Hierarchical Multiple Markov Chain Model for Unsupervised Texture Segmentation

G. Scarpa, R. Gaetano, M. Haindl, J. Zerubia

Hierarchical Multiple Markov Chain Model for Unsupervised Texture Segmentation

Giuseppe Scarpa, Raffaele Gaetano, Michal Haindl and Josiane Zerubia, *Fellow, IEEE*

Abstract

In this work, we present a novel multiscale texture model, and a related algorithm for the unsupervised segmentation of color images.

Elementary textures are characterized by their spatial interactions with neighboring regions along selected directions. Such interactions are modeled in turn by means of a set of Markov chains, one for each direction, whose parameters are collected in a feature vector that synthetically describes the texture. Based on the feature vectors, the texture are then recursively merged, giving rise to larger and more complex textures, which appear at different scales of observation: accordingly, the model is named *Hierarchical Multiple Markov Chain* (H-MMC).

The *Texture Fragmentation and Reconstruction* (TFR) algorithm, addresses the unsupervised segmentation problem based on the H-MMC model. The “fragmentation” step allows one to find the elementary textures of the model, while the “reconstruction” step defines the hierarchical image segmentation based on a probabilistic measure (texture score) which takes into account both region scale and inter-region interactions.

The performance of the proposed method was assessed through the Prague segmentation benchmark, based on mosaics of real natural textures, and also tested on real-world natural and remote sensing images.

G. Scarpa is Assistant Professor at the University “Federico II”, DIET, via Claudio 21, 80125, Naples (I). Tel: +390817683768. Fax: +390815934448. Email: giscarpa@unina.it

R.Gaetano is PhD Student at the University “Federico II”, DIET, via Claudio 21, 80125, Naples (I). Tel: +390817683837. Fax: +390815934448. Email: raffaele.gaetano@unina.it

M. Haindl is head of the PR Department of ÚTIA, Czech Academy of Sciences, Pod vodarenskou vezi 4, 182 08 Prague 8 (CZ). Tel: +420266052350. Fax: +420284683031. Email: haindl@utia.cas.cz

J. Zerubia is head of the ARIANA research team of INRIA-I3S, 2004 route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (F). Tel: +33492387865. Fax: +33492387643. Email: josiane.zerubia@sophia.inria.fr

I. Introduction

Image segmentation is a low-level processing of critical importance for many applications in such diverse domains as medical imaging, security, remote sensing, industrial automation, and many others. Although it has been widely studied in recent decades, in many cases it still remains an open problem, as is the case of textured images where the spatial interactions may cover long ranges, asking for complex high order modeling. The situation is especially critical in the unsupervised case since no prior information is given and the process is completely blind.

It is widely recognized that a visual texture, which humans can easily perceive, is very difficult to define [17]. The difficulty results mainly from the fact that different people can define textures in application-dependent ways or with different perceptual motivations, and there is no generally agreed-upon definition [44]. It is not our intention to add here a new one: we simply observe that it should be as general as possible, because a too strict definition would allow one to confine his/her work to images that better fit with it, eventually leading to narrow-domain solutions.

Less subjective, instead, are certain categorizations made for “elementary” textures, like *structured* vs. *non-structured* textures, and *micro*- vs. *macro*-textures. The former classification arises from the nature (deterministic or stochastic, respectively) of a possible model generating the texture. The latter refers to the spatial correlation scale of the texture, which spans a continuous range whose extremes are micro- and macro-textures. Natural textures, however, are rarely homogeneous to be considered belonging to one category or another, as it may happen that a single texture can be regarded as composition of different textures based on the resolution. In those cases we will generally speak of “complex” textures.

In current literature, the matter of texture segmentation is mostly regarded as the composition of two different problems: on one side, the choice of a proper representation of textures, in order to establishwhat is to be identified, and on the other side the definition of a framework and strategy for the actual segmentation. Of course, though an effective separation of the problem is realized in many cases, in general the two tasks are not treated independently, since the second can be strongly dependent from the first.

Due to the aforementioned multiplicity of possible definitions, the problem of determining an efficient representation for textures can be treated according to a wide variety of different approaches, from the extraction of basic or complex features to the construction of a proper image model.

A quite classical example is the use of statistical features, for example in the form of co-occurrence matrices [13], [23], introduced in the pioneering work of Haralick [23]. These matrices account for co-occurring colors in pairs of image sites whose relative positions are fixed by choosing a distance and the orientation, which eventually parameterize the matrices. The discriminative potential of co-occurrence matrices is higher when a few assumptions can be made about the directionality, the spatial interaction scale and the color content of the textures involved, in order to avoid the otherwise complex selection of the proper matrices to use.

A more complex feature extraction approach can take into account the use of geometrical features, as presented in some works centered on fractal dimension [11], [47]. In these cases, the choice of fractal geometry is motivated by the observation that the fractal dimension is relatively insensitive to image scaling, and shows a strong correlation with human judgment of surface roughness. Fractal features are sometimes not very effective for texture analysis because they may not represent sufficient texture discriminatory information.

At present, most of the literature about texture representation via feature extraction relies on method based on signal processing [10], [19], with Gabor [13], [24], [35] and wavelet [26], [45] filters being by far the most used to enhance textural properties. The success of Gabor filters is mainly due to their outstanding properties of optimal joint resolution in the space/spatial-frequency domain [35] as well as orientation and frequency selectivity. The main drawback of Gabor filtering is the excessive computational effort to pay due to the large number of filters that can be selected by varying spatial scale, carrier frequency and orientation, that causes a strong parameterization. Wavelet-based methods have received a great deal of attention in recent years [10], [26], [45] due to several appealing properties, like their multi-scale definition and flexibility in the choice of the basis functions, that considerably help the tasks of texture classification and discrimination. However the adaptivity of the filtering w.r.t. the application domain is still an open issue and this somehow limits the applicability of wavelet methods in unsupervised contexts.

A different, yet very popular, approach to texture representation considers the use of a suitable texture model [1], [20], [21], [27], [37]. Markov Random Fields (MRF) models [1], [27], [36] are very popular due to their appealing theory: the Hammersley-Clifford theorem [4] relates the local MRF characteristics to the global distribution, allowing the definition of a global model through the local characteristics. Resulting robustness to noise is another qualifying point of this approach. Models that proved to work very good on non-textured images are widespread in literature, as [4], [36], [46] just to cite a few, but due to their locality they usually fail in capturing long range interactions, occurring very intensively in images with structured, near-regular and/or macroscopic textures [1], [27]. For this reason, more complex causal models like multi-resolution Hierarchical MRFs [5], [27] (where the Markov property applies causally through the different resolution levels) or two-dimensional causal autoregressive models [21], [37], are often preferred, at the price of a generally higher computational complexity and/or an increased difficulty in constructing the model and managing its parameters.

Concerning the actual segmentation methods arising from the chosen texture representation framework, it is reasonable to refer to the classical image segmentation literature, considering the numerous techniques belonging to the *edge-based* and the *region-based* families. For the first category, some interesting variational techniques for texture segmentation that rely on boundary detection have been proposed recently [33], [38], [7], [6], where boundaries among textures are retrieved using curve evolution driven by some energy minimization criterion. Major drawbacks of these methods are the sensitivity to initial conditions and, in particular for textures, the difficulty to correctly locate boundaries of structured and macro-textured areas.

In the region-based framework, besides the well known optimization procedures associated to MRF-based modeling like in [5], [27], usually heavy in terms of computational complexity, some region growing techniques have been recently proposed for the texture segmentation problem that are typically based on the split-and-merge paradigm, like for example in [16] where image is first decomposed by means of spectral and spatial clustering and then the resulting elementary regions are used as seeds for a region growing process. Finally, some result on texture segmentation has been presented also using graph-cuts methods over a suitably chosen textural feature space [42], [14], where no specific modification is proposed in terms of optimization procedure to deal with textures, especially in the structured and macro-textured case.

The solution presented here, relying on a model-based texture representation, starts from two main observations. First, a pixel-level texture description, no matter which model is used, is very limited when the object image contains macro textural features, i.e. large textons [48]. The use of multiple scales [2], [19] is certainly a first step to mitigate this problem, but an additional gain can be achieved if one moves to a region-level description, where textons can be handled as atomic components. Second, in unsupervised segmentation the cluster validation is very often an ill-posed problem and the only reasonable solution is a hierarchical segmentation [2], [24], [30] (sequence of nested segmentations) where the number of texture segments is not explicitly singled out.

The proposed *Texture Fragmentation and Reconstruction* (TFR) algorithm, whose preliminary study we presented in [39], [40], [41], follows the paradigm of splitting and merging where a first (over-)segmentation step provides the elementary regions that are processed (essentially merged) in the subsequent step. The TFR algorithm is based on a hierarchical region-level description, where inter-region interactions are modeled through simultaneous Markov chains whose states are recursively merged according to their mutual interaction, providing the desired hierarchical texture segmentation. A similar approach can be found in [30].

The experiments carried out on the Prague benchmark [32] data set allow a comparison with other methods [9], [12], [16], [19], [20], [21], [22] using the same benchmark, and prove the potential of the proposed technique which has been also successfully applied to many natural images from the Berkeley Segmentation Dataset [29] and a remotely sensed image.

In next Section the proposed texture modeling is presented, while Section III deals with the TFR segmentation algorithm. The experiments on the Prague Benchmark are discussed in Section IV, while the applications to real word images are shown in Section V, and finally Section VI draws conclusions and outlines future research.

II. HIERARCHICAL TEXTURE MODELING

A complex scenario can be usually segmented in different, equally reasonable, ways depending on the scale of observation. As an example, consider the front of a building with an array of windows. At a very fine scale one is likely to distinguish the *glasses*, the *frames* of the windows, and the *walls*. Then, at a coarser scale, frames and glasses can be considered as a unique texture (*window*), since they are strongly related spatially, while at the coarsest scale window and walls, which also relate to each other but with longer range spatial interactions, merge into the *building* texture. In other words, the cluster validation problem becomes an *ill-posed* problem, if the scale is not fixed somehow. The ill-positioning of the cluster validation problem is very common in many computer vision applications, and, in the case of the textures, it arises directly from their intrinsic multi-scale definition. Based on this observation, we propose here a method which provides a hierarchical segmentation, rather than a single segmentation with an estimated (somewhat unreliable) number of regions. By doing so, we get a scale-dependent interpretation of the image, represented by a set of nested segmentations which can be associated with a tree structure where each of its prunings corresponds to a possible segmentation.

In order to achieve this goal, we resort to a *hierarchical* and *discrete* modeling of the textures. To do this, a discretization in the color domain is therefore needed. Such a process is just a color partition applied either directly to the original image or, more generally, to a transformed image, like pixel-wise feature planes properly extracted from the original one.

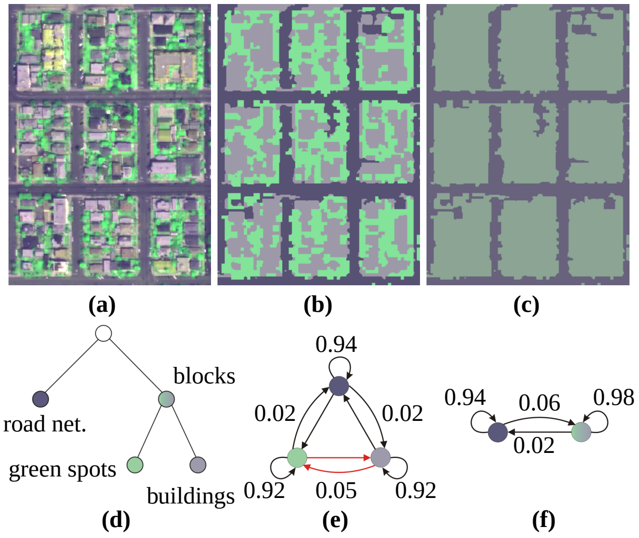


Fig. 1. H-MMC model: *urban area* sample (a); *3-state* (b) and *2-state* (c) maps; states hierarchy (d); *3-state* (e) and *2-state* (f) Markov chains for the north direction.

A. Hierarchical Multiple Markov Chain Model

The proposed modeling provides region-wise features which carry information about region shape and contextual region interaction.

The starting point for the construction of the image model is an appropriate image partition in which each segment corresponds to an "elementary texture", or simply "elementary state"[1], that will be a collection of connected regions which are close both in their color response and in their contextual model features (defined below) which account for region shape and interactions among neighboring regions. A complete hierarchical description of the image is then obtained by pairwise associating and merging together the so defined elementary states, implicitly providing a set of progressively coarser resolution textures, from the initial partition to the final single full-image state.

In order to detail the model, let us assume that an image partition in elementary states is available. Consider the eight main spatial directions (north, northeast, east, etc...) and for each of them focus on the pixel-wise state evolution along it. These processes can be modeled through *Multiple Markov Chains* (MMC). Fig.1 clarifies the idea on a simple (urban) texture (a). In (b) the partition in three states is shown while in (e) is represented a corresponding chain on a fixed direction (north). According to the

[1] "Texture" in the sense suggested by the proposed model. In the following, the terms state, texture or class are to be meant as interchangeable.

idea of hierarchical interpretation, the next step is the selection of two, out of three, states to merge. In this simple example it is easily justified, intuitively, the choice of *green spots* and *buildings*, see the 2-state map (c) and the hierarchy tree (d), which are spatially strongly related (how do we automatically address this issue will be explained later). After merging all chains will be reduced by one state, as graph (e) reduces to (f) for the northern direction, and the 3-state MMC reduce to a 2-state MMC as well. In general we would start from a L-state partition (corresponding to the finest scale texture segmentation) to reach a single global state (no segmentation at all) after $L-1$ merging steps, while collecting L MMC's corresponding to different scales.

The so obtained *Hierarchical* MMC (H-MMC) stack can be formally defined as follows. Let $\Omega^{(n)}$ be the state set at a given "scale" n (n is also the cardinality of $\Omega^{(n)}$), the transition probability matrix for any chain (direction) $j = 1, \ldots, 8$ (describing both intra- and inter-state transitions) is defined as $\mathbf{P}_j^{(n)} = \{p_j^{(n)}(\omega'|\omega) \,:\, \omega', \omega \in \Omega^{(n)}\}$ where

$$p_j^{(n)}(\omega'|\omega) \triangleq \Pr(x_{s+1} = \omega' | x_s = \omega, \text{ chain} = j) \tag{1}$$

$\forall \omega, \omega' \in \Omega^{(n)}$, x_s represents the state of a generic site $s \in \mathcal{S}$, and $s+1$ is the site next to s along direction j. These probabilities are easily estimated as

$$p_j^{(n)}(\omega'|\omega) = \frac{|\mathcal{S}_{\omega \xrightarrow{j} \omega'}|}{|\mathcal{S}_\omega|} \tag{2}$$

where $\mathcal{S}_\omega$ is the set of pixels labeled ω and $\mathcal{S}_{\omega \xrightarrow{j} \omega'} = \{s \in \mathcal{S}_\omega \,:\, s+1 \in \mathcal{S}_{\omega'}, \text{ chain} = j\}$. The H-MMC model is consequently associated with the transition probability set

$$\mathbf{P} = \{\mathbf{P}_j^{(n)} \,:\, 1 \leq j \leq 8, \, 1 \leq n \leq L\}, \tag{3}$$

and $\mathbf{P}^{(n)} = \{\mathbf{P}_j^{(n)} \,:\, 1 \leq j \leq 8\}$ is just the n-th MMC model component.

The transition probabilities indicated on the graphs (e)-(f) of Fig.1 give an idea of their relationship with the visual appearance of the texture. First, note that, for each fixed scale n, the *intra*-state transition probabilities of a given state account for the shape of its region components. As an example for the *road network* we expect rather large values for the north direction w.r.t. other directions. On the other hand, the remaining *inter*-state transition probabilities provide a statistical description of the context, that is the spatial interaction between states, accounting for the relative occurrence and mutual positioning of adjacent regions.

As the states are progressively coupled in a fine-to-coarse texture representation a sequence of state sets is generated: $\Omega^{(L)}, \Omega^{(L-1)}, ..., \Omega^{(1)}$. Observe that, once the transition probabilities are known at a given scale n of the process, they are also automatically obtained for the coarser level $n-1$ above and, eventually, if the hierarchy tree is given one has just to estimate these attributes at the finest level L. In fact, if we either denote with $(\omega_a, \omega_b) \in \Omega^{(n)} \times \Omega^{(n)}$ the couple of states whose merging generated $\omega \in \Omega^{(n-1)}$, i.e. $(\omega_a, \omega_b) \equiv \omega$, or just $(\omega_a, \omega_b) \equiv (\omega, \emptyset)$ when ω is not the merging state associated with step n, then by using the total probability law it can easily be shown that[2]

$$\begin{aligned} p(\omega'|\omega) &= \Pr(\omega'_a \cup \omega'_b | \omega_a \cup \omega_b) = \\ &\quad \frac{p(\omega_a)}{p(\omega)}[p(\omega'_a|\omega_a) + p(\omega'_b|\omega_a)] + \\ &\quad + \frac{p(\omega_b)}{p(\omega)}[p(\omega'_a|\omega_b) + p(\omega'_b|\omega_b)], \end{aligned} \tag{4}$$

where $p(\omega) = p(\omega_a) + p(\omega_b)$, and eventually any element of $\mathbf{P}_j^{(n-1)}$ can be obtained by a linear combination of elements of $\mathbf{P}_j^{(n)}$.

Thanks to the above-mentioned property, $\mathbf{P}^{(n)}$ does not need to be computed for each $n < L$, and the H-MMC model is completely specified by the triple $(\Omega^{(L)}, \mathbf{P}^{(L)}, \mathcal{T})$, where $\mathcal{T}$ is the binary hierarchy tree.[3]

Similarly, the MMC parameters of a given state (distributed on several unconnected regions) can be related to the parameters of the locally (to the single connected regions) defined MMCs through a simple weighted average (5). This property which is summarized below is very useful during the segmentation task, as it allows to characterize the image from the bottom starting with the featuring of single connected regions, or “fragments”.

Region-wise MMC features: Suppose that a region $\mathcal{S}_\omega \in \Omega^{(L)}$ associated with state ω is composed of N_ω fragments $\{\mathcal{S}_{\omega_k}\}_{k \in 1,\ldots,N_\omega}$, where ω_k is the substate of ω identifying the k-th fragment: $\omega = \bigcup_{k=1}^{N_\omega} \omega_k$. Therefore the total probability law yields

$$p_j^{(L)}(\omega'|\omega) = \sum_{k=1}^{N_\omega} p_j^{(L)}(\omega'|\omega_k) p(\omega_k), \tag{5}$$

which relates the global description of a texture to the region-wise features $p_j^{(L)}(\omega'|\omega_k)$ and $p(\omega_k)$ given by

$$p_j^{(L)}(\omega'|\omega_k) = \frac{|\mathcal{S}_{\omega_k \xrightarrow{j} \omega'}|}{|\mathcal{S}_{\omega_k}|} \triangleq A_{\omega_k}(\omega', j) \tag{6}$$

and $p(\omega_k) = |\mathcal{S}_{\omega_k}|/|\mathcal{S}|$, respectively. Eventually the $L \times 8$ feature matrix $A_{\omega_k}(\omega', j)$ defined in (6), which characterizes each fragment in terms of shape and context, can be used to carry a fragment-level clustering in order to define the initial states $\Omega^{(L)}$.

[2] We neglected indices j and n for the sake of simplicity.

[3] Hence, $\Omega^{(L)}$ is the set of terminals on $\mathcal{T}$, while for each $n < L$, $\Omega^{(n)}$ is the set of terminals of a pruning of $\mathcal{T}$.

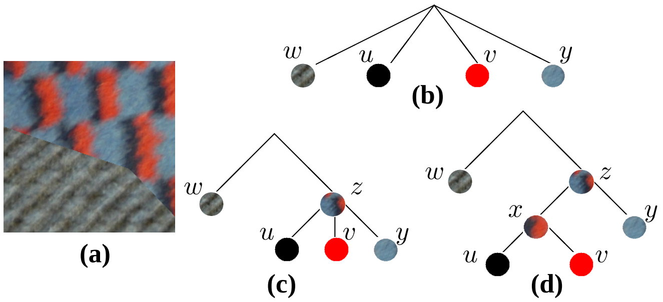


Fig. 2. Image structure ambiguity. A texture mosaic (a) and several binary (d) and non-binary (b)-(c) hierarchical trees.

B. The segmentation problem

Let us now turn to the segmentation problem. Since we are assuming an unsupervised context, we do not *a priori* know how many and what kind of textures may be found in the image to be segmented.

The determination of the number of textures of a given image, classically referred to as the *cluster validation problem*, is strictly related to that of finding the internal structure of each single texture. Indeed, according to the H-MMC modeling, a texture is nothing but a local visual property of a surface where the locality has to be meant at multiple spatial scales. This definition allows to describe complex textures but it also says that textures which seems distinct at fine spatial scale collapse in a single texture, sooner or later, at a coarser scale, even if their spatial interaction is weak. As a consequence the application of this model eventually allows us to circumvent the cluster validation problem, since it aims at recursively retrieving textures which cover larger and larger areas of the image until the whole image is associated with a single global texture. The final result is therefore a hierarchical segmentation map, that is a stack of nested segmentations varying for number of classes: the smaller the number of classes, the coarser the scale. In general evaluating the accuracy for such a product is quite difficult, but if one has data with ground-truth at a single scale, then he only has to seek for the best-fitting segmentation map contained into the stack for the comparison. The automatic recognition of the right scale (number of classes) is not object of this work but is something that in any case can be separately addressed in a subsequent step, possibly aware of the final application for which the segmentation is needed.

To better fix the above considerations let us discuss the example of Fig.2. The image (a) is composed by “two” textures represented as states w and z. According to the H-MMC modeling we must somehow relate progressively the elementary textures until we have a unique state representing the whole image. Assume without loss of generality that we start from only four elementary textures, denoted w, u, v,

y, easy to localize in the image. In (b)-(d) are depicted some possible choices for the model hierarchy which represent both intra- and inter-texture dependencies. A first observation is about the ill-positioning of the cluster validation problem. We said we have two textures, but actually a human observer could also guess there are four: *it depends on the application*[4] Therefore we can expect that such data will be even more confusing for a computer. The question is rather *how to correctly relate the fine textures in order for the hierarchical segmentation to contain both the 2- and the 4-class partition.*

To this end the structure (b) seems to be the worst since we jump directly from a 4-class partition to the 1-class one, by merging all 4 classes in one step. Structure (c) appears a more reasonable solution that contains both the desired partitions. However, if we better look at the data we realize that states u and v are strongly related and may be merged apart from y which only later on will be joined to form state z, as represented by *binary* structure (d). Although this is just a case, indeed there are two good motivations to restrict our attention to “binary” structures. The former is computational: we restrict our search when seeking the hierarchy tree. The latter is about the information conveyed by the hierarchical segmentation: a larger number of internal nodes (the maximum is achieved with binary structures) means more possible prunings and, therefore, a larger number of image interpretations/segmentations provided. For these reasons we only deal with binary hierarchies in the following.

III. The TFR segmentation algorithm

In the previous section we have introduced the H-MMC texture model and shown that it can be used for the task of hierarchical segmentation. We have also shown that such a model is completely defined by the triple $(\Omega^{(L)}, \mathbf{P}^{(L)}, \mathcal{T})$, and motivated the restriction on $\mathcal{T}$ to be a *binary* tree. Here we clarify how these three items are determined by the proposed *Texture Fragmentation and Reconstruction* (TFR) segmentation algorithm which follows the splitting-and-merging paradigm and whose general scheme is shown in Fig.3.

The proposed solution is quite simple. The first two blocks, CBC (*Color Based Clustering*) and SBC (*Spatial Based Clustering*), perform an over-partition of the image that provides the initial finest-scale texture states $\Omega^{(L)}$ which are therefore progressively related in the last merging process yielding the desired hierarchical segmentation with the associated tree structure $\mathcal{T}$.

Any finest resolution texture $\omega \in \Omega^{(L)}$ is a collection of image fragments homogeneous w.r.t. both their internal “visual appearance” (average color) and the contextual characteristics (shape and spatial

[4] For example, think about a region-based coding algorithm which would be more efficient on a 4-class partition.

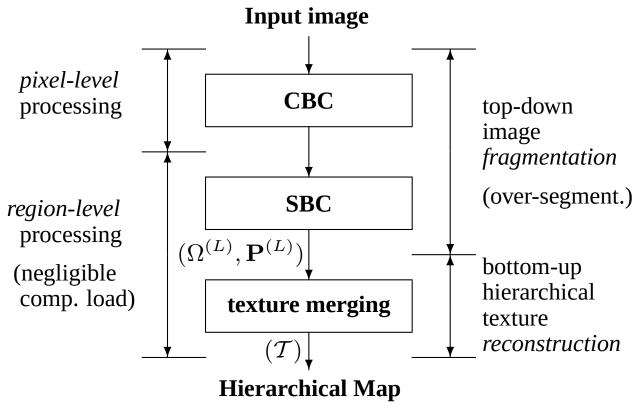


Fig. 3. TFR flow chart.

interaction with adjacent states) conveyed by the MMC feature set (Eq.6). In order to perform such a classification task, the first CBC block outputs a pixel-by-pixel “color” classification (see Sec.III-A) in K_c color states, also referred to as *partial* (MMC) states. At this level each group of adjacent pixels having a same label are assigned to an image “fragment” and all subsequent TFR processing is made considering fragments (rather than pixels) as atomic elements. All contours are therefore fixed in the CBC step, and later, in case, they can only disappear because of region merging. Each color state is therefore further split in K_s (full-defined) states by the SBC block (see Sec.III-B) which operates a clustering aimed at putting together fragments with similar MMC features (Eq.6). Therefore a total of $L = K_c \times K_s$ states are eventually defined.

Once the set of L initial finest texture states, $\Omega^{(L)}$, is completed, the last texture merging process (see Sec.III-C/D) can recursively retrieve textures at larger and larger scale.

In order to clarify the overall process an experiment is detailed in Fig.4. In (a) is the image to be segmented, whose K_c-color segmentation map (CBC output, $K_c = 24$) is shown in (b) in false colors. Given the complexity of the image, a partial CBC map (involving only 4 out of 24 color states) is shown in (c) for an easier interpretation of the subsequent SBC step (since $K_s = 12$, the complete SBC map would have $L = 288$ states!). The 4 color states are associated with different false colors: yellow, green and violet, spanning over two textures, and red, spanning over three textures. Focusing on these selected states it is now easy to recognize the effect of the SBC processing on each of them (d) and, in particular it should be evident that each of the 48 states shown in (d) practically never belong to more than one single texture, which is fundamental for the texture discrimination.

On the other hand, it is also worth to notice that although K_s was set much larger than the strictly

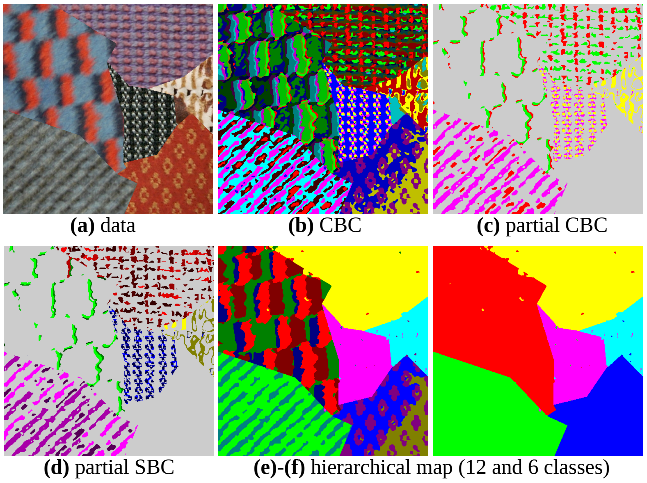


Fig. 4. TFR process evolution.

needed (the example shows that a value of 2 or 3, depending on the case, could suffice for the selected color states), the subsequent merging process (two snapshots of which are shown in (e)-(f)) is able to correctly rejoin over-split states at coarser levels. The same consideration holds for the over-split present at the CBC level as well. Nonetheless, it is also clear that there exists superior limits for K_c and K_s over which the states begin to be less significative and too much localized, so that the textures may result irreparably over-split.

Aware of this trade-off we have used heuristic rules to fix *a priori* both K_c and K_s (and hence $L = K_c K_s$), as to ensure a large (but not exceeding) number of states, L, in order to avoid *under*-segmentation which could not be recovered by the merging process. If we let M be either the number of textures expected in the image or its maximum value (depending on the information we have), on the basis of our experimental observations, we found $K_c = 2M$ to be a reasonable choice. This can be intuitively justified by the fact that any non-trivial texture has at least two modes in the color space. Hence, we are ensuring that, on average, we have at least two color states per texture. For K_s, instead, a good compromise is to fix it equal to M. This way, each color may occur simultaneously in each texture (but in one contextual configuration only) and the algorithm could keep working properly.

A. Color-based clustering (CBC)

The color segmentation task (CBC) is here achieved by means of the *tree-structured* MRF (TS-MRF) model-based algorithm presented in [15], [36] and briefly recalled in the following. This algorithm has several characteristics which are attractive in this context. It uses a MRF prior modeling which helps to regularize elementary regions, improving the robustness with respect to the noise. Moreover, a data likelihood description based on a multivariate Gaussian modeling helps to take into account the correlation in the color space. Finally, its tree structured formulation speeds up the processing, ensures convergence to the desired number of classes, and reduces large-scale effects thanks to its progressive localization.

A discrete random field X defined on a lattice $\mathcal{S}$ is said to be a MRF with respect to a given neighborhood system if the Markovian property holds for each site s. Moreover if a MRF is positive then its global distribution has a Gibbs form,

$$p(x|\theta) = \frac{1}{Z}\exp[-U(x,\theta)], \tag{7}$$

with $U(x,\theta) = \sum_{c \in C} V_c(x_c,\theta)$, where x is the realization of the field X, θ is the set of parameters of the model, the V_c functions are called potentials, U denotes the energy, Z is a normalizing constant that depends on θ, and c indicates a clique of the image. Note that each potential V_c depends only on the values taken on the clique sites $x_c = \{x_s, s \in c\}$ and, therefore, accounts only for local interactions. As a consequence, local dependencies in X can be easily modeled by defining suitable potentials $V_c(\cdot)$. In particular the second order Potts MRF model [4] is considered in this work, where only pairwise cliques are taken into account, that is:

$$V_c(x_c) = \begin{cases} \beta & \text{if } x_p \neq x_q, \quad p,q \in c \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where $\beta > 0$ is the model parameter.

Turning to the segmentation problem, such a MRF X can be used as prior for the desired segmentation map $\hat{x}$ according to the MAP criterion, that is, $\hat{x} = \arg\max_x p(y|x)p(x)$, once a likelihood model is assigned as well: we did the common assumption of conditional independence, $p(y|x) = \prod_{s \in \mathcal{S}} p(y_s|x_s)$, and multivariate Gaussian distribution for the likelihood of single pixels.

The inherent high complexity of this Bayesian formulation of the segmentation problem, indeed, is consistently reduced if the TS-MRF model is used since it allows a faster optimization procedure [15]. The TS-MRF model defines a K_c-label field X as a stack of $K_c - 1$ nested 2-label Potts MRFs (8). The *root* MRF serves for splitting the image in two classes. Then, *local* binary MRFs are associated with each of classes singled out in order to further split the image. Such process goes on recursively until a suitable condition is met for each of the current classes and, if $K_c - 1$ binary splits have been accepted, a K_c-class segmentation is provided. In this work, the condition we used to decide whether to proceed in splitting or not a given class was simply that the desired (*a priori* fixed) number of classes K_c has not yet reached and that its split would provide the largest decrease (w.r.t. other current candidate splits) of overall distortion when fitting its data with two local likelihoods instead of one.

B. Spatial-based clustering (SBC)

The color segmentation provided by CBC is passed to the spatial-based clustering (SBC module) which further splits each of the color states in order to generate the state set $\Omega^{(L)}$, where each $\omega \in \Omega^{(L)}$ is associated with a cluster of fragments $\{\omega_k\}$ which are therefore similar (the color has been already taken into account) also w.r.t. the contextual information carried by the MMC features $A_{\omega_k}(\omega', j)$, with $\omega' \in \Omega^{(L)}$, defined in Eq.6.

In principle, a joint estimation of $\Omega^{(L)}$ and $\mathbf{P}^{(L)}$ should be provided, for example by means of some iterative procedure which starts from an initial state set and alternates the computation of $\mathbf{P}^{(L)}$ and $\Omega^{(L)}$ until convergence. We have tested this solution, but the results were not satisfying because of two main reasons: a) the *curse of dimensionality* ($L \times 8$) into the feature space, since L is definitively too large (in our setting $L = K_c K_s = 2M^2 = 288$, if $M = 12$); b) the instability of the iterative process.

For the above reasons we decided to consider a simpler solution, where the color state set $\Gamma^{(K_c)}$ computed in CBC is used in place of $\Omega^{(L)}$ to provide the needed fragment level characterization. Hence, each color state $\omega \in \Gamma^{(K_c)}$ is independently further split, generating K_s offspring states of $\Omega^{(L)}$, as follows. For each of the N_ω fragments labeled ω, say the k-th, the corresponding A_{ω_k}, $k \in \{1, \ldots, N_\omega\}$, is computed by Eq.6 on the reduced state set $\Gamma^{(K_c)}$. Once the probabilities $A_{\omega_k}(\omega', j) = p_j^{(K_c)}(\omega'|\omega_k)$ are computed, we convert them in the following features, which we found experimentally more effective:

$$F_{\omega_k}(\omega', j) \triangleq \begin{cases} \log[1 - p_j^{(K_c)}(\omega'|\omega_k)], & \omega' = \omega \\ \log[\frac{p_j^{(K_c)}(\omega'|\omega_k)}{(1 - p_j^{(K_c)}(\omega|\omega_k))}], & \omega' \neq \omega. \end{cases} \tag{9}$$

Behind this solution there are two reasons. Since the original probabilities have quite different dynamics, while being all equally important for the clustering, the logarithm helps to have more uniform dynamics. Moveover, the normalization in the second row of (9) and the log operation help reducing the dependency on the scale, emphasizing the importance of the context.

Finally, before performing the clustering in such a feature space, a feature reduction via PCA is performed since the dimensionality of that space ($K_c \times 8$) is still too large for a reliable clustering. In particular, this task has been split in two steps. A first PCA, retaining only the first component, is applied independently for each fixed row ω' of $F_{\omega_k}(\omega', j)$, as to obtain a dimensionality reduction factor 8. Then, the resulting L-dimensional feature set is further reduced by means of a PCA which retains a number of meaningful components such that the 75% of the energy is kept (the same rule is used for each of the color state to be split).

Based on these (fragment-wise) features, each color state is therefore split by clustering its fragments by means of a simple k-means algorithm.

C. *Region merging: the* texture score

The result of the sequence of steps described above (CBC and SBC) is a partition of the image in regions corresponding to the finest-scale textures, collected as $\Omega^{(L)}$[5]. According to the H-MMC model formulated above, these terminal states have now to be related until all collapse in the macro state associated with the hierarchy root, i.e. with the whole image (coarsest scale), which corresponds to a recursive region merging. The aim of this process is to collect together finer textures in order to get larger and larger (in scale) textures and provide a nested hierarchical texture segmentation.

Since the merging process goes always on until all nodes collapse in the tree root, what we need is a tool that indicates, at each step, which couple of nodes must be merged, that is to say, which classes are most likely to belong to the same texture. In doing this, we should encourage the merging of strongly interacting classes, as they are likely to belong to the same textured area, and take into account short-range interactions before long-range ones. To fix the problem, let us come back to the example of Fig.2 and suppose we have currently four states, u, v, y and w, two of which should be selected for merging. As already discussed structure (d) would be preferable, and so the merging of u and v would move in that direction. Moreover, we observe that u (corresponding to the black regions) is the current smallest scale texture (this makes u a good candidate), and is "spatially" strongly interacting with v.

Based on these considerations for each terminal class ω we define a synthetic parameter called "Texture Score" [6]

$$\mathrm{TS}^{\omega} = \frac{p(\omega)}{\max_{\omega' \neq \omega} p(\omega'|\omega)}, \tag{10}$$

and for each step $n = L, L-1, \ldots, 2$, the state with smallest score and its "dominant neighbor" are merged, so as to move from $\Omega^{(n)}$ to $\Omega^{(n-1)}$.

The Texture Score measures the "completeness" of a texture, based on its spatial scale and the interactions with neighboring classes: incomplete classes (small TS) will be merged first, so as to obtain complex textures that are more and more self-consistent (large TS).

[5] Now L is no longer just the number of colors given by CBC but it has increased because of the splitting of each color-state by SBC.

[6] Originally called "Region Gain" in [39].

To understand why the TS measures completeness, let us rewrite it as the product of three terms

$$\mathrm{TS}^{\omega} = p(\omega) \cdot \frac{1}{p(\bar{\omega}|\omega)} \cdot \frac{p(\bar{\omega}|\omega)}{\max_{\omega' \neq \omega} p(\omega'|\omega)}, \tag{11}$$

where $p(\bar{\omega}|\omega) = 1 - p(\omega|\omega)$ is the probability of leaving state ω in any direction. Such terms take into account, respectively, the size of class ω, its compactness, and the presence of a dominant neighboring class. Classes with very small TS are typically small (small $p(\omega)$), dispersed over a large number of even smaller fragments (large $p(\bar{\omega}|\omega)$), and with a single dominant neighbor ($\max_{\omega' \neq \omega} p(\omega'|\omega) \simeq p(\bar{\omega}|\omega)$), that is, texture fragments that should be merged with some larger neighbors. On the contrary, a large, compact class, with no dominant neighbor, and hence a large TS, is probably a complete texture that should be considered for merging only in the last steps of the process. Notice also that the product of the first two terms is an indicator of the spatial scale of the class, while the third one measures the interaction between the class and its dominant neighbor.

Therefore, at each step of the merging process, the class $\widehat{\omega}$ with the smallest score is merged with its dominant neighbor ω^*, singled out as

$$\omega^* = \arg\max_{\omega \neq \widehat{\omega}} p(\omega|\widehat{\omega}) \tag{12}$$

Transition probability matrices and scores are then computed for the merged classes and their neighbors (a task of negligible complexity, since it is carried out at the class-level with no pixel-wise computation) and the process goes on recursively until a single node is reached.

Once the complete sequence of merging is defined, a nested hierarchical segmentation is obtained. Therefore, the user can select the segmentation that better serves his/her current needs. To this end a simple rule for selecting the pruning was suggested in [39] which refers directly to the spatial scale of the classes by defining a suitable threshold for the texture score.

D. Enhanced texture score

The texture score defined above measures how likely a region corresponds to a texture w.r.t. the hypothesis that it is just a part of a larger one. When the score is small we let the region be absorbed from the dominant neighbor, the one that shares the largest boundary with the given region. Although in the most cases this criterion provides satisfactory results, there are other ones where it fails. In fact, the presence of noise may increase the length of the boundary between two regions and make them “closer” according to the score definition. This problem often occurs because of the boundary fragmentation phenomena caused by color quantization during the CBC step.

In order to reinforce the measure and to improve the robustness, we considered not only the degree of contact between regions but also their spatial distribution similarity. To do so we have introduced an additional term in the score, which is the Kullback-Leibler divergence (KLD) between the spatial location distributions of the regions to be compared. The KLD between two distributions, p and q, is defined as:

$$D(p\|q) \triangleq E_p\left[\log\frac{p(\mathbf{x})}{q(\mathbf{x})}\right] = \int p(\mathbf{x})\log\frac{p(\mathbf{x})}{q(\mathbf{x})}d\mathbf{x}, \tag{13}$$

where $E_p[\cdot]$ is the statistical average according to the distribution p. Since $D(p\|q)$ is the average log-likelihood ratio between p and q, it is a measure of the inefficiency of assuming q in place of p. Hence it is well adapted to describe how close two objects are w.r.t. their spatial locations. In particular, named $q_\omega(\mathbf{x})$ the distribution of the spatial location of state ω, where $\mathbf{x}$ is the 2-D spatial position, then the modified texture score $\mathrm{TS}^{\omega}_{\mathrm{KL}}$ of state ω is defined by

$$\log \mathrm{TS}^{\omega}_{\mathrm{KL}} \triangleq \min_{\omega'\neq\omega}\left\{\log\frac{p(\omega)}{p(\omega'|\omega)} + D(q_\omega\|q_{\omega'})\right\}, \tag{14}$$

where we refer to the logarithmic formulation to properly combine the previous score with the KLD term. Notice that by removing the KLD term the score reduces to the original one.

The computation of the KLD is in general quite difficult for most of the distributions, and admits a closed form only in a few cases. One such case is that of two Gaussian distributions p and q for which the divergence $D(p\|q)$ is given by [34]:

$$D(p\|q) = \frac{1}{2}(\log\frac{|\Sigma_q|}{|\Sigma_p|} + \mathrm{tr}(\Sigma_q^{-1}\Sigma_p) + (\mu_p-\mu_q)^T\Sigma_q^{-1}(\mu_p-\mu_q) - d) \tag{15}$$

where $p \sim \mathcal{N}(\mu_p, \Sigma_p)$, $q \sim \mathcal{N}(\mu_q, \Sigma_q)$ and $d = 2$ is the distribution dimensionality. Due to its simplicity, the above modeling has been considered here.

IV. Experimenting with the Prague Benchmark

The Prague segmentation benchmark [32], developed by UTIA Institute of the Czech Academy of Sciences, has a two fold objective: to mutually compare and rank different texture segmenters and to support the development of new segmentation and classification methods.

The benchmark server provides a comparative analysis of all the results uploaded by users according to several accuracy indicators (see [25], [29], [32] for additional details) which are grouped in the three following categories.

- **Region-based criteria:** CS, correct (region) detection; OS, over-segmentation; US, under-segmentation; ME, missed regions; NE, noise region.

- **Pixel-wise criteria:** O, omission error; C, commission error; CA, class accuracy; CO, recall; CC, precision; I, type I error; II, type II error; EA, mean class accuracy estimate; MS, mapping score; RM, root mean square proportion estimation error; CI, comparison index.
- **Consistency measures:** GCE and LCE, global and local consistency error, respectively.

A. Reference segmentation algorithms

The different algorithms which have been run on the same benchmark data sets are listed and briefly described below:

1) GMRF/EM (Gaussian MRF model with EM) [20]: Single decorrelated monospectral texture factors are assumed to be represented by a set of local Gaussian Markov random field (GMRF) models, each centered on a pixel and limited by a sliding window of fixed size. The segmentation algorithm, based on the underlying Gaussian mixture (GM) model, operates in the decorrelated GMRF space of parameters. The algorithm starts with an over-segmented initial estimation which is adaptively modified until the optimal number of homogeneous texture segments is reached.

2) AR3D/EM (3-D Auto Regressive model with EM) [22]: This algorithm is similar to the previous one, but the GMRF model is replaced by a 3-D auto-regressive model, thus spectral space correlations can be modeled without approximating the spectral information.

3) JSEG [16]: The method consists of two independent steps, color quantization and spatial segmentation. In the first step, colors in the image are quantized to several representative classes that can be used to differentiate regions in the image. The image pixels are then replaced by their corresponding color class labels, thus forming a class-map of the image. The subsequent spatial segmentation step applies to the class-map, so as to obtain the so-called “J-image”, where high and low values correspond to likely boundaries and interiors, respectively, of color-texture regions. A region growing method is then used to provide the final segmentation on the basis of a multi-scale J-images.

4) SWA (Segmentation by Weighted Aggregation) [19]: The SWA algorithm uses a bottom-up aggregation framework that combines structural characteristics of texture elements with filter responses. The texture shapes are adaptively identified and characterized by their size, aspect ratio, orientation, brightness, etc. Then, various statistics of these properties are used to discriminate the different textures. In this process the shape measures and the responses of filters applied to the image crosstalk extensively. Finally, a top-down cleaning process is applied to avoid mixing the statistics of neighboring segments.

5) Blobworld [3], [9]: This is the basic segmentation tool used in the content-based image retrieval system *blobworld* [9]. Each image is segmented into regions by fitting a mixture of Gaussians to the data in a joint color-texture-position feature space by means of an EM algorithm. Each region ("blob") is then associated with color and texture descriptors, where the textural features taken into consideration are contrast, anisotropy and polarity. Finally, the optimal number of Gaussian components is automatically selected by means of the Minimum Description Length (MDL) criterion.

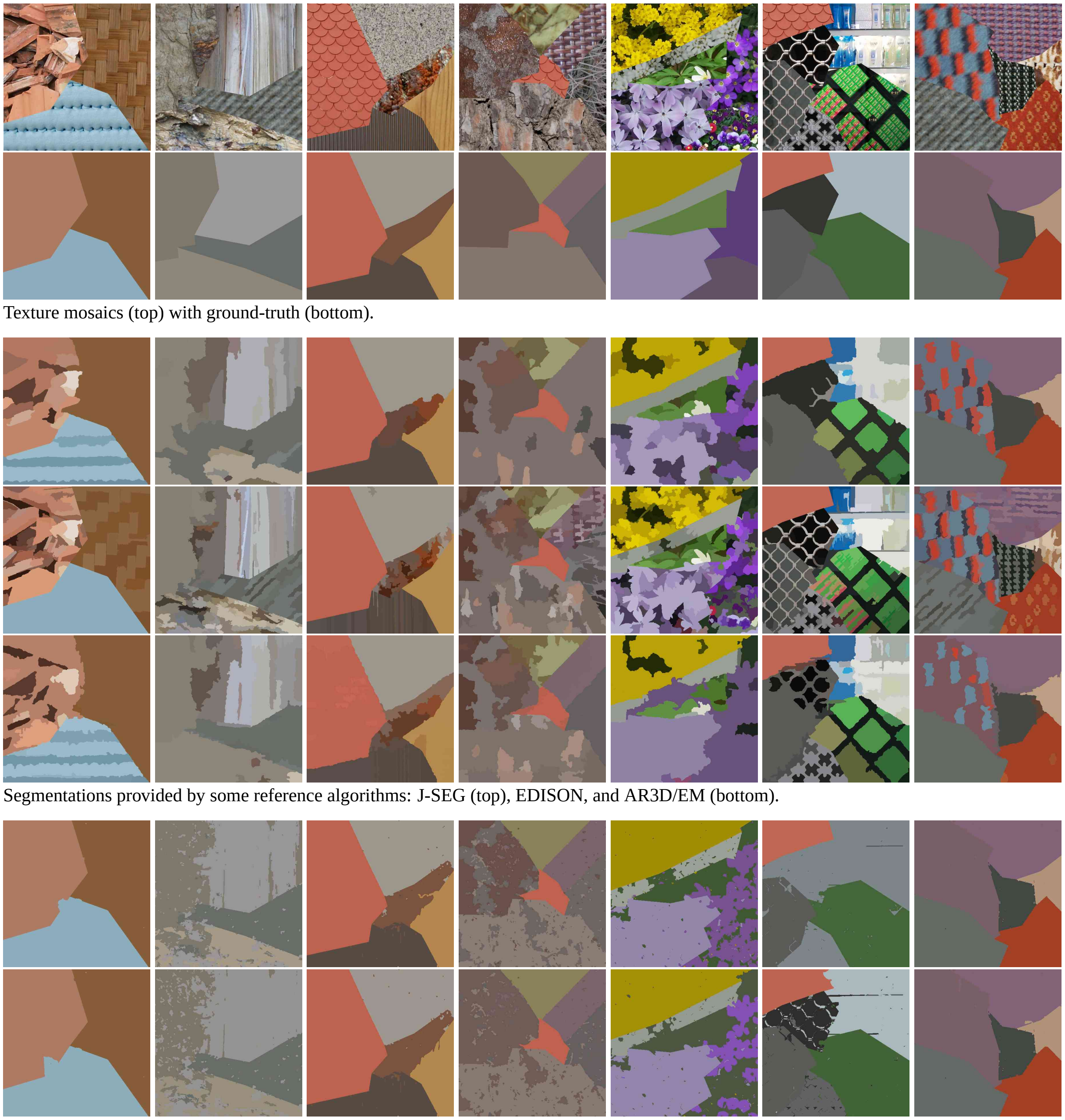
6) EDISON (Edge Detection and Image SegmentatiON system) [12]: This algorithm is based on the fusion of two basic vision operations, that is, image segmentation and edge detection; the former is based on global evidence, while the latter focused on local information. This integration is realized by embedding the discontinuity (edge) information into the region formation process, and then using it again to control a post-processing region fusion. In particular EDISON combines the *mean shift* based segmentation with a generalization of the traditional Canny edge detection procedure [8], which employs the confidence in the presence of an edge [31].

B. Segmentation results

Two versions of the proposed segmentation method were tested on the data set, referred to as TFR and TFR+, which are associated with the two definitions of texture score, see Eq.10 and Eq.14 respectively.

The benchmark data set is composed of twenty different 512×512 texture mosaics, seven of which are shown in Figure 5 together with the associated ground-truth and the corresponding segmentations performed by some reference techniques mentioned above and by the TFR method. The numerical results (averaged over the whole benchmark data set) are shown in Tab.I. As for the tuning parameters, we simply observed that all mosaic images never contains more than $M = 12$ different textures, and consequently we have $K_c = 2\,M = 24$ and $K_s = M = 12$, according to the heuristic rule discussed in section III. Indeed, we have run some tests with different values of M and obtained only slightly different results.

Observe that our segmenter is hierarchical, and hence it provides a stack of nested segmentation maps, among which one can pick the one that best matches the source data. This further selection step is by no means trivial, and simple rules, like the one proposed in [39] based on the region scale, perform poorly on such an heterogeneous data set. Here, we skip this problem, that goes beyond the scope of this work, and *manually* select the map that better fits visually the original mosaic. In other words, we keep separate the tasks of producing a good segmentation, and of selecting it amid the whole stack. Of course, this puts the proposed technique at an advantage w.r.t. the reference techniques. However, the reader should be aware that, for such complex images, producing even just *one* good map in the hierarchy is a remarkable result, and most reference techniques do not offer any easy option how to correct their wrong segmentation map, as can be seen from visual and numerical results.



Texture mosaics (top) with ground-truth (bottom).

Segmentations provided by some reference algorithms: J-SEG (top), EDISON, and AR3D/EM (bottom).

Segmentations provided by proposed TFR (top) and TFR+ (bottom) algorithms.

Fig. 5. Benchmark segmentation results. Data sets: 1, 2, 3, 4, 12, 14 and 19, from the left to the right.

	Benchmark – Colour							
	TFR+	**TFR**	**AR3D/EM**	**GMRF/EM**	**JSEG**	**SWA**	**Blobworld**	**EDISON**
↑ *CS*	**51.25**	46.13	37.42	31.93	27.47	27.06	21.01	12.68
↓ *OS*	5.84	**2.37**	59.53	53.27	38.62	50.21	7.33	86.91
↓ *US*	7.16	23.99	8.86	11.24	5.04	4.53*	9.30	**0.00**
↓ *ME*	31.64	26.70	12.54*	14.97	35.00	25.76	59.55	**2.48**
↓ *NE*	31.38	25.23	13.14*	16.91	35.50	27.50	61.68	**4.68**
↓ *O*	**23.60**	27.00	35.19	36.49	38.19	33.01	43.96	68.45
↓ *C*	22.42	26.47	11.85*	12.18	13.35	85.19	31.38	**0.86**
↑ *CA*	**67.45**	61.32	59.46	57.91	55.29	54.84	46.23	31.19
↑ *CO*	**76.40**	73.00	64.81	63.51	61.81	60.67	56.04	31.55
↑ *CC*	81.12	68.91	91.79*	89.26	87.70	88.17	73.62	**98.09**
↓ *I.*	**23.60**	27.00	35.19	36.49	38.19	39.33	43.96	68.45
↓ *II.*	4.09	8.56	3.39	3.14	3.66	2.11*	6.72	**0.24**
↑ *EA*	**75.80**	68.62	69.60	68.41	66.74	66.94	58.37	41.29
↑ *MS*	**65.19**	59.76	58.89	57.42	55.14	53.71	40.36	31.13
↓ *RM*	6.87	7.57	4.66	4.56*	4.62	6.11	7.52	**3.09**
↑ *CI*	**77.21**	69.73	73.15	71.80	70.27	70.32	61.31	50.29
↓ *GCE*	20.35	15.52	12.13*	16.03	18.45	17.27	31.16	**3.55**
↓ *LCE*	14.36	12.03	6.69*	7.31	11.64	11.49	23.19	**3.44**

TABLE I

PRAGUE TEXTURE SEGMENTATION BENCHMARK RESULTS. UP [DOWN] ARROWS INDICATE THAT LARGER [SMALLER] VALUES ARE BETTER BOLD NUMBERS INDICATE THE BEST TECHNIQUE, WHILE * MARKS A REPLACING BEST WHEN EDISON IS IGNORED.

The visual inspection of the segmentation maps shown in Fig. 5 is quite eloquent. For these images, in fact, TFR and TFR+ algorithms provide better results, and succeed in identifying very low frequency (macro) textures. This is well shown by data sets 14 and 19 (last two columns) for which TFR and TFR+ work properly, J-SEG has an almost acceptable over-segmentation, while other techniques excessively fragment the mosaics. In general, the reference algorithms seem to be able to model mainly micro textural features, which is likely the reason for this over-segmentation, confirmed numerically by the benchmark through the over-segmentation index OS (see Tab. I).

To be more precise, a common weakness of the reference techniques is that they either do not really classify the textures, but mainly detect contours among different neighboring textures, or they use single resolution texture representation. Therefore in most cases when the same texture occurs in different *unconnected* regions, each single region is differently labeled. As a typical example, see Fig.5, consider the 6th mosaic, where the green blocks on a black background are separated by all reference methods.[7] This last observation should make clear that a large gap exists between the proposed and the reference methods, which is not due to our manual selection.

Moving on the numerical results shown in Tab. I, it is interesting to notice the extremal behavior of EDISON which does not under-segment at all ($US = 0.0$), but almost always over-segments ($OS =$

[7] This holds also for the other methods not shown in figure for the sake of brevity.

86.91). Actually this is due to the fact that this algorithm was developed for very low order texture images, and can be viewed in this context almost as a color-based segmenter. For this reason the reader should not be surprised by its very good performance w.r.t. certain accuracy indicators, since they are all (directly or inversely) correlated with the degree of over-/under-segmentation.

Based on the above considerations, it would be legitimate to exclude EDISON from the analysis; nonetheless, we preferred to report its performance as well, since it represents in a sense an ideal case (the color-based segmenter). This allows us to recognize the indicators favored in case of over-segmentation, and for which EDISON scores serve as bounds for the other algorithms that do not over-segment.

On the opposite side, we have TFR which has the highest under-segmentation index $US = 23.99$ (see also the texture mosaic nr. 14, Fig. 5, 6th column, where only 4 out of 6 regions are recognized) while the modified version, TFR+, seems to reach the best tradeoff among all the algorithms, by keeping both indices very small ($OS = 5.84$, $US = 7.16$).

In Tab. I some of the indicators are to be minimized while the remaining are to be maximized (see arrows on the left-hand side). In any case the best method is emphasized with boldface numbers. Moreover, when EDISON is ignored the corresponding best points move on to other methods which are marked by *. As can be seen, all indices which are not optimized by EDISON are favorable to TFR+, except for OS which is minimized by TFR. The remaining parameters, when EDISON is not considered, mainly indicate AR3D/EM, except a few cases, as the best one. However, this is not very surprising if we look at the corresponding OS rate, which is rather high (59.53), and in any case, TFR+ provides quite good results even w.r.t. these indicators.

V. Experimental Result on Real Images

In order to provide a more solid assessment for the proposed technique and show its potential also w.r.t. different real life applications, this section discusses segmentation results obtained on natural and remote sensing images.

A. Application to the Berkeley Segmentation Dataset

Here we briefly discuss the application of the proposed algorithm to the domain of natural images, using a set of several color images taken from the Berkeley Segmentation Dataset [29].

For such images, we observed in general the presence of no more than $M = 6$ different textures, and consequently, according with the heuristic rule defined in Section III, we set $K_c = 12$ and $K_s = 6$.

Experimental results for some test images are reported in Fig.6. For each image we show the original on the left, the TFR segmentation map in the middle, and on the right the map obtained by SWA which is itself a hierarchical segmentation technique. As for the final segmentation result, the best matching maps are manually picked from the hierarchical stacks provided by the algorithms. For each segmentation map, the Local and Global Consistency Errors (LCE and GCE) indicators are evaluated w.r.t. each available ground truth, averaged and reported below the corresponding image. Moreover, by further processing the TFR maps with some simple morphological tools, we obtain smooth region contours which are superimposed on the original image to enable an easy interpretation.

Segmentation results are quite promising in many cases, with image textures and textured objects correctly identified in general: notably, the most accurate results have been obtained on images with at least one macro-textured object, such as the trivial foreground/background of the first two (top-left) images and the *wooden shoes* image. Here, large and regularly shaped fragments are gathered together to form quite well-defined states, whose interactions are consequently very well described by the H-MMCs. Besides, also in images characterized by the presence of areas of different nature (homogeneous, micro- and macro-textural), like the *zebras*, *woman*, and *buildings* images, results show all the potential of the method. Here, some problems occur in the presence of quasi-flat or gradient areas, that are more likely to be over-split, like the sky in the *buildings* image, and sometimes partially merged with unrelated textures, as occurs for the piece of background fused with the subject's hair in the *woman* image. A slightly lower accuracy is finally obtained with images that are mainly micro-textured and with loosely structured areas, above all because of the presence of over-fragmented elements or continuous regions whose characterization ends up to be less reliable. Nonetheless, even in these cases the main textures and objects are well identified in general.

The promising nature of the presented results is confirmed by numerical comparison with SWA. The TFR algorithm always outperforms the reference technique, except for a few cases where a better LCE is obtained by SWA, typically due to the presence of one or more refinement contours for which this indicator is more tolerant, as stated in [29].

B. Application to Multiresolution remotely-sensed data

We present here the results of a segmentation experiment carried out on a two-resolution remotely-sensed Ikonos image, of the city of San Diego (USA), containing both dense and residential urban areas, as well as a significant area covered with vegetation. In Fig.7(a) we show a false color representation of the image, that enhances the difference between urban areas and vegetation. In this case no ground-truth

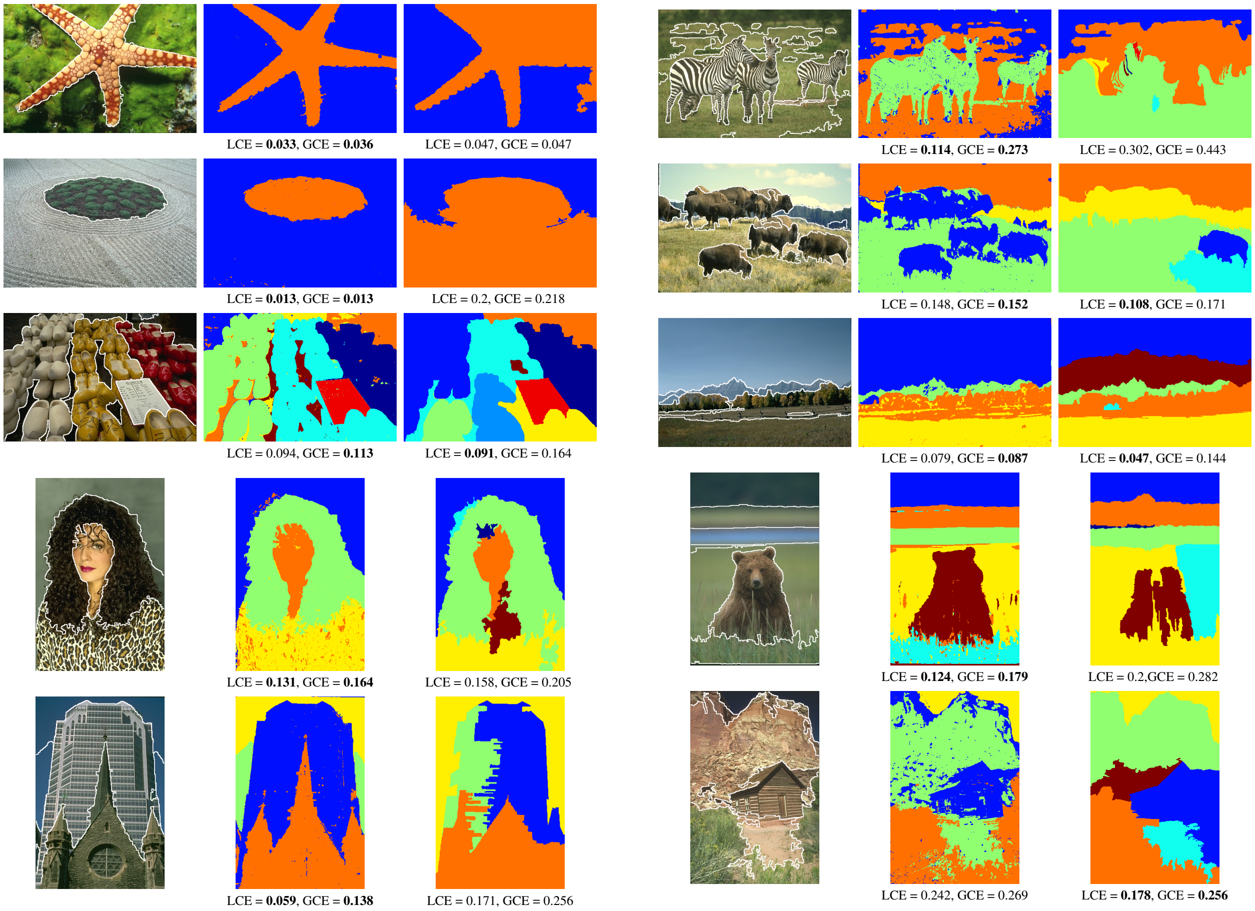


Fig. 6. Segmentation of natural images: some results obtained using the TFR algorithm on several color images taken from the Berkeley Segmentation Dataset. Below each image the mean *Local* and *Global Consistency Errors* (LCE and GCE) are reported (in bold, the best values for each experiment).

is available and hence we limit our analysis to the visual inspection of the segmentation results.

For these data we needed to adapt the CBC block to account for the multiple resolutions and the presence of a multispectral component. A detailed description of this algorithm can be found in [18].

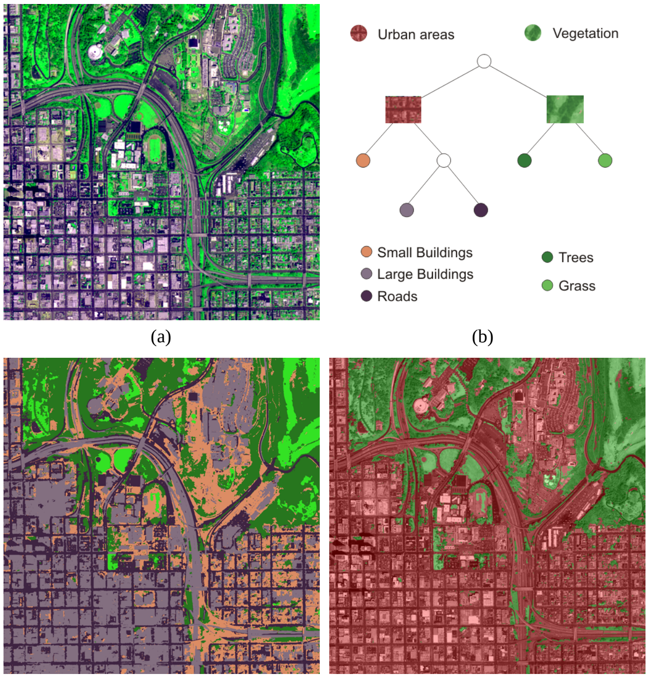
In Fig.7(b) we show the top part of the tree representing the merging process, pruned at an especially significant level, when only 5 nodes remain. By visual inspection of the corresponding segmentation map, shown in Fig.7(c), the nodes can be easily associated with classes of obvious significance for an observer, that is, the “small buildings”, “large buildings” and “roads” classes on one side of the tree and the “trees” and “grass” classes on the other side. With this compelling identification, image classification is rather accurate, considering that the segmentation process is totally unsupervised. Here, the aforementioned separation between “large buildings” and “small buildings” classes, with the latter generated by the fusion, at lower levels of the tree, of different clusters recognized as part of a more complex texture, is even more evident. Something similar happen for the “trees” and “grass” classes on the other branch. It is also worth underlining that the formulation of the texture score preserved the wide road network area from being fused with other smaller clusters in former stages of the process despite its strong interaction with other classes.

Going on with the merging process, we obtain eventually the two-class segmentation associated with the two top-level nodes, corresponding to the “urban” and “vegetation” macro-textures. The aforementioned binary segmentation is shown in Fig.7(d), where the urban area has been highlighted in red and the vegetation part in green. The detection of the two macro-textures is quite accurate, especially if one considers that some complex subtextures of the image, like the residential area in the lower right part, have been uniformly included in the “urban” class, as clear in Fig.7(d), although they include many large patches of vegetation. The key for this association seems to be the presence of a regular road network in this area, which acts as a collector of interacting classes: an information that a human interpreter would have certainly exploited to correctly classify this image, but that is taken into account automatically, here, by means of a fully unsupervised process.

VI. Conclusion and Future Research

In this paper we have presented a hierarchical model (H-MMC) for texture representation, particularly suited for unsupervised segmentation, and a related algorithm (TFR). In order to apply the model, the first step of the algorithm is a color-based segmentation, realized by TS-MRF, which provides a rough discrete approximation of the original data to be fitted with the texture model at the region level. The fitting is performed in two steps, the first (SBC) singles out the individual states of the model, the second relates them hierarchically according to the scale of the corresponding regions and their mutual spatial interaction. The bottom-up growth of the structure is controlled by a *texture score* parameter.

The performance of the proposed segmentation algorithm was assessed by experimenting with the texture mosaics of the Prague benchmark [32], that scores segmentation algorithms by means of several accuracy indicators. Moreover, the algorithm was also tested on the natural images of the Berkeley dataset, and on a multiresolution satellite image. Both numerical evidence and visual inspection show that the TFR outperforms all reference algorithms, mostly because of its ability to capture spatial correlations at multiple scales. On the contrary, all the methods using pixel-based texture modeling present serious limitations in representing macro-textural features, which is the case for most of the texture models found in the current literature. The experimental results also show that the performance of TFR improves when



(a) (b)

(c) (d)

Fig. 7. IKONOS image segmentation: 4m-resolution multispecral image, size 501×501, false color representation of the original image (a); 5-class pruning of the retrieved tree structure (b); 5-class segmentation (c); top-level classes: urban areas and vegetation (d).

the texture score includes the Kullback-Leibler divergence between the spatial distribution of the regions, since under-segmentation phenomena are reduced.

The main advantages of the proposed technique can be summarized as follows.

- **Robust**. Due to its region-based formulation and contrary to pixel-based models, the one proposed here is able to represent spatial interactions at multiple scales, leading to a nested hierarchical segmentation. Therefore, it does not require the choice of a specific observation scale, whose selection is left to the user, and the resulting algorithm is quite robust.
- **Fast**. Another consequence of modeling the image at a region level is the strong reduction of computational load, since the image processing involves regions, instead of pixels. Both TFR versions have about the same computational complexity (about 20 seconds of CPU time on a notebook with a 1.66 GHz processor for each 512×512 color image of the Prague benchmark), almost entirely due to the pixel-based processing of TS-MRF. Indeed the TS-MRF is not strictly needed and it could be replaced by much simpler color segmenters in all those applications where the definition of the

color classes can be easily provided. Think of video sequences, for example, where in most cases the color states may not change between subsequent frames, and a real-time video segmentation could be likely realized by means of TFR.

- **Blind**. The algorithm can be considered unsupervised because it does not require prior learning of involved textures, in spite of few non critical tuning parameters.

Although the TFR algorithm has provided encouraging results in several different applications, a few drawbacks need to be mentioned as well, mainly due to some of the simplifying assumptions both in the modeling and the optimization part. Discrimination of micro-textural features, for example, is often incorrect, since the small size of component regions (sometimes approaching a single pixel) makes their region-wise characterization unreliable. A possible solution is to identify small micro-textured regions at the CBC level, or even introduce a new layer with this specific aim.

As for spatial clustering, the presence of fragments whose characterization is loose can lead to the definition of unreliable states, that incorrectly include many “outliers” whose presence can significantly alter adjacency statistics w.r.t. neighboring states. The automatic detection and processing of such critical elements is certainly another point of our future research.

Finally, another peculiar problem of TFR is the processing of “continuous” connected regions, which typically occurs for textures containing background constant-colors. In this case, when two neighboring textures have a common color state which presents such continuous elements, due to their large scale they serve mostly as collectors during the region merging, attracting regions from the two different textures and eventually making their separation impossible. In order to overcome this last problem we are currently investigating the possibility of fragmenting continuous regions.

Acknowledgment

This work was carried out during the tenure of an ERCIM fellowship (Scarpa’s postdoc), and supported by EU MUSCLE project (e-team: shape modeling), FP6-507752, and partially by the project 102/08/0593. The authors are also grateful to the authors of the SWA algorithm [19] for providing their segmentation tool. Finally, a special thank goes to Prof. G. Poggi for his precious inspiration and valuable feedback.

References

[1] P. Andrey and P. Tarroux. Unsupervised segmentation of Markov random field modeled textured images using selectionist relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.20, No.3, pp.252–262, March 1998.

[2] A. Barbu and S. C. Zhu. Multigrid and Multi-level Swendsen-Wang Cuts for Hierarchic Graph Partitions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 731–738, 2004.

[3] S. Belongie, C. Carson, H. Greenspan, and J. Malik. Color- and texture-based image segmentation using EM and its application to content-based image retrieval. In *Proceedings of the Sixth International Conference on Computer Vision*, pages 675–682, Bombay, India, January 1998.

[4] J. Besag. Spatial Interaction and the Statistical Analysis of Lattice Systems. *Journal of the Royal Statistical Society, series B*, B-36(2):192–236, February 1974.

[5] C. A. Bouman and M. Shapiro. A multiscale random field model for Bayesian image segmentation. *IEEE Transactions on Image Processing*, Vol.3, No.2, pp.162–177, March 1994.

[6] T. Brox, J. Weickert. Level Set Segmentation With Multiple Regions *IEEE Transactions on Image Processing*, Vol. 15, No. 10, pp. 3213–3218, October 2006.

[7] T. Brox, J. Weickert. A TV Flow Based Local Scale Measure for Texture Discrimination *Proc. 8th Eur. Conf. Computer Vision*, Vol. 2, pp. 578-590, May 2004.

[8] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 8:679–698, 1986.

[9] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik. Blobworld: A system for region-based image indexing and retrieval. In *Third International Conference on Visual Information Systems*, pages 509–516, Amsterdam, The Netherlands, 1999.

[10] D. Charalampidis and T. Kasparis. Wavelet-based rotational invariant roughness features for texture classification and segmentation. *IEEE Transactions on Image Processing*, Vol.11, No.8, pp.825–837, August 2002.

[11] B. B. Chaudhuri and N. Sarkar. Texture segmentation using fractal dimension. *IEEE Transactions on Pattern Analysis Machine Intelligence*, Vol.17, No.1, pp.72–77, January 1995

[12] C. M. Christoudias, B. Georgescu, and P. Meer. Synergism in low level vision. In R. Kasturi, D. Laurendeau, and C. Suen, editors, *Proceedings of the 16th International Conference on Pattern Recognition*, volume 4, pages 150–155, Los Alamitos, August 2002.

[13] D. A. Clausi and H. Deng. Design-based texture features fusion using Gabor filters and co-occurrence probabilities In *IEEE Transactions on Image Processing*, Vol. 14, No. 7, pp. 925–936, July 2005.

[14] T. Cour, F. Bnzit, J. Shi. Spectral Segmentation with Multiscale Graph Decomposition *Proc. of IEEE Conference on Computer Vision and Pattern Recognition CVPR 2005*, Vol. 2, pp. 1124–1131, June 2005.

[15] C. D'Elia, G. Poggi, and G. Scarpa. A tree-structured Markov random field model for Bayesian image segmentation. *IEEE Transactions on Image Processing*, 12(10):1259–1273, October 2003.

[16] Y. Deng and B. S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 23(8):800–810, 2001.

[17] G. Fan and X.-G. Xia. Wavelet-based texture analysis and synthesis using hidden Markov models *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Vol. 50, Nr. 1, pp. 106–120, January 2003.

[18] R. Gaetano, G. Scarpa and G. Poggi. A hierarchical segmentation algorithm for multiresolution satellite images. In *Proceedings of IEEE International Geoscience and Remote Sensing Symposium*, 2007.

[19] M. Galun, E. Sharon, R. Basri and A. Brandt. Texture segmentation by multiscale aggregation of filter responses and shape elements. In *Proceedings of IEEE International Conference on Computer Vision*, Vol. 1, pp. 716–723, 2003.

[20] M. Haindl and S. Mikeš. Model-based texture segmentation. In A. Campilho and M. Kamel, editors, *Image Analysis and Recognition*, Lecture Notes in Computer Science 3212, pages 306–313, Porto, Portugal, 2004.

[21] M. Haindl and S. Mikeš. Colour texture segmentation using modelling approach. In *Proc. 3th ICARP*, Lecture Notes in Computer Science 3687, pages 484–491, Bath, UK, 2005.

[22] M. Haindl and S. Mikeš. Unsupervised Texture Segmentation Using Multispectral Modelling Approach. In *Proceedings of the 18th International Conference on Pattern Recognition, ICPR 2006*, Volume 2, pages 203-206, Hong Kong, China, August 2006.

[23] R. M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, Vol.67, No.5, pp.786–804, May 1979.

[24] T. Hofmann, J. Puzicha and J. M. Buhmann. An optimization approach to unsupervised hierarchical texture segmentation. In *Proceedings of IEEE International Conference on Image Processing*, Vol. 3, pp. 213–216, 1997.

[25] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. W. Bowyer, D. W. Eggert, A. W. Fitzgibbon, and R. B. Fisher. An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 18(7):673–689, 1996.

[26] H. C. Hsin. Texture segmentation using modulated wavelet transform. *IEEE Transactions on Image Processing*, Vol.9, No.7, pp.1299–1302, July 2000.

[27] S. Krishnamachari and R. Chellappa. Multiresolution Gauss-Markov random field models for texture segmentation. *IEEE Transactions on Image Processing*, Vol.6, No.2, pp.251–267, February 1997.

[28] M. R. Luettgen, W. C. Karl, A. S. Willsky, R. R. Tenney. Multiscale representations of Markov random fields. In *IEEE Transactions on Signal Processing*, Volume 41, Number 12, pages 3377–3396, December 1993.

[29] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th ICCV*, volume 2, pages 416–423, Vancouver, Canada, July 2001.

[30] Y. Ma, H. Derksen, W. Hong and J. Wright. Segmentation of Multivariate Mixed Data via Lossy Data Coding and Compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1546–1562, 2007.

[31] P. Meer and B. Georgescu. Edge detection with embedded confidence. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 23(12):1351–1365, 2001.

[32] S. Mikeš and M. Haindl. Prague texture segmentation data generator and benchmark. *ERCIM News*, number 64, pages 67-68, 2006. http://mosaic.utia.cas.cz

[33] N. Paragios, R. Deriche. Geodesic Active Regions and Level Set Methods for Supervised Texture Segmentation *International Journal of Computer Vision*, Vol. 46, No. 3, pp. 223–247, 2002.

[34] W. D. Penny. Kullback-Leibler divergences of normal, gamma, Dirichlet and Wishart densities. Technical report, Wellcome Department of Imaging Neuroscience, University College Longon, 2001.

[35] O. Pichler, A. Teuner and B. J. Hosticka. An unsupervised texture segmentation algorithm with feature space reduction and knowledge feedback. In *IEEE Transactions on Image Processing*, Vol.7, No.1, pp.53–61, January 1998.

[36] G. Poggi, G. Scarpa, and J. Zerubia. Supervised segmentation of remote-sensing images based on a tree-structured MRF model. *IEEE Transactions on Geoscience and Remote Sensing*, 43(8):1901–1911, August 2005.

[37] J. Portillo - Garcia, I. Trueba - Santander, G. de Miguel - Vela and C. Alberola - Lopez. Efficient multispectral texture segmentation using multivariate statistics. *Vision, Image and Signal Processing, IEE Proceedings*, Vol.154, No.5, pp.357–364, October 1998.

[38] M. Rousson, T. Brox, R. Deriche. Active Unsupervised Texture Segmentation on a Diffusion Based Feature Space *Proc. of IEEE Conference on Computer Vision and Pattern Recognition CVPR 2003*, Vol. 2, pp. 699–704, June 2003.

[39] G. Scarpa and M. Haindl. Unsupervised Texture Segmentation by Spectral-Spatial-Independent Clustering. In *Proc. 18th ICPR*, Vol.2, pp.151–154, Hong Kong (China), August 2006.

[40] G. Scarpa, M. Haindl and J. Zerubia. A hierarchical finite-state model for texture segmentation. In *Proceedings of ICASSP 2007*, Vol.1, pp.I-1209 – I-1212, Honolulu, HI (USA), April 2007.

[41] G. Scarpa, M. Haindl and J. Zerubia. A hierarchical texture model for unsupervised segmentation of remotely sensed images. B.K.Ersbøll and K.S.Pedersen (Eds.): SCIA 2007, LNCS 4522, 303–312, Aalborg (Denmark), June 2007.

[42] J. Shi, J. Malik. Normalized Cuts and Image Segmentation *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 8, pp. 888–905, August 2000.

[43] L.-K. Soh and C. Tsatsoulis. Texture analysis of SAR sea ice imagery using gray level co-occurrence matrices. *IEEE Transactions on Geoscience and Remote Sensing*, Vol.37, No.2, pp.780–795, March 1999.

[44] M. Tuceryan and A. K.Jain. Texture analysis. *The Handbook of Pattern Recognition and Computer Vision*, 2nd Edition. C.H.Chen, L.F.Pau, P.S.P.Wang, Ed. River Edge, NJ: World Scientific, pp. 207-248, 1998.

[45] M. Unser. Texture classification and segmentation using wavelet frames. *IEEE Transactions on Image Processing*, Vol.4, No.11, pp.1549–1560, November 1995.

[46] J. Wu, A. C. S. Chung. A Segmentation Model Using Compound Markov Random Fields Based on a Boundary Model. *IEEE Transactions on Image Processing*, Vol.16, No.1, pp.241–252, January 2007.

[47] Y. Xia, D. Feng and R. Zhao. Morphology-based multifractal estimation for texture segmentation. *IEEE Transactions on Image Processing*, Vol.15, No.3, pp.614–623, March 2006.

[48] S. C. Zhu, C. E. Guo, Y. Z. Wang and Z. J. Xu. What are Textons? *International Journal of Computer Vision*, Vol. 62, No. 1/2, pp. 121–143, 2005.