and the number of classes. The extracted features are classified using unsupervised clustering in an image block basis to provide the required stability for the estimation of classes’ features, while pixel based algorithms that follow, effectively rule out the artifacts due to the block based feature classification.

The main steps of the whole algorithm are given in Table I. The image is divided into overlapping blocks, whose content is described by the distribution of three color Lab components and a textureness measure. At the first stage of learning, topology constraints are used. Based on the Mallows distance the blocks are grouped in clusters with homogeneous and equivalent content, by constructing and partitioning a minimum spanning tree. Then a k-centroids clustering procedure is applied using the Bhattacharyya distance between the distributions of features extracted from each block cluster. Based on the resulting clustering, the feature classes are extracted and described, one per cluster, and then, probabilistic distances are used to select and label a set of pixels that belong to each class with high confidence, thus providing an initial map of almost correctly labelled pixels. Different existing components of the same class are captured by the initial seed-regions.

<table>
<thead>
<tr>
<th>Input</th>
<th>Original image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>Block size, Maximum number of classes</td>
</tr>
<tr>
<td>1.</td>
<td>Block-wise feature description computation</td>
</tr>
<tr>
<td>2.</td>
<td>Image blocks partition using a MST representation and Mallows distance</td>
</tr>
<tr>
<td>3.</td>
<td>Clustering using k-centroids algorithm and Bhattacharyya distance</td>
</tr>
<tr>
<td>4.</td>
<td>Computation of a posteriori distributions and distances</td>
</tr>
<tr>
<td>5.</td>
<td>Initial labelling</td>
</tr>
<tr>
<td>6.</td>
<td>Priority Multi-Class Flooding Algorithm</td>
</tr>
<tr>
<td>7.</td>
<td>Region merging</td>
</tr>
<tr>
<td>Output</td>
<td>Segmentation map</td>
</tr>
</tbody>
</table>

TABLE I

THE MAIN STEPS OF THE PROPOSED ALGORITHM.

Having the data modelling available and the initial map of almost correctly labelled pixels, we propose a new algorithm in order to fill the initial map of decisions, using statistical dissimilarity criteria. The new algorithm, Priority Multi-Class Flooding Algorithm (PMCFFA) introduced in [12], imposes strong topology constraints in such a way that also allows topological flexibility. The segmentation map is constructed respecting the proximity principle, with linear computational cost to the number of unlabelled image sites.

The final step of our framework is a new statistical region merging algorithm. The proposed merging predicate is based on both region similarity and boundary separability criteria. Regions are iteratively merged by a greedy, graph-based algorithm according to the proposed measure, which is updated after each merging of regions.

This paper is organized as follows: in Section II, the tree equipartition algorithm and the feature extraction method are presented. In Section III, the new flooding algorithm is described in detail. In Section IV, our method for region merging is presented. In Section VI, we present the obtained results for natural image segmentation on the Berkeley benchmark data.
set.

II. AUTOMATIC FEATURE EXTRACTION

Features considered for segmentation are Lab color components and textureness. For an unsupervised approach and for simplicity reasons the number of classes is prefixed. As far as the number of classes is at least equal to the number of meaningful existing classes, the over-classification is not a cause of errors, as the final step of region merging handles the trade-off between region similarity and boundary reliability. Unsupervised clustering of meaningful data is used in order to obtain relevant classes and their description.

As the class descriptors are used for visual groupings, they should be spatially coherent. A simple approach for obtaining spatial coherence is to divide the whole image into blocks, maybe overlapping, and extract features from the blocks. As the empirical probability density is a powerful, and often sufficient, statistic for classifying pixels, the block size should be sufficiently large to accurately estimate such descriptors, and not very large in order to not miss semantically interesting regions. In this work, the block size is prefixed and half overlapping is assumed. A minimum spanning tree is constructed on the blocks' grid and partitioned for automatically clustering the neighboring blocks. The obtained clusters are further grouped using the k-centroids algorithm with a fixed number of classes providing good descriptions for the visual classes.

In what follows, we describe in detail the textureness measure, the probabilistic distance metrics used, the new tree partition method and the final block clustering technique.

A. Textureness

A detailed description of texture content could be obtained using a wavelet representation [13]. However, in natural image segmentation only the discrimination in regions with texture or without texture is often sufficient. Therefore we are limited to use a measure of textureness. In the past, the variance or the gradient matrix have been used for measuring the amount of texture. These measures are not convenient for segmentation, because they detect contours too. A measure with low sensitivity to strong edges is needed. This distinction is well clarified in [14]. In our work, we define a simple and rotation-invariant measure with low sensitivity to edges.

As textureness is not in general related to color, we obtain at first the principal image component Y from the three RGB components. Let \( Y_m \) be the output of a median filter, in our implementation of size 5 \( \times \) 5. The textureness is measured by applying again the same median filter on \( |Y - Y_m| \). Therefore at every image site, in addition to the three color components, we have a measure of textureness. The proposed texture feature has high values at areas with high-frequency content, but low response at strong image edges. Indeed, since the median filter is invariant to step edges, the difference \( Y - Y_m \) results to the removal of these edges. The second median filter can give a robust statistic measure of the amount of texture. In Figure 1 is given the image of the textureness for images TR-314016 and TR-106020 of the Berkeley Segmentation Data Set. Boundary lines on the body of the animals have been essentially removed.

As the textureness is not relevant for discriminating image segments for the whole image data set, we apply a multimodality test to decide, if textureness will be a feature useful for segmenting a specific image. The test is based on the empirical probability density function of textureness. It is simple, seeking for local maxima of the histogram, which could be considered as distinct modes. If more than one local maxima are found, and if the distance between the last and the first local maxima is greater than 0.05 times the maximum value of the textureness, the distribution is considered as multimodal. From our experience (Section VI), we conclude that the use of textureness in the case where a unimodal distribution is detected could result in slight loss of both performance and efficiency\(^1\), as non relevant cues are used.

B. Probabilistic Distance Metrics on Data Sets

The image being divided into blocks, we describe visual features by the empirical distributions of the three color components and the textureness measure, thus obtaining a rich and powerful tool for clustering and classification. Therefore, we need probabilistic distance measures for block data clustering. Since the block size is constant, we can adopt a simple and powerful distance measure, namely the Mallows distance [15]. In general, the Mallows distance between two distributions can be defined by the minimum of the expected difference between the two random variables, taken over all possible joint probability distributions. The distance is efficiently computed under the assumption of independent data components. For each component, the corresponding data are sorted and then the distance is measured by the mean absolute difference [16].

After clustering the blocks, since the size of clusters differs, and since our objective is an image data classification, as accurate as possible, we use the Bhattacharyya distance. Indeed, this distance measure is strongly related to a bound of the classification error probability [17]. The Bhattacharyya distance between two distributions corresponding to two classes

\(^1\)Efficiency refers to the computation time and performance to the quality of the results.
is:
\[ D(p_i, p_j) = -\ln \left( \int_{\xi} \sqrt{p_i(\xi)p_j(\xi)}d\xi \right), \]  
(1)
where \( \xi \) is the classification feature vector, and \( p_i(\xi), p_j(\xi) \) are the conditional probability density functions \( p(\xi|i), p(\xi|j) \) of feature vector. In the case of discrete random variables the distance takes the form

\[ D(p_i, p_j) = -\ln \left( \sum_k \sqrt{p_i(k)p_j(k)} \right), \]  
(2)
where \( p_i \) and \( p_j \) are probability distributions. The kernel density estimator proposed by Botev et al. [18] is used for the distribution estimation. It is obvious that the distance depends on the number of histogram bins. In our work the number of bins for the \( Lab \) components is adapted to the image content. We give this number for the \( L \) component,

\[ H = 2^{\lfloor \log_2(\max L - \min L) \rfloor - 1}. \]

The same formula is used for the other components. We assume that the textureness follows a Gamma distribution. Its parameters are estimated using the moments method.

Finally, admitting again the independence of image components and textureness, the distance is the sum of the respective distances. We emphasize here that the description of visual features by their probability distributions and the use of a probabilistic distance leads to a generic framework, independent of the different visual cues.

C. Tree Equipartition

This section describes the proposed Tree Equipartition algorithm that provides an automatic clustering of neighboring blocks taking into account topology constraints. The given frame is divided into blocks, where each block is overlapping with all its adjacent blocks by the half of the block size. The graph weights are given by the Mallows distance of three color components and of the textureness measure of the corresponding blocks, as defined in the previous section. First, the Minimum Spanning Tree (MST) on the blocks’ grid is constructed.

Prim’s algorithm implemented using a Fibonacci heap and an adjacency list, with computational cost \( O(E + N \log N) \) [19], could be used for MST computation. In the graph corresponding to the \( N \) image blocks, which are the nodes of the graph, there are nearly \( E = 4N \) edges. Therefore, \( O(E + N \log N) = O(4N + N \log N) = O(N \log N) \), making the MST-based method computationally efficient. MST keeps most important graph edges, reducing the problem of graph partitioning to tree partition. If we remove an edge from the MST, we get its partition into two sub-trees, which could be considered as two clusters of image blocks. Thus, tree clustering to \( M \) clusters can be performed by the selection of \( M − 1 \) edges that should be removed. According to the standard MST partitioning algorithm, at first the MST of the given graph is constructed, and then the \( M − 1 \) longest edges are removed to produce \( M \) trees, each of which corresponds to a cluster. Intuitively, the longest edges, which are removed, separate the natural clusters, and the shortest edges, which are retained, connect close data points within natural clusters. However, if we apply this algorithm to the MST, adjacent clusters of low boundary difference will not be detected and over-segmentation of rich in content clusters will be observed.

In order to solve this issue, we formulate and solve the tree equipartition problem similarly to the curve equipartition problem [20]. The equipartition principle has been successfully applied on polygonal approximation [21] and key frames detection problems [22], yielding equal in error and content segments, respectively. The goal of curve equipartition is to locate consecutive curve points, so that the curve can be divided into segments with equal chords under a distance function. Similarly, in a given undirected weighting tree the goal is the uniform partition of the edge set of the tree (of \( N \) nodes) into \( M \) connected components (sub-trees) under the objective of almost equal diameters, where the sub-tree diameter is defined as the “longest shortest path” of the sub-tree. Finally, the resulting clusters will have equal diameters. Similar to key frames detection problem [22], the content variation within the resulting clusters is roughly equal. Moreover, a near optimal solution of minimization of maximum sub-tree diameter is achieved when the sub-tree diameters are equal. This property is analyzed in [21], where a near optimal solution of polygonal approximation is achieved, when the approximation errors per line segment are equal, as the error is shared between all the segments.

The proposed hierarchical tree equipartition method (HTE) sub-optimally solves the Tree equi-partition problem. The input of the method is the undirected weighting MST (\( T \)) and an objective function \( f \) to be minimized. In our case, \( f \) is the cluster (sub-tree) diameter. Let \( F = T_1 \cup \ldots \cup T_k \) be a set of sub-trees, an intermediate result of the method.

- Initially, \( F = T \). Let \( \hat{T} \in F \) be the sub-tree of \( F \), where the objective function \( f \) is maximized \( f(\hat{T}) \geq f(T_i), \forall i \in \{1, \ldots, k\} \).
- Then, it splits the sub-tree \( \hat{T} \) into two sub-trees minimizing the maximum diameter of the two sub-trees, by removing an edge from \( \hat{T} \).
- The method iteratively runs for at least \( M_{min} \) (e.g. \( M_{min} = 10 \)) steps and stops when the maximum sub-tree diameter is lower than a threshold \( d_{max} \), or attains a maximum number \( M_{max} \) (e.g. \( M_{max} = 0.25N_B \)) of clusters, where \( N_B \) is the total number of blocks. \( d_{max} \) is automatically computed by a quotient (e.g. \( 1/N_B \)) of diameter of \( T \), or according to the distribution of graph edge weights.

In the final partition, according to HTE, the resulting hierarchical forest of sub-trees should have almost equal diameters. In order to reduce the computation of sub-trees’ diameter that is needed in sub-tree splitting step, first we estimate the shortest paths between every pair of nodes in the MST, using Johnson’s algorithm [23]. When a \( N \)-nodes MST is given, Johnson’s algorithm has a \( O(N^2) \) time complexity to compute the \( N \times N \) matrix of shortest path distances between every pair of nodes.

In each step of the proposed method, we split a sub-tree \( \hat{T} \) (of \( N \) nodes) into two sub-trees minimizing the maximum diameter of the two sub-trees, by removing an edge from \( \hat{T} \).
The computation cost of minimizing the maximum diameter of the two sub-trees is $O(N^2)$ using the $N \times N$ matrix of shortest path distances between every pair of nodes. Then, we need $O(N^2)$ to construct the two matrices of shortest path distances between every pair of nodes for the two resulting sub-trees. In the first step it holds that $N = N$, since $T = T$. Therefore, it holds that

$$T(N) = T(N - K) + T(K) + O(N^2),$$

(3)

where $T(N)$ and $K$ denote respectively the total computation cost and the number of nodes of a resulting sub-tree. According to Equation (3), in the worst case we need $O(N)$ iterations and when the resulting sub-trees are not balanced ($\min(N - K, K) = O(1)$), then the total computation cost is $T(N) = O(N^3)$. In the case where the resulting sub-trees are almost balanced (with almost the same number of nodes), that is often appears in real images, the total computation cost is $T(N) = O(N^2)$, since it holds that $T(N) = 2 \cdot T\left(\frac{N}{2}\right) + O(N^2)$.

Fig. 2 illustrates two examples of MST partitioning using the proposed hierarchical tree equipartition method. The red edges correspond to the edges removed by the HTE algorithm. The number behind each red edge shows the order of edge removal. Different colors (blue - green) have been used to illustrate the sub-trees highlighting their visual discrimination. In both cases, the largest clusters correspond to homogeneous in visual content regions, so that the resulting clusters could be considered equivalent on image content summarization.

**D. Feature Clustering**

In our approach pixel classification and grouping are based on the classes’ statistical description. In a general framework the number of classes is unknown, the best features should be selected and the feature descriptors have to be estimated. Automatic data clustering being the appropriate methodology, various aspects have to be considered in order to address globally and in a generic way the problem. In this work, for simplifying the computation cost and without essential loss of accuracy, we fix the maximum number of classes for the whole image data set.

From our experience on large image data sets, we concluded that either the $k$-means or the $k$--centroids algorithm performs well for grouping feature vectors and for the computation of feature descriptors of classes, determining with acceptable accuracy the corresponding sufficient statistics. The first step of clustering is the choice of the initial cluster centers, where we use the algorithm proposed by Kaufman and Rousseeuw [24] (cf. Appendix II). Then, the $k$-centroids iterations are applied and each cluster determined by the previous tree partition step is assigned to its most similar class. Bhattacharyya distance between color component and texturenss distributions is used for measuring dissimilarity. From the final clustering the image component distribution of a class is computed from all blocks assigned to the class and taking into consideration the blocks’ overlapping. In Figure 3 is given a classification result with a maximum likelihood criterion for image TR-314016 of the Berkeley Segmentation Data Set shown in Figure 5.

**III. FLOODING PROCESS FOR LABEL PROPAGATION**

In this Section, we describe in detail the proposed flooding algorithm, as well as its relation to Minimum Spanning Tree (MST) construction and Watershed methods. The relations of the two last algorithmic approaches have been presented in [25]. Label initialization is discussed at first in the next Subsection, since it is a pre-request for the flooding algorithm.

**A. Label Initialization**

The output of label initialization is a set of pixels, which are classified to class $l$ with high confidence using statistical tests. At first we define the distances of pixel features from classes based on the Bayesian rule, using therefore the a posteriori probabilities

$$d_l(s) = -\ln Pr\{l|\xi(s)\} = -\ln \frac{\varpi_k Pr(\xi(s))}{\sum_{k=1}^{L} \varpi_k Pr(\xi(s))},$$

(4)

$\varpi_k$ being the a priori probability of class $k$, and $\xi(s)$ being the feature vector at site $s$. Distances $d_l(s)$ can be interpreted as heights, as they are always non negative.

The topology constraint is also used for initializing the labelling. On the map of maximum a posteriori probability classification the connected components of each class are extracted separately. On the connected components a decision is taken, constraining therefore the whole component. The decision is based on the size of the component and the geometric mean of the a posteriori probability on the whole component. This is equivalent to take as statistics the mean distance from the corresponding class. The test is based on a heuristic. A component is considered as classified with high confidence, if the geometric mean of the a posteriori probability on the whole component, $\mathcal{P}_c$, satisfies the following inequality

$$\mathcal{P}_c > 1 - (1 - P_{\min})\frac{\log n_c}{\log n_{\max}}.$$
where $n_c$ is the number of pixels belonging to the component, $n_{\text{max}}$ is the maximum size of a component for the corresponding class. $P_{\text{min}}$ results from the classification of all image pixels according to the maximum a posteriori probability principle. It is

$$P_{\text{min}} = \min_{s} \max_{l} \Pr[l|\xi(s)].$$

For $n_c = n_{\text{max}}$ the threshold defined above (5) becomes $P_{\text{min}}$, and therefore, at least one component for any class is initialized. Hence, the threshold is defined so that it is decreasing with the size of the component. In Figure 3 are given the initial components of the segments for image TR-314016 of the Berkeley Segmentation Data Set shown in Figure 5.

The initially labelled pixels are defined to be at the zero level, in the topographic map interpretation of the classification criterion, while the height of the unlabelled pixels is given by the Bayesian rule in Equation (4). Starting with these initial regions a minimum spanning tree (MST) can be constructed for each label as described in detail in the next subsection.

### B. A min-max criterion for labelling

It is now assumed that an arbitrary number of seed regions has been assigned to each class with high confidence. Pixels initialized this way are hard constrained to belong to the selected class, admitting that the decision is confident. Let $R^\text{init} = \bigcup_{l=1}^{L} R^\text{init}_{l}$ be the set of initially labelled pixels. For any unlabelled pixel $s$ we can consider all the paths linking it to a labelled region. A path $C_l(s)$ is a sequence of adjacent pixels $\{s_0, \ldots, s_n\}$, where $s_n = s$ and all pixels of the sequence are unlabelled, except $s_0$ which has label $l$. The cost of a particular path is defined as being equal to the maximum cost of a pixel classification according to the Bayesian rule in Equation (4) and along the path

$$\max_{i=1, \ldots, n} d_l(s_i).$$

Finally, the labelling problem becomes equivalent to search for the shortest path under the above cost, as we can define the distance of any unlabelled pixel from the different classes as being the lowest height to climb for reaching site $s$,

$$\delta_l(s) = \min_{C_l(s)} \max_{s_i \in C_l(s)} d_l(s_i).$$

Therefore, the labelling decisions depend on the topographic surface. The sequence of labelling decisions introduce implicitly topology constraints.

If we consider the graph of unlabelled sites with 4-connections and the labelled connected components, we can define an edge weight as follows

$$w_l(s_{i-1}, s_i) = \max(d_l(s_{i-1}), d_l(s_i)).$$

According to the cycle property applied to the MST of this graph the heaviest edge of a path belonging to the MST is lighter than the heaviest edge of any other path connecting two vertices. Therefore, paths defined by Equation (6) belong to the MST of the graph defined above and the computation of $\delta_l(s)$ necessitates the construction of this MST. Prim’s algorithm, with computational cost $O(N \log(N))$ [19], could be used as well.

On the other hand, it is very interesting to remark that the labelling problem, as posed here, consists of constructing a topographic surface, on which watershed lines [26] can be determined. Hence, we can use a region growing procedure, like the immersion or flooding algorithm [27], for computing the above defined heights and distances and for classifying pixels, taking into account region features and topology constraints. In the following subsection our algorithm, based on the principle of the min-max Bayesian criterion for labelling, is presented in detail.

### C. Priority Multi-Class Flooding Algorithm

The Priority Multi-Class Flooding Algorithm (PMCFCA) imposes strong topology constraints based on Equation (6). All initially labelled regions are propagated simultaneously and most likely decisions are taken as soon as possible. This algorithm shares the Bayesian criterion and the simultaneous label propagation with the Multi-label Fast Marching Algorithm [28], while here a graph-based discrete approach permits to overcome some weaknesses of the former method. Experimentation has shown that, in some cases, Fast Marching cannot appropriately expand the initial regions on noisy statistical surfaces, which are not characterized by pixel-wise smoothness. Such noisy surfaces may arise from the statistical modelling of texture regions. The result becomes worse, if the size of initial regions is small. Furthermore, in such cases, segmentation using Fast Marching results in strongly inaccurate boundaries between regions. However, this is an expected behaviour, because the Fast Marching algorithm aggregates the digitally computed gradient for each path of pixels and this computation is sensitive to noisy data.

All the contours of initially labelled regions are propagated towards the space of unlabelled image pixels, according to similarity criteria, which are based on the label and the segmentation features. Contour pixels $s$ are sorted according to their dissimilarity $\delta_l(R(s))$ from the label $l(R)$ of regions $R$ they adjoin and at each step, a group of contour pixels of minimum dissimilarity are set to the label that most probably belong.

In addition, group labelling refers to pixels $s$, which are placed on region contours at a step of the propagation progress and their metric $\delta_l(s)$ against label $l$ is similar. This fact, implies the quantization of distance metric $d_l(-)$, which in turn leads to the reduction of spatial redundancy that often appears during growing. Spatial coherence arises when neighboring pixels have almost equal distance value or equivalently, as it is clear by the definition of $d_l(-)$, when neighboring pixels have almost equal posterior probabilities for a class given their feature vector.

Furthermore, two decision thresholds, namely $T^1_l$ and $T^2_l$, with $T^1_l < T^2_l$, may be defined on the metric, when $d_l(s)$ measures the uncertainty of the decision to assign pixel $s$ to label $l$. Given the thresholds, if $d_l(s) < T^1_l$, pixel $s$ belongs to $l$ with great certainty and should be labelled as soon as it is scanned. We note that according to the MAP criterion, a pixel
The merging cost is defined as follows. 

\[ \text{cost} = \sum_{s \in B(i,j)} \log \left( \frac{p_h(G(s))}{p_i(G(s))} \right) - \nu |B(i,j)|, \]

where \( B(i,j) \) is the set of boundary points between \( A_i \) and \( A_j \), \( |B(i,j)| \) is the number of these points and \( \Pr \{ B(i,j) \} \) is a probability attributed to this boundary, which is estimated from the result of edge detection.

The first term is the dissimilarity in appearance between the two adjacent regions and is measured by the Bhattacharyya distance using the distribution of \( Lab \) components and of the textuness. The size of regions is taken into account such a way that the merging of small-size regions is favored. The second term is measuring the likelihood ratio for testing the hypothesis of a strong gradient on the boundary. Therefore, if the boundary separating two adjacent segments is found in a homogeneous region, then this term is negative and merging is favored. The third term regularizes the final segmentation results by penalizing the over-segmentation. For edge detection we use Canny's algorithm [1]. The gradient magnitude results from the three \( Lab \) components. In Equation (8), we need the distribution of gradient magnitude under two hypotheses and an estimation of the probability \( \Pr \{ B(i,j) \} \). Details are given in Appendix IV.

To our knowledge the definition of the merging cost (8) is new. It is inspired from the Mumford-Shah functional [29]. In the second term the log likelihood ratio of the gradient magnitude is used in place of the gradient magnitude directly. The third regularization term is adapted to situations where false contours may appear, and therefore they should be taken into account by defining a suitable probability measure. As the merging costs compares the cost of merging two regions to the cost of non-merging, the threshold should be set to zero. We therefore select the values of parameters \( \lambda \) and \( \nu \), such that a negative value of the above criterion should result to the region merging.

In addition, it was useful to evaluate the risk for merging two regions, which should depend on the inhomogeneity of each region or the presence of a specific structure that distinguishes the region. Therefore, the risks of the two adjacent regions are added to the merging cost of Equation (8). For measuring the distance on textuness between two regions we assume Gamma distributions and use the Bhattacharyya distance. The first parameter of Gamma distribution, denoted by \( a \) (cf. Appendix I), could be interpreted as the equivalent number of independent variables, while the second parameter, denoted by \( b \), could be considered as a measure of the variance per independent variable. The discriminative power of the textuness measure is not satisfactory in case of regions with strong and similar variance. For this reason, a term is added to the above merging criterion (8), which becomes:

\[ M(A_i, A_j) = M'(A_i, A_j) + \frac{n_in_j}{2(n_i + n_j)} \log b_i b_j. \]

The main problem now is to fix the two regularizing parameters, \( \lambda \) and \( \nu \). We fix the parameters so that the merging stops when \( M(A_i, A_j) \geq 0 \), for reasons presented previously. If two regions appear to be similar, that is, if \( d(A_i, A_j) \) is nearly zero, the merging criterion \( M(A_i, A_j) \) should be negative, even if an edge could be in theory detected. In this particular case we have

\[ M(A_i, A_j) \approx \lambda \sum_{s \in B(i,j)} \log \left( \frac{p_h(G(s))}{p_i(G(s))} \right) - \nu |B(i,j)|. \]

Taking the expected value, we obtain a relation between \( \lambda \) and \( \nu \),

\[ \nu \geq \lambda \int p_h(g) \log \left( \frac{p_h(g)}{p_i(g)} \right) dg. \]

If we fix the ratio \( \nu/\lambda \) using the equality in the above relation, we have only to determine the value of \( \nu \). It may be useful to note that the ratio \( \nu/\lambda \) is bounded by the Kullback distance between the two gradient distributions.

The regularization parameter \( \nu \) should depend on the image content and on the likelihood of the segmenting boundaries. It should be greater, when the probability of over-segmentation is greater. As we cannot discriminate different textures, the more textured is an image the more probable will be to obtain an over-segmented map after the classification and flooding stages. Therefore the parameter \( \nu \) should be proportional to
a global statistical measure of textureness. For this reason we define $R_t$, the ratio of the mean value of the textureness measure to its median value. The more important is $R_t$, more the image is dominated by textured segments, which risk to be over-segmented. On the other hand, we should test the validity of the segmenting boundaries. The mean value $L_g$ of $\log \frac{p(G)}{p(B)}$, computed on the candidate region boundaries, could be a relevant statistic for the validity test. $L_g$ can be positive or negative. We can also define a global measure indicating if there are large homogeneous regions. Such a measure could be the ratio of the mean value of the gradient magnitude to its median value, denoted $R_g$. Roughly, we could consider that, if the ratio is superior to $r = 1.5$, there exist large homogeneous regions. The last case has been adequately considered by the dependence of $\nu$ on $R_t$, as described above. In the contrary case, when homogeneous regions do not exist, $L_g$ should be in addition relevant for setting $\nu$. A term approximately proportional to $L_g$ is added or subtracted. Finally, using sigmoid functions, we propose to use the following value for $\nu$:

$$\nu = \nu_0(1 + R_t + (1 - \tanh(10(R_g - r))) \tanh(0.2L_g)).$$  \hspace{1cm} (9)

A greedy algorithm iteratively merges the regions according to the merging criterion (8). The algorithm proceeds by merging at each step the two regions linked by the minimum $M(A_i, A_j)$. The description of the segments is updated after each step. Therefore, a hierarchy is constructed, which is stopped when the criterion becomes positive, giving the final segmentation map. In Figure 4 is illustrated the result of the hierarchy constructed by the merging algorithm on image TR-314016 of the Berkeley Segmentation Data Set shown in Figure 5. The first map is the output of flooding algorithm, with 90 segments, while the image in the medium of second raw is the final result with 31 segments, which is obtained using the automatic algorithmic stop.

V. SEGMENTATION

VI. EXPERIMENTAL RESULTS

We applied our method to the well known Berkeley Segmentation Data Set (BSDS) [30]. BSDS consists of natural color images, divided in a “train” (TR) and a “test” (TE) subset, manually segmented by humans. All the results obtained using our unsupervised algorithm to the whole Berkeley Segmentation Data Set (BSDS) are given in http://www.eecs.berkeley.edu/~yang/software/lossy_segmentation/. The same parameters are used for the whole data set, showing that our algorithm has good robustness properties. The block size was $24 \times 24$ with an overlapping of 12 pixels, for an image size equal to $320 \times 214$. This image size is selected, because it is used in the literature for comparing different algorithms. The block size corresponds to about 1% of the image size considered as an acceptable segment resolution. The maximum number of classes is fixed to 11 for the whole data set. PMCFA is applied by quantizing heights in 50 bins. The merging parameters are: $\nu_0 = 4, r = 1.5$.

In Figures 5-6 are given results from our algorithm (PMCFA), as well as those of Felzenszwalb and Huttenlocher (FH) [6] and Yang et al. (CTM) [9]. Segmentation maps for each image and algorithm are displayed in a uniform color corresponding to the mean color of the segmented region. The images selected for the illustration are characterized by high diversity of visual content or gradual transitions and reflections, which, in general, make segmentation a difficult task. The qualitative comparison of the two proposed algorithms against two algorithms (FH and CTM) known for their good results can be partially done on these Figures. The results of CTM have been downloaded from http://www.eecs.berkeley.edu/~yang/software/lossy_segmentation/.

As a general remark, we could say that algorithm FH often leads to over-segmentation, while CTM leads to under-segmentation, depending on parameter $\gamma$ [9]. Our algorithm, adequately using the description of feature classes leads mostly to a meaningful region decomposition. In most results in Fig. 5-6, regions extracted by our algorithm correspond to objects, or classes of objects, in particular when the automatically computed statistics of the selected cues are sufficient to describe their class, as well as to distinguish them from their background.

The proposed method has been tested against the ground
truth maps given by humans for each color image of BSDS. The metrics used for the comparison against human provided maps are: the Probabilistic Rand Index (PRI) [31], the Global Consistency Error (GCE) [30], the Variation of Information (VoI) [32] and the Boundary Displacement Error (BDE) [33]. For each image, algorithm and metric, the segmentation result is compared against each one of the ground truth results that have been provided by humans for the color image set. The average of metric scores is used to measure the performance of the algorithm on each image.

The results shown in Fig. 5-6 are among those with bad indices. We give in Table II these indices to compare with the average performance in Table III.

The classification stage of our algorithm gives better mean performance scores for all considered indices. In the comparison are included the following algorithms: Felzenszwalb and Huttenlocher (FH) [6], Normalized Cuts [5], Mean-Shift [34], CTM [9] and Modal Image Segmentation [10]. For these algorithms we report here the indices as given in the literature.

<table>
<thead>
<tr>
<th></th>
<th>PRI</th>
<th>GCE</th>
<th>VoI</th>
<th>BDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Humans</td>
<td>0.8754</td>
<td>0.0797</td>
<td>1.1040</td>
<td>4.9040</td>
</tr>
<tr>
<td>PMCFA</td>
<td>0.8103</td>
<td>0.1711</td>
<td>1.9045</td>
<td>7.2310</td>
</tr>
<tr>
<td>FH</td>
<td>0.7841</td>
<td>0.1895</td>
<td>2.6647</td>
<td>9.4947</td>
</tr>
<tr>
<td>Ncuts</td>
<td>0.7229</td>
<td>0.2182</td>
<td>2.9329</td>
<td>9.6038</td>
</tr>
<tr>
<td>Mean-Shift</td>
<td>0.7550</td>
<td>0.2598</td>
<td>2.4770</td>
<td>9.7001</td>
</tr>
<tr>
<td>CTM (γ = 0.10)</td>
<td>0.7561</td>
<td>0.1796</td>
<td>2.4640</td>
<td>9.4211</td>
</tr>
<tr>
<td>CTM (γ = 0.15)</td>
<td>0.7627</td>
<td>0.1846</td>
<td>2.2035</td>
<td>9.4902</td>
</tr>
<tr>
<td>CTM (γ = 0.20)</td>
<td>0.7617</td>
<td>0.1877</td>
<td>2.0236</td>
<td>9.8962</td>
</tr>
<tr>
<td>MIS (λ = 50)</td>
<td>0.7825</td>
<td>0.1942</td>
<td>1.9348</td>
<td>7.8263</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>PRI</th>
<th>GCE</th>
<th>VoI</th>
<th>BDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>TR-15088</td>
<td>0.5090</td>
<td>0.0842</td>
<td>2.4032</td>
<td>6.1134</td>
</tr>
<tr>
<td>TR-175043</td>
<td>0.3437</td>
<td>0.1048</td>
<td>2.7776</td>
<td>15.0352</td>
</tr>
<tr>
<td>TR-304034</td>
<td>0.4696</td>
<td>0.0677</td>
<td>2.8596</td>
<td>16.3260</td>
</tr>
<tr>
<td>TR-15088</td>
<td>0.5152</td>
<td>0.0357</td>
<td>1.5608</td>
<td>9.7229</td>
</tr>
<tr>
<td>TR-26031</td>
<td>0.4207</td>
<td>0.1146</td>
<td>3.3791</td>
<td>12.6078</td>
</tr>
<tr>
<td>TR-54005</td>
<td>0.4374</td>
<td>0.1401</td>
<td>3.6066</td>
<td>14.5380</td>
</tr>
<tr>
<td>TR-138032</td>
<td>0.6366</td>
<td>0.2397</td>
<td>2.7509</td>
<td>19.8574</td>
</tr>
<tr>
<td>TE-210088</td>
<td>0.3725</td>
<td>0.0908</td>
<td>3.5235</td>
<td>19.3691</td>
</tr>
</tbody>
</table>

The flooding algorithm yields, as expected, an over-segmented image, and that the merging criterion is sufficiently robust. The merging risk criterion based on textureness appears to be useful for avoiding under-segmentation, as shown by the BDE measure in the last row of Table V which is severely deteriorated, while the VoI index is improved.

We can conclude that the classification stage of our algorithm gives better mean performance scores for all considered indices. In the comparison are included the following algorithms: Felzenszwalb and Huttenlocher (FH) [6], Normalized Cuts [5], Mean-Shift [34], CTM [9] and Modal Image Segmentation [10]. For these algorithms we report here the indices as given in the literature.

We have implemented our segmentation algorithm in Matlab, with the flooding algorithm in C. On a workstation at 2.66 GHz with 2GB RAM the average time for segmenting an image of 214 x 320 pixels was 13.2 sec.

We have tested extensively the robustness of the proposed algorithm in respect of different parameter values. In Table IV are given the performance metrics concerning parameters which are used in the feature extraction process: block size, number of classes, tree equipartition stopping criterion and textureness test. We can conclude from the experimental results that:

- the textureness test improves slightly the performance and the efficiency;
- the maximum number of tree clusters does not influence the result, if it is at least \(0.25N_B\);
- the best maximum number of classes being 11 for the whole data set, increasing it to one more deteriorates some indices about 1%,
- increasing the block size in pixels by 125% deteriorates some indices about 1.5%.

We can conclude that the classification stage of our algorithm has good robustness properties. The Tree Equipartition algorithm, which is the first step in block clustering, contributes to gain about 20% in efficiency and up to 3% in performance.

In Table V are given the performance metrics concerning two parameters of the merging criterion given in Equation (9): \(\nu_0\) and \(r\). At first we can see that the flooding algorithm yields, as expected, an over-segmented image, and that the merging process has interesting behaviour. As the regularization factor \(\nu_0\) increases from 4 to 7, the Variation of Information is improved. In addition all the indices have small difference from the best one, and we can conclude that the merging criterion is sufficiently robust.

VII. CONCLUSION

A generic unsupervised feature classification method and a new image segmentation framework have been proposed.
The framework involves feature extraction and clustering in the feature space, followed by region growing and merging in the spatial domain, based on the computed features of classes. A new block-based unsupervised clustering method is introduced which ensures spatial coherence using an efficient hierarchical tree equipartition algorithm. Therefore the description of appearance of the classes and the extraction of relevant features explicitly take into account the existing image regions. Topology plays also important role in initializing the regions, because it is based on the connected components of the maximum likelihood map of the described classes.

The second strong point of our approach consists on establishing topological constraints. We have adopted a Bayesian framework for attributing a height per pixel and per label. Finally, a new region growing algorithm is introduced, which is named Priority Multi-Class Flooding Algorithm, because it is similar to the flooding process used in the watershed localization. We have also proved that the growing algorithm for optimizing the labelling is equivalent to the minimum spanning tree construction. The third contribution of this work is a new algorithm for iteratively merging regions. Our algorithm takes into account not only the similarity of regions, but also the gradient image and the edge map.

Finally, we propose an unsupervised segmentation algorithm which combines edge features, region features and topological constraints for obtaining a visual grouping conform to the principles of proximity, similarity and continuation. Segmentation results on the Berkeley segmentation data set are presented and demonstrate the interesting properties of our approach.

**APPENDIX I**

**BHATTACHARYYA DISTANCE FOR GAMMA DISTRIBUTION**

Let us denote the probability density function for Gamma distribution as follows

\[ p(x) = \frac{x^{a_i}e^{-\frac{x}{b_i}}}{b_i^{a_i+1}\Gamma(a_i+1)} \]

The Bhattacharyya distance between two distributions with parameters \((a_i, b_i)\) and \((a_j, b_j)\) is

\[
B(p_i, p_j) = -\ln \int_0^\infty \frac{x^{a_i}e^{-\frac{x}{b_i}}}{b_i^{a_i+1}} \cdot \frac{x^{a_j}e^{-\frac{x}{b_j}}}{b_j^{a_j+1}} dx
\]

\[
= -\ln \int_0^\infty \frac{x^{a_i+a_j}e^{-\frac{x(b_i+b_j)}{b_i b_j}}}{b_i^{a_i+1} b_j^{a_j+1}} \sqrt{\Gamma(a_i+1)\Gamma(a_j+1)} dx
\]

\[
= -\ln \frac{\Gamma\left(\frac{a_i+a_j}{2}\right) \left(\frac{2b_i b_j}{b_i+b_j}\right)^{\frac{a_i+a_j}{2}+1}}{b_i^{a_i+1} b_j^{a_j+1}} \sqrt{\Gamma(a_i+1)\Gamma(a_j+1)} (a_i + 1)
\]

\[
= \frac{1}{2} \ln(\Gamma(a_i+1)\Gamma(a_j+1)) - \ln\left(\frac{a_i+a_j}{2}+1\right) + \frac{a_i+a_j}{2} \ln b_i + \frac{a_j}{2} \ln b_j - \frac{a_i}{2} \ln b_i - \frac{a_j}{2} \ln b_j.
\]

**APPENDIX II**

**CLUSTERING INITIALIZATION**

The initial clustering is obtained by the successive selection of representative blocks using the method proposed by Kaufman and Rousseeuw [24]. The first representative block is the median block in the feature space, that is the block which minimizes the sum of distances to all the others. The rest of representative blocks are selected according to the heuristic rule of choosing the blocks that are dissimilar to already selected representative blocks and that promise to have near them a higher number of blocks. The block indexed by \(i\) is selected, if it minimizes the following criterion:

\[
\sum_j \max_r \left(\min_r D(p_r, p_j) - D(p_i, p_j), 0\right),
\]

where the Bhattacharyya distance is used and \(r\) indicates an already selected representative block, while \(j\) is the index of a non-selected block.
initially unlabelled pixels that are inserted in
the number of quantization intervals and
\( \phi \) where \( N \) to a constant factor
(normal priority lists are empty too, the items of lists
\( P L \leq \) and quantize only that interval.
[min the unlabelled pixels are handled by the normal priority lis
ts. priority thresholds may not be used, where, in the later case,
that.
In the current implementation,
k is assumed as safe pixel, if,
neighborhood of \( z \) tests the local topology of decisions in the
neighborhood of \( z \), at the time that \( z \) is assigned to label \( k \).
In the current implementation, \( z \) is assumed as safe pixel, if,
and only if, it adjoins labelled pixels of label \( k \) and only of
that.
Depending on the application, one of the two or both
priority thresholds may not be used, where, in the later case, all
the unlabelled pixels are handled by the normal priority lists.
Furthermore, if \( \{ \min \{ d_i \}, \max \{ d_i \} \} \) is an interval of interest in
the domain of \( d_i \), we may set \( T_{\ell} = \min \{ d_i \}, T_{\ell} = \max \{ d_i \} \)
and quantize only that interval.
The computational cost of the proposed algorithm is (subject
to a constant factor
\[ T = ((1 - \phi)I_Q + \phi)N_U, \]
where \( N_U \) is the number of initially unlabelled pixels, \( I_Q \) is
the number of quantization intervals and \( \phi \) is the percentage of
initially unlabelled pixels that are inserted in low/high priority
lists. From Equation (10) \( I_Q \) is a function of \( \Delta_Q \) and since \( I_Q \)
is kept fixed, the computational time \( T \) of the growing process
becomes a linear function of \( N_U \) only.

APPENDIX III
IMPLEMENTATION OF PMCFA USING PRIORITY LISTS

Contour pixels \( s \) of classes \( l \) that satisfy inequalities \( d_i(s) < T_{\ell}^1 \), \( 1 \leq l \leq L \), are inserted in high priority simply
connected lists \( (PL_{l,0}) \) once they are scanned. Similarly, pixels
\( s \) for whom \( d_i(s) > T_{\ell}^2 \), are inserted in low priority simply
connected lists \( (PL_{l,q+1}) \). By contrary, the rest of contour pixels are inserted in normal priority simply connected lists,
denoted as \( PL_{l,i} \) \( (1 \leq l \leq L, 1 \leq i \leq q) \), corresponding to the \( i^{th} \) among \( q \) quantization intervals of dissimilarity criteria values.
The number of intervals is predefined and the index is
obtained using the estimated quantization step
\[
\Delta_Q = \frac{\max \{ d_i \} - \min \{ d_i \} }{q}, \\
i = Q(d_i) = \left[ \frac{d_i - \min \{ d_i \} }{\Delta_Q} \right] (10)
\]
In that way, an array of simply connected lists is enough to
keep the dissimilarity information of pixels, resulting in a
sorting procedure, which resembles the non-comparison sorting
algorithms (Chapter 8 of [19]). The quantization intervals as well as the priority decision thresholds are depicted graphically in Fig. 7.

At each step, first the items of \( PL_{l,0} \) for \( 1 \leq l \leq L \) (if any)
are popped and assigned to the corresponding label. Otherwise,
if high priority lists are empty, the items of lists \( PL_{l,i} \) \( (1 \leq l \leq L, 1 \leq i \leq q) \) of minimum \( i \) are popped and get labelled. Last, if
normal priority lists are empty too, the items of lists \( PL_{l,q+1} \)
are examined and processed by the algorithm.

Then, for each safe, popped and labelled pixel \( z \), its
unlabelled neighbors are considered and are inserted in the
corresponding list, if they are not already in one of them.
Safety of \( z \) tests the local topology of decisions in the
neighborhood of \( z \), at the time that \( z \) is assigned to label \( k \).
In the current implementation, \( z \) is assumed as safe pixel, if,
and only if, it adjoins labelled pixels of label \( k \) and only of
that.

Fig. 7. Quantization of dissimilarity metric and low/high priority thresholds.

APPENDIX IV
GRADIENT DISTRIBUTION

The gradient is computed independently for the three image components. The gradient magnitude is obtained as the square root of the sum of squares of the components. We use a new thresholding technique for edge detection. We consider the distribution of the magnitude as given by its sorted values. We seek then the percentage where the error of fitting the distribution using two linear segments is minimal. The threshold obtained by this technique is taken as the high threshold in the hysteresis method of Canny edge detector. The low threshold is fixed at 0.4 times the high. The map of detected edges is used for estimating the distance of any region boundary from the detected edges. Let \( d_e(i,j) \) be the mean distance of the boundary points, then the probability for the boundary could be measured by
\[
Pr\{B(i,j)\} = e^{-\alpha d_e(i,j)}.
\]
The parameter \( \alpha \) depends on the edge detection thresholds. It depends on the detection rate and and on the localization accuracy. As it is difficult to have a reliable estimate of this parameter for each image, and for more robustness, we take a constant value \( \alpha = 0.1 \).

Gamma distributions are assumed for gradient magnitude for both hypotheses: near edges and homogeneous areas. A simple method of fitting the two distributions is adopted. For the near edge model we use the data above the high threshold defined in the previous paragraph, while the rest of magnitude data are used for describing homogeneous areas. An illustration of the results for image TR-314016 is given in Figure 8.

ACKNOWLEDGEMENT

The work of Costas Panagiotakakis has been supported by postdoctoral scholarship (2009-10) from the Greek State Scholarships Foundation (I.K.Y.). The whole work is also partially funded by 03ED417 research project, implemented within the framework of the “Reinforcement Programme of Human Research Manpower” (PENED) and co-financed by National and Community Funds. We thank the anonymous reviewers for their helpful comments and suggestions.
REFERENCES