TESI DI DOTTORATO

UNIVERSITÀ DEGLI STUDI DI NAPOLI "FEDERICO II"

DIPARTIMENTO DI INGEGNERIA ELETTRONICA
E DELLE TELECOMUNICAZIONI

DOTTORATO DI RICERCA IN
INGEGNERIA ELETTRONICA E DELLE TELECOMUNICAZIONI

HIERARCHICAL MODELS FOR IMAGE SEGMENTATION: FROM COLOR TO TEXTURE

RAFFAELE GAETANO

Il Coordinatore del Corso di Dottorato	Il Tutore
Ch.mo Prof. Giovanni POGGI	Ch.mo Prof. Giovanni POGGI

“I can’t think of anything to say except...
I think it’s marvelous!
HaHaHa!”

Acknowledgments

First of all, I want to express my vigorous, sincere and deep feeling of gratitude to my advisor Prof. Giovanni Poggi and my collaborator Giuseppe Scarpa, for their constant and invaluable support, enlightening advices for research as for life, for treating me as a friend before than a co-worker. I want them to know that, without their inspiring presence, my experience throughout the PhD would not have been started at all. Thanks also to all the people that shared beautiful moments, as well as troubled times, with me during these three and more years, especially to my friend Jurek and Basciando, with whom I feel we've grown up together, and all of my friends and family that pushed me and supported me in many ways from "outside". I would also like to thank Prof. Josiane Zerubia and all the people from ARIANA group, for their active collaboration and friendship during the months I spent at their laboratory in Sophia Antipolis (F). Thanks are also due the French Space Agency (CNES) for providing the SPOT data through the ISIS programme and the GSTB, and the COSTEL laboratory for providing the ground-truth. Finally, I thank the Italian MIUR for its support through the PRIN 2006 program.

Raffaele Gaetano

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction to Image Segmentation

In the image processing domain, *segmentation* is the operation that allows one to partition an image into a set of different regions, each one homogeneous with respect to some properties like intensity, color, texture, shape, *etc.* It is a low level task that proved to be useful in a wide range of high-level processing and applications in such diverse fields as remote-sensing [1, 2], medical imaging [3, 4], video coding [5, 6], and industrial automation [7, 8], just to name a few. More recently, image segmentation has been often used as a basic step in many techniques related to the analysis of image contents, as it happens in the framework of *Content Based Image Retrieval* (CBIR) [9] for applications like multimedia digital libraries or digital image databases. Given the wide-ranging scope of image segmentation, it is easily understood that such a problem can be addressed with a wide variety of approaches, typically leading to application-specific solutions that can also make sense at different levels of abstraction.

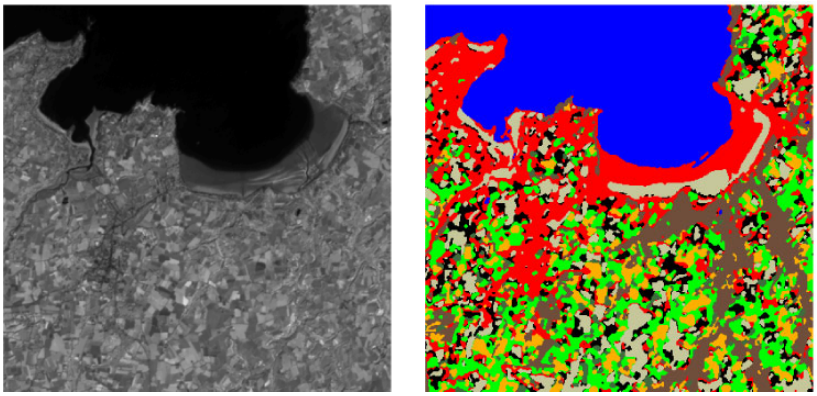
From its very beginning, dating back to the early 70's, research on image quantization has been characterized by a very large spectrum of approaches and solutions, which can be loosely grouped in three main categories [10]: *clustering* based methods, where pixels of the image are grouped together according to some aggregative or divisive criterion involving only their values in the intensity/color space (*e.g. histogram* based methods); *region growing* methods, where pixels are gradually aggregated starting form properly selected seeds by means of a suitable "distance" metric; and *edge detection* methods, where image regions are identified starting from their contour, that is, by iden-

tifying points where significant changes in the image properties, *e.g.* strong variations in pixel intensities, take place.

In all these early methods the segmentation process was *deterministic*, relying only on the observed data, without any assumption on the nature of the source and any use of prior knowledge. This turned soon out to be a strong limitation in many real-life applications, and from this point on, barring trivial situations in which such a simplistic approach could suffice, the *model based* approach became the dominant paradigm for image segmentation. With this approach, the whole prior knowledge about the data is used to build a mathematical model of the image, which in turn defines the rules for the aggregation of image elements.

Currently, two main model-based frameworks dominate the research field. On one side, we find the *variational* methods [11], which rely on the definition of an energy functional depending on the data and their partition: minimization of this energy over the possible partitions, typically performed using variational mathematical methods such as partial differential equations (PDEs), provides the desired segmentation map. A notable example is represented by the active contour [12] techniques (a special case of the well-known level set methods [13]), where the main idea is to evolve contour curves towards their lowest energy configuration, fitting to the actual boundaries among different image objects. On the other side we have the *bayesian* framework and the *Markov Random Field* (MRF) models [14, 15], which gained a large popularity because of their effectiveness and flexibility in defining "local" dependencies among adjacent pixels, thus encompassing prior knowledge in the segmentation process with a reasonable complexity. MRFs represent the basis for a consistent part of this work of thesis, and will be discussed in details in the following sections. Needless to say, other approaches exist which do not fit in the former frameworks, like the *graph based* methods [16] relying on a graph-theoretic formulation of the concept of "grouping". Their discussion, however, goes beyond the scope of this thesis.

In this work, we focus on *probabilistic* image models where images are supposed to be generated according to some probability law, and the final segmentation map is obtained by means of a statistical inference between the model and the image itself. This choice arises a number of complex issues, and in particular: the formulation of a suitable image model; the definition of accurate optimization procedures; the development of limited-complexity algorithms. The latter point, in particular, should not be understated, since a trade-off exist between the accuracy of image description and the efficiency



(a) (b)

Figure 1.1: Example of land cover classification from a low-resolution (20 m) remote sensing image: regions of interest highlighted in the map (b) are homogeneous in terms of spectral properties, as can be seen on the source (a).

of the segmentation process, and the actual success of a segmentation technique might strongly depend on its overall complexity. All these topics will be dealt with in this thesis with reference to a variety of different applications and domains. In particular, we will follow an ideal path that goes from the problem of *color based segmentation*, based on some form of homogeneity in the color/spectral properties of the image, to the more complex task of *texture based segmentation*, where the aim is to recognize complex structures in the image which are typically non homogeneous in terms of spectral properties.

1.2 From Color to Texture

As said above, we will first consider the segmentation based only on the spectral features of the image, meaning that the only processed data will be the spectral responses of image pixels, like the *red*, *green* and *blue* bands for color images. More in general, a source image can be compounded by an arbitrary number of spectral bands, as happens in remote sensing where up to a few hundreds of bands can be made available by capture sensors (multispectral to hyperspectral images).

As a matter of facts, classification of remotely sensed images is one of the

(a) (b)

Figure 1.2: Samples of *vegetation* area (a) and *urban* area from a high-resolution (1 m) satellite image.

most common applications strictly related to color based segmentation, which is particularly useful in the field of Earth observation (resource monitoring, disaster prevention and risk management, *etc.*). In Fig. 1.1, an example using a low-resolution (20 m) SPOT multispectral image of Lannion Bay, France, is reported: one of the three spectral bands available is depicted in (a), while a possible classification map is shown in (b), where each color corresponds to a region of interest with a precise physical meaning (a *ground-truth* of 8 different land covers is also available with the data). For this kind of data, it is quite evident that each land cover class is characterized by high spectral homogeneity, to the point that, for this particular domain, a segmentation technique relying only on spectral properties can functionally represent a classification method by itself.

Still in the very same domain of remote sensing, let us now consider on a different kind of data, such as the high-resolution (1 m) images shown in Fig. 1.2, provided by a new generation sensor of the Ikonos satellite. Here, unless we focus on elementary objects (isolated buildings, roads, trees, *etc.*), land covers of actual interest for classification, such as *vegetation* (a) or *urban areas* (b), are by no means homogeneous in terms of spectral properties, and more complex features must be used to correctly single them out in the context of a complex image. As a matter of fact, the images of Fig. 1.2 are both charac-

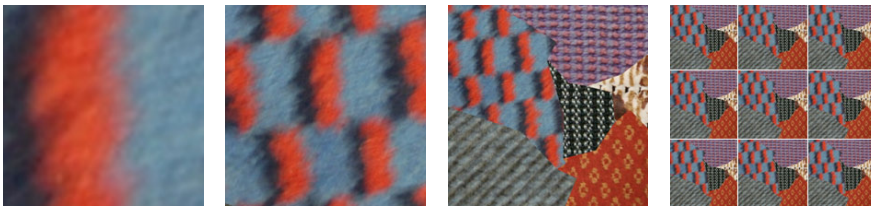


Figure 1.3: From color to texture: *fine-to-coarse* hierarchical inspection of a textured image.

terized by a marked textural nature: in the case of vegetation, the image can be regarded as a quasi-random composition of dark green and light green patches, while a more structured pattern of buildings, green spots and roads exist in the urban area. Therefore, a segmentation technique that can recognize textured regions as a whole, in spite of their non-homogeneous spectral properties, becomes a very important tool in view of a subsequent classification of this kind of images.

Anyway, it should be clear that textural and spectral information are deeply intertwined, and telling them apart depends on the point of view or the *scale*. To better understand this point, let us take a look at the image in Fig. 1.3: a "natural" segmentation of the first image on the left will reasonably involve only color information, since the *blue*, *black* and *red* regions represent the only significant entities. Zooming out from this area (second image) a pattern of color patches emerges, that a human observer quickly attributes to a single textural entity because of the intensive spatial interaction among the elementary color patches. Only moving to coarser scales the spectral information starts to become less relevant, since the same colors are present in different textures which are identified only by means of their spatial and contextual characteristics. The last image, finally, can be regarded again as a unique macro-texture.

Two main considerations arise from this example. First, textures must be looked for and identified at multiple scales of observations, since the same area, depending on the scale, can be seen as a single homogeneous region or a component of a larger and more complex texture. This motivates our use of a hierarchical approach, to be described in the following. Second, the accurate detection of spectrally homogeneous areas remains an important step towards an effective texture segmentation, since the finest-scale interactions among uniform areas are the basis for texture detection.

For these reasons, in this work we devote attention both to color-based and texture-based segmentation, treating these two topics independently so as to provide a deeper insight in the context of the respective frameworks and to avoid the use of too specific solutions.

1.2.1 Hierarchical MRF based Image Segmentation

Markov Random Field (MRF) models in computer vision have been first formalized by Besag [17], and have become popular with the seminal paper of S.Geman and D.Geman on image restoration [15]. The field has grown up rapidly in recent years addressing a variety of low-level[1] image tasks, such as *image compression*, *restoration*, *segmentation*, *etc*.

The use of MRF for image modeling is related to the fundamental assumption that each single pixel depends statistically on the rest of the image only through a selected set of neighbors. For image segmentation, considering the aforementioned *bayesian* framework, all the *a priori* knowledge available on the image can be transferred onto the model, in the general case, by defining the joint probability $p(x, y) = p(y|x)p(x)$ where y represents the data and x is the unknown segmentation map. Thanks to the Bayes' rule, this can be done by defining separately a *conditional data likelihood* model $p(y|x)$ and a *prior* model $p(x)$, where the latter can be defined, using MRFs, as the sum of relatively simple local contributions in the form of suitable *potential* functions. Segmentation is finally performed by selecting the map according to some useful statistical criterion: for example, a very popular choice is the *Maximum A Posteriori* (MAP) criterion, aimed at maximizing the *posterior* probability $p(x|y)$ over x.

The definition of a suitable MRF model through its potentials is all but a simple task, and typically results from a trade-off between description accuracy and mathematical/numerical tractability. In particular, if the potentials, as well as the likelihood model, are defined in a parametric form, the resulting optimization procedure will have an iterative nature, alternating between parameter estimation and the minimization of a cost function (*e.g.* the MAP estimation). In this case, the more sophisticated is the model (with many free parameters), the heavier is the computational load of the resulting segmentation algorithm, including parameter estimation. As a matter of fact, computational complexity is the major weakness of MRF based techniques, so much so that a substantial part of the research in this field has been devoted to the

[1] *Low-level* is a traditional terminology for preliminary tasks to image understanding.

study of modeling strategies which provide reliable segmentations with limited computational effort.

In this thesis we focus on a particular class of hierarchical MRF models, the *tree-structured Markov random fields* (TS-MRF) [18], relying on the observation that images often present a hierarchical structure, namely, an image can be viewed as a collection of regions at multiple scales of observation, hierarchically related to each other by means of a tree structure. The image is therefore regarded as a tree of regions, where each node represents a portion of the image (with the root corresponding to the whole image) and the children nodes are associated with the different areas of a partition of the given region. TS-MRF models aim at describing such a structured image by means of a corresponding tree of MRFs, each one adapted to a particular region of the image, and each one corresponding to a node in a tree of models, with all model parameters defined locally to that node.

Such models are defined recursively and, as such, allow for a recursive optimization which reduces the K-ary segmentation task to a sequence of much simpler local segmentations. Each temporary region will be then associated with a node of the tree, while its segmentation corresponds to a node split. The global image segmentation map is therefore obtained as a result of the component segmentations and corresponds to the regions attached to terminal nodes.

Innovation

Segmentation based on the TS-MRF model has proven very successful in the supervised case [1], when the number of classes of interest and their synthetic parameters are known *a priori*. In the unsupervised case [18] results are also good, especially if compared with those of unstructured techniques, but some critical issues remain to be addressed. In fact, lacking any prior information, one is forced to estimate, by recursive optimization at each node, the very same tree structure underlying the data. If the optimization is inaccurate at some nodes, the whole tree structure might deviate from the most suitable one, with various undesirable effects, like the fusion of different classes or the oversplitting of others.

In this work we propose an improved version of the TS-MRF unsupervised segmentation algorithm that addresses the major problems briefly outlined above. The main improvements come from the use of a Mean-Shift based clustering. As a matter of fact, the Mean-Shift procedure [19] was already used in a preliminary stage of research to dynamically detect the number of modes

at each step of the tree growing procedure, and hence the number of children for each node of the tree, implicitly allowing the use of generic tree structure by removing the binary constraint. Here, however, its use is carried further, and besides finding the dominant modes for each class, it replaces the *Generalized Lloyd Algorithm* (GLA) [20] as the initial clustering technique, providing a much more reliable starting point for the subsequent MRF-based segmentation, and a much easier and stable detection of the correct tree-structure for the data. This is obtained through some significant modification of the Mean-Shift clustering itself, which now makes use of a variable-bandwidth strategy based on the k-*Nearest Neighbors* (k-NN) technique, and is implemented with a speed-up strategy that cuts significantly the computational complexity, otherwise intolerable for such applications.

1.2.2 Hierarchical Models for Texture Segmentation

When dealing with images with a significant textural content, spatial interactions among image elements usually cover long ranges, asking for complex high order modeling. Such a task is especially demanding 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 spot automatically. The main problem lies in texture definition itself which is still quite debated [21, 22] without any general agreement. As a matter of fact, the definition of what constitutes a texture depends too often on the intended application, or on different perceptual motivations, leading frequently to a number of constraints that fit very well some specific class of images but are meaningless for other more general applications. Therefore, in this work we escape the hardship of giving yet another definition, focusing instead on two quite objective and agreed-upon categorizations for "elementary" textures, that is, *structured* vs. *non-structured*, and *micro-* vs. *macro*-textures. The former classification arises from the nature (deterministic or stochastic, respectively) of a possible model for texture generation. The latter refers to the spatial correlation scale of the texture, which spans a continuous range whose extremes are micro- and macro-textures. In any case, natural textures are rarely so homogeneous to be ascribed precisely to one category or another, and it often happens that a single texture can be regarded in turn as the composition of several subtextures, in which cases we will generally speak of "complex" textures.

In current literature, texture segmentation is mostly regarded as the composition of two different (although tightly related) problems, the choice of a

suitable representation of textures, in order to establish what is to be identified, and the definition of a framework and strategy for the actual segmentation.

Texture representation, as noted before, can be addressed using many different approaches, the most well-known being the use of statistical, geometrical, or transform-domain features and the use of suitable image models. Co-occurrence matrices [23, 24], introduced in the pioneering work of Haralick [24], are a classical example of statistical features. Such matrices account for co-occurring colors in pairs of image sites parameterized by their distance and orientation, and they provide a good discrimination power, with acceptable complexity, if some prior knowledge is available about the directionality, spatial interaction scale and color content of the textures involved. A more complex feature extraction approach can take into account geometrical features, like the fractal dimension used in [25, 26]. On the up side, fractal dimension is relatively insensitive to image scaling and shows a strong correlation with human judgment of surface roughness. Unfortunately, they provide limited texture discriminatory information, and hence are not very effective for texture analysis. At present, most of the literature about texture representation focuses on transform-domain features [27, 28], with Gabor [23, 29, 30] and wavelet [31, 32] filters being by far the most popular. Indeed, Gabor filters exhibit excellent space/frequency resolution [30] as well as good orientation and frequency selectivity. Their main drawback is the excessive computational load due to the large number of filter parameters to select, from spatial scale, to carrier frequency and orientation. Wavelet-based methods present a much smaller complexity which, together with their many appealing properties, like the inherent multi-resolution and the high flexibility, have merited them a great deal of attention [27, 31, 32]. 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, and equally popular, approach to texture representation is based on the use of a suitable texture model [33, 34, 35, 36, 37]. Markov Random Fields (MRF) models, given their success on non-textured images [1, 38] are natural candidates, 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 [33, 36]. For this reason, multi-resolution Hierarchical MRFs [39, 36] or two-dimensional causal autoregressive models [35, 37] have been proposed, which allow to model long-range dependencies at the price of a generally higher computational or modeling complexity.

Turning to the actual segmentation methods, it is reasonable to refer to

the classical image segmentation literature, and classify the many proposals as *edge-based* or *region-based* techniques. For the first category, some interesting variational techniques have been proposed recently [40, 41, 42, 43], where boundaries among textures are retrieved using curve evolution methods driven by a suitable 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 [39, 36], usually heavy in terms of computational complexity, some region growing techniques have been recently proposed [44], based on the split-and-merge paradigm, where the 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, graph-cuts methods have been applied over a suitably chosen textural feature space [16, 45], where no specific modification is proposed in terms of optimization procedure to deal with textures, especially in the structured and macro-textured case.

Innovation

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 [46]. The use of multiple scales [47, 28] 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 [47, 29, 48] (sequence of nested segmentations) where the number of texture segments is not explicitly singled out.

As a consequence, the proposed *Texture Fragmentation and Reconstruction* (TFR) algorithm follows the *split-and-merge* paradigm: the first (split) step provides the "elementary" regions of the image, that is, the basic components of all the different textures present in the scene, while the subsequent (merge) step reconstructs the textural content in a hierarchical, multi-scale fashion. As already recalled, segmentation and modeling are deeply dependent on one another, and in fact the proposed TFR algorithm is based in turn on a hierarchical region-level description of the image, where inter-region interac-

tions are modeled by means of a set of Markov chains, referring to different spatial orientations. Based on such spatial interactions, elementary regions are also recursively coupled, giving rise to a hierarchy of nested image models, which accounts for the desired multi-scale property and leads naturally to a hierarchical texture segmentation algorithm.

Chapter 2

Tree Structured Markov Random Fields for Segmentation

In this chapter we provide the necessary background about MRFs in general, describing the most important theoretical results, as well as one of the most successful practical models, namely the Potts model in its generalized formulation. Then, the MRF-based approach to segmentation is considered, and the most relevant related problems are analyzed in detail. In the second part of the chapter, theoretical and practical achievements concerning the Tree Structured Markov Random Field *(TS-MRF) class of models are discussed, and the related properties are analyzed in depth, such as the flexibility of the definition, the recursive nature of the model and the corresponding recursive optimization procedures, the robustness of the TS-MRF methods. Then the focus goes on unsupervised algorithms derived from TS-MRFs, with the analysis of their critical points.*

2.1 Markov Random Field based Image Modeling

In the following sections, we concentrate the attention on the use of MRFs for image segmentation, and provide some details on this application, from the general statistical framework to the specific modeling strategy.

2.1.1 Using MRFs for Segmentation

The Bayesian Approach

In the statistical framework, the segmentation problem is approached by choosing an *ad hoc* probabilistic model, to fit the data and the unknown segmentation map. In the basic formulation[1], image data are represented by a continuous vectorial 2-D field $y = \{y_s : s \in \mathcal{S}\}$, with $y \in \mathcal{R}^B$, where s is a site of the *lattice* $\mathcal{S}$ and B is the number of image channels. The data are then assumed to be the realization of a random field Y, namely the *observation field*, whose probability distribution is $p(y)$[2] Likewise, the unknown segmentation map $x = \{x_s : s \in \mathcal{S}\} \in \Omega = \Lambda^{\mathcal{S}}$, where $\Lambda = \{0, 1, 2, \ldots, K-1\}$ is the label set and K is the number of *classes*, is the realization of a discrete 2-d random field X, the *label field*, with distribution $p(x)$.

Once the probabilistic model is defined, solving the segmentation problem relies on finding a proper estimate of the map x, say $\hat{x}$. In the *bayesian decision theory* (see [49] for further details), an estimator is typically derived from the definition of a *cost function* $R(x, x')$, that quantifies the errors made by estimating the "real" solution x with x', and the minimization of the corresponding *Bayes' risk*, defined as the mean of the cost function over x, leading to

$$\hat{x} = \arg\min_{x' \in \Omega} \int_{x \in \Omega} R(x, x') p(x|y) dx$$

where, thanks to the Bayes' formula we can explicit the *a posteriori* distribution $p(x|y)$ as:

$$p(x|y) = \frac{p(x, y)}{p(y)} = \frac{p(x)p(y|x)}{p(y)}.$$

A very popular estimator, in particular in the image processing domain, is the so called *Maximum a Posteriori* (MAP) estimator, that makes use of a very simple cost function having value 0 if no errors occurr and 1 otherwise, irrespective of the total number of errors:

$$R(x, x') = 1 - \Delta_{x'}(x),$$

[1] In this context data are considered as raw, without any processing or transformation, and the segmentation is similarly represented as 2-D map although other points of view could be assumed (*e.g.*, contour set).

[2] Where unambiguous, we will indicate the probability law associated with X simply as $p(x)$, to be meant as either a density or a distribution function depending on the case.

where Δ is the *Dirac* function in x'. The corresponding estimator hence takes the following form [50]:

$$\hat{x}_{MAP} = \arg\min_x p(x|y) = \arg\min_x p(x)p(y|x), \tag{2.1}$$

in which the term $p(y)$ is neglected, since the observation occurrs with probability equal to 1. Such estimator gives, for a given observation y, the modes of the posterior distribution, that is, the most likely segmentation maps x given y.

MRF-based Image Modeling

In the described context, image modeling completely relies on the specification of the two terms in RHS of Eq. 2.1: the first one, $p(x)$, is called *prior model*, and is useful to encompass any prior knowledge into the segmentation process, while the second one, namely the *likelihood term* $p(y|x)$, is responsible to take into account data similarity with respect to the segmentation map.

For the latter, a classical choice is to consider it to be spatially independent, meaning that each site is independent from each other and with a local conditional density whose parameters are class-dependent:

$$p(y|x) = \prod_{s \in \mathcal{S}} p(y_s|x_s). \tag{2.2}$$

Such densities are often modeled using Gaussians, that is:

$$p(y_s|x_s = k) = \frac{1}{(2\pi)^{B/2}|\Sigma_k|^{1/2}} \exp[-\frac{1}{2}(y_s - \mu_k)^T \Sigma_k^{-1}(y_s - \mu_k)], \tag{2.3}$$

where μ_k and Σ_k are the mean vector and the covariance matrix of class k respectively.

When no assumption is made on the prior model, that is, when $p(x)$ is modeled with a uniform distribution, the estimator of Eq. 2.1 becomes the well known *Maximum Likelihood* estimator, for which, under the mentioned hypotesis, optimization can be pursued separately for each x_s, considerably reducing the computational burden.

However, in presence of noisy data, ML segmentation often proves unsatisfactory, having neglected any helpful contextual information, like the spatial correlation. To obtain acceptable results, one cannot rely solely on the observed data, but must take advantage of all available prior information about the image or class of images under analysis.

The Markov random field (MRF) modelling [14, 15, 51, 52] is a relatively simple, yet effective, tool to encompass prior knowledge in the segmentation process. When image segmentation is formulated as a Bayesian estimation problem, all prior information available on the image to be segmented must be contained, as already said, in the probability distribution of its segmentation map $p(x)$. By modelling the segmentation map as a MRF, *i.e.*, assuming that each given pixel X_s depends statistically on the rest of the image only through a selected group of neighbors $X_{\eta(s)}$, one simplifies the difficult problem of assigning a prior: one needs only to specify the local characteristics of the image $p(x_s|x_{\eta(s)})$. What is more important, local dependencies can be conveniently expressed through the definition of suitable potential functions in a *Gibbs* distribution, as we will see in the following of this chapter.

2.1.2 Basic Elements and Definitions

Generally speaking, the Markov Random Field represents a probabilistic model for a set of variables that interact on a lattice structure. The probability distribution for a single variable at a particular site depends on the configuration of a predefined neighborhood surrounding that site, and given such configuration it is independent of the rest of the process. This effectively defines the Markov property of the process: the process is Markov not in the causal or even the bilateral sense, but with respect to this particular neighborhood structure. Let us proceed now to give the needed definitions and basic elements of the MRF theory. To do so, let us consider a generic lattice $\mathcal{S} \equiv \{s_1, \ldots, s_N\}$ of finite dimension N.

Definition 2.1.1 (Neighborhood System) *A* neighborhood system η *on* $\mathcal{S}$ *is defined as a collection of subsets* η_s *of* $\mathcal{S}$*,*

$$\eta \equiv \{\eta_s : s \in \mathcal{S},\, \eta_f \subset \eta\}$$

where for each η_s*, neighborhood of site* s*, holds*

- s *does not belong to* η_s*;*
- $r \in \eta_s \Longrightarrow s \in \eta_r, \forall s \in \mathcal{S}$.

In other words a neighborhood system is the collection of all the local neighborhoods.

The most commonly used neighborhood systems are referred to as η^1 and η^2, where $\eta^1 = \{\eta_s^1\}$ is such that for each site, except those on the border, η_s^1

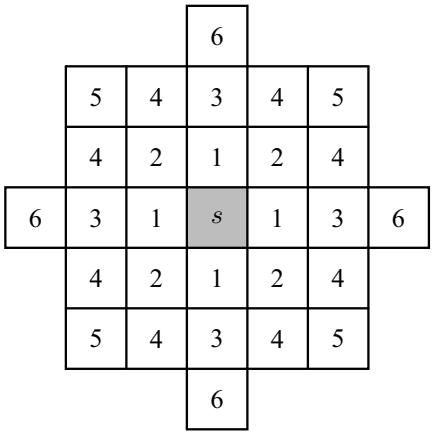


Figure 2.1: Neighborhood system $\eta^m = \{\eta_s^m\}$.

is the set of the 4 closest sites, while η^2 takes the 8 closest sites, and so on, as depicted in Fig. 2.1. η^m is said *neighborhood of order* m.

Definition 2.1.2 (Clique) *A subset $c \subseteq \mathcal{S}$ is a* clique *with respect to η if one of the following conditions is satisfied:*

- *c is a single site;*
- *every pair (r, s) of distinct sites in c are neighbors, that is:*

$$r \neq s \Longrightarrow r \in \eta_s$$

$\mathcal{C} = \mathcal{C}(\mathcal{S}, \eta)$ denotes the set of cliques with respect to $\mathcal{S}$ and η.

In Figures 2.2 and 2.3 all possible cliques corresponding to systems η^1 and η^2 are shown.

Definition 2.1.3 (Random Field) *A* random field *(RF) defined on a lattice $\mathcal{S}$ is a set of random variables $X = \{X_s\}, \ \forall s \in \mathcal{S}$.*

Notice that, said $\Omega = \Lambda^N$ the space of the realizations x of a random field X, then

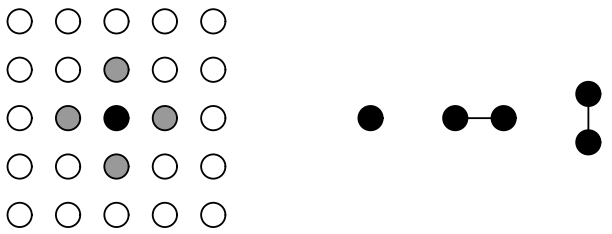


Figure 2.2: Cliques (right) for the neighborhood system η^1 (left).

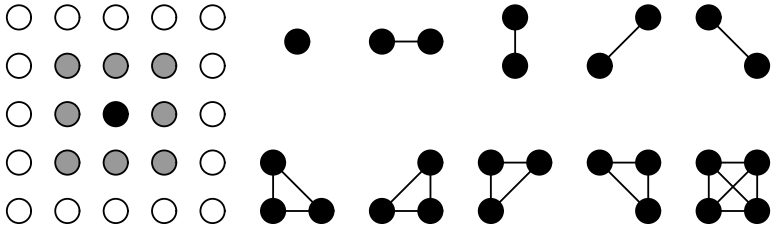


Figure 2.3: Cliques (right) for the neighborhood system η^2 (left).

$$\{X = x\} \Longleftrightarrow \{X_1 = x_1, \ldots, X_N = x_N\} \ \ \forall x \in \Omega$$

where Λ is the space of a single variable x_s, also referred as *labelling space*.

Definition 2.1.4 (Markov Random Field) *A random field X defined on a lattice $\mathcal{S}$ is a MRF with respect to a neighborhood system η if [53, 54]*

1. $p(x) > 0 \ \ \forall x \in \Omega$;
2. $p(x_s|x_r,\, r \in \mathcal{S},\, r \neq s) = p(x_s|x_r,\, r \in \eta_s)$,

for every $s \in \mathcal{S}$ and $x \in \Omega$.

The functions on the right-hand side of *2.* are called the *local characteristics* of the MRF and it turns out that the (joint) probability distribution $p(x)$ of any process satisfying (1) is *uniquely* determined by these conditional probabilities [17]. It can be shown that probability distribution of a MRF can always be written as a Gibbs distribution [17, 55], defined below.

Definition 2.1.5 (Gibbs Distribution/Gibbs Random Field) *A* Gibbs distribution *relative to a pair $\{\mathcal{S}, \eta\}$ is a probability measure π on Ω with the following representation [15]:*

$$\pi(x) \stackrel{\triangle}{=} p(X = x) = \frac{1}{Z} \exp[-\frac{U(x)}{T}] \tag{2.4}$$

where Z and T are constants and U, called the energy function*, has the form*

$$U(x) \stackrel{\triangle}{=} \sum_{c \in \mathcal{C}} V_c(x).$$

Recall that $\mathcal{C}$ denotes the set of cliques for η. Each V_c is a function on Ω with the property that $V_c(x)$ depends only on those coordinates x_s of x for which $s \in c$. The family $\{V_c, c \in \mathcal{C}\}$ is set of potentials *of the field. Z is the normalizing constant:*

$$Z \stackrel{\triangle}{=} \sum_{x \in \Omega} \exp[-\frac{U(x)}{T}]$$

and is called partition function*. Finally, T is called* temperature *for historical reasons.*

The V_c functions represent contributions to the total energy from external fields (singleton cliques), pair interaction (doubletons cliques), and so forth. The equivalence between Markov and Gibbs Random Field is provided by the following main theorem [17, 55]:

Theorem 2.1.1 (Hammersley and Clifford. MRF/GRF equivalence) *Let η be a neighborhood system. Then X is a MRF with respect to η if and only if $\pi(x) = \Pr(X = x)$ is a Gibbs distribution with respect to η.*

This equivalence provides a simple, practical way of specifying MRFs, namely by specifying the potentials V_c, which is clearly an easy task if compared with the direct specification of local characteristics.

2.1.3 The Generalized Potts Model

The *Generalized Potts Model* [56] represents a specific MRF modeling strategy that takes advantage of the simplifications coming from the Hammersly-Clifford theorem introduced in the last section. Potentials are here specified by means of a simple parametric form, as it will be clear in the following.

The core of such modeling strategy has to be searched in the well known *Ising model* [57], a very classical tool in literature that originates from the statistical mechanic theory of phase transitions. His main use in the domain of origin concerned the modeling of the behaviour of particles in ferromagnetic materials: the rationale behind it was to describe the macroscopic characteristics of a lattice material through the specification of its microscopic or intermolecular interactions.

Similarly, its transposition in the image analysis domain relies on the same concept, that is, to provide a global description of the image through the superposition of local characteristics, in terms of spatial interactions among neighboring sites. The model was first introduced for the case of binary MRFs, that is, assuming only two different values, according to the original application in physics where the local phenomena observed were the "spins" of the molecula over the lattice, each one having only two possible directions.

The extension of this model to the case of generic K-valued MRFs is called *Potts model* [52], and is completely determined by specifying the potential functions introduced in the definition 2.1.5, that in this particular case apply exclusively to the η_1 and η_2 neighborhood systems, and are defined as follows:

$$V_c(x_c) = V_c(x_p, x_q) = \begin{cases} \beta & \text{if } x_p \neq x_q, \quad p, q \in c \\ 0 & \text{otherwise} \end{cases} . \tag{2.5}$$

Single-site cliques are not used because there is no reason to favor a label over the other, and larger cliques are neglected to speed-up processing. Once given the potential functions, the global distribution $p(x)$ is completely defined, and the local characteristics $p(x_s|x_{\eta(s)})$ can be expressed [15] as:

$$p(x_s = k|x_{\eta(s)}) \propto \exp[\beta N_k] \propto \exp[-\beta N_{\bar{k}}],$$

where N_k ($N_{\bar{k}}$) is the number of neighbors of s with label k (different from k).

With this model, the vector of parameters θ associated with the prior model $p(x)$ reduces to a single parameter $\beta > 0$, interpreted as an "edge-penalty". In fact, when $\beta = 0$ all realizations are equally likely, whereas larger values of β tend to penalize non-homogeneous cliques making smoother realizations more and more likely. Of course, β is not known a priori, and must be estimated together with the map x.

It should be clear, by now, that the effect of such modeling strategy is to impose a certain regularization onto the segmentation map, with the main aim of reducing the effect of noise on the final segmentation. Such regularization is controlled by the unique β parameter, thus having the same effect over the whole image.

However, in many cases it could be preferable to vary the effect of regularization over the image, above all in presence of complex and structured data. For this reason, a further generalization of the described model has been finally proposed [56], namely the *Generalized Potts Model*, that removes the constraint of equivalence among the non-homogeneous cliques of the image and hence substitutes the β parameter with a set of $\frac{1}{2}K(K-1)$ parameters β_{hk}, one for each different label coupling within a clique of the map ($h, k \in \Lambda$ and $h \neq k$, with $\beta_{hk} = \beta_{kh}$). Formally:

$$V_c(x_c) = V_c(x_p, x_q) = \begin{cases} \beta_{hk} > 0 & \text{if } x_p = h \neq k = x_q, \quad p, q \in c \\ 0 & \text{otherwise} \end{cases}, \tag{2.6}$$

and hence local characteristics can be expressed as:

$$p(x_s = k|x_{\eta_s}) = \frac{1}{Z} \exp[-\sum_{h \neq k} \beta_{hk} N_h]. \tag{2.7}$$

with Z being a normalizing constant.

2.1.4 Optimization Methods

Assuming that the prior and likelihood models are fully specified, as already stated above, the problem of segmentation relies on the maximization of the product $p(x)p(y|x)$ over x. The value of x corresponding to the maximum posterior probability, say $\hat{x}$, is the desired segmentation map. This optimization process lies on the fundamental statement that the posterior distribution can be itself written in a Gibbs-MRF form, as shown in the following.

Let us recall that, for the hypothesis of spatial indipendence, the likelihood model can be written as a product of conditional local distributions (see Eq. 2.2), here explicited in a logarithmic form:

$$\ln p(y_s|x_s = k) = -\frac{B}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma_k| - \frac{1}{2}(y_s - \mu_k)^T\Sigma_k^{-1}(y_s - \mu_k).$$

Let us define now the following singleton-clique potential functions which associate, pixel-by-pixel, the label field with the *external* observation field[3]:

$$V_s''(x_s = k) \stackrel{\triangle}{=} \frac{1}{2}\ln|\Sigma_k| + \frac{1}{2}(y_s - \mu_k)^T\Sigma_k^{-1}(y_s - \mu_k).$$

Accordingly, Eq. 2.3 can be more compactly written as

$$p(y|x) = \frac{1}{Z''}\exp[-\sum_{s\in\mathcal{S}} V_s''(x_s)] = \frac{1}{Z''}\exp[-U''(x)], \qquad (2.8)$$

where $Z'' = (2\pi)^{-NB/2}$. Finally, thanks to the Bayes formula (see Eq. 2.4 and 2.8), the posterior distribution can be written as[4]

$$p(x|y) = p(x)p(y|x) = \frac{1}{Z}\exp[-U(x)] = \frac{1}{Z}\exp[-\sum_{c\in\mathcal{C}} V_c(x_c)], \qquad (2.9)$$

where, if we write the prior $p(x)$ as

$$p(x) = \frac{1}{Z'}\exp[-sum_{c\in\mathcal{C}}V_c'(x_c)],$$

then $Z = Z'Z''p(y)$ and

[3] We neglect y_s as an explicit argument of V_s'' since it is known and does not represent a variable to be estimated in the segmentation process.

[4] Notice that $\mathcal{S} \subseteq \mathcal{C}$, since every site $s \in \mathcal{S}$ is a clique for each arbitrary neighborhood system.

$$V_c(x_c) = \begin{cases} V'_c(x_c) + V''_c(x_c) & \text{if } c \text{ is a singleton clique} \\ V'_c(x_c) & \text{otherwise} \end{cases}, \qquad (2.10)$$

which has still a Gibbs-MRF form with the same neighborhood system as the prior field of Eq. 2.4 and modified singleton-clique potentials. Usually, single-site potentials are neglected since they do not carry any contextual information, while the data-dependent potentials V''_s are strictly associated with the observations.

The Iterated Conditional Modes (ICM) Algorithm

In the form of an energy minimization problem (maximizing the posterior distribution defined in Eq. 2.9 corresponds to minimizing the energy $U(x)$), optimization can be finally carried out by means of one of the many techniques known in literature that deal with this kind of problem.

A possibly optimal solution is represented by the *Simulated Annealing* (SA) technique [15], based on the analogy between the annealing of solids and the solving of combinatorial optimization problems. The Gibbs distribution is here put in a parametric form using a "temperature" value T:

$$p(x) = \frac{1}{Z(T)} \exp[\frac{-U(x)}{T}],$$

and an optimization process is run that iteratively comutes current minimum for $U(x)$ for certain fixed values of T, according to a suitable *cooling schedule*, that is, starting from a sufficiently high initial temperature value that decreases at each step until the system is frozen (no relevant decrease of the energy happen).

The algorithm is initialized with a random guess of the unknown x. Clearly, as the temperature decreases, the above distribution concentrates on the states with lower energy and when the temperature approaches zero, only the minimum energy states have a non-zero probability. Optimality of the process, that is, retrieving the global minimum of $U(x)$, is guaranteed if a sufficiently slow cooling schedule is applied.

In the case of image segmentation, using SA to maximize the posterior probability is often unfeasible in practice due to the excessive computational complexity, even if sub-optimal variants of SA are considered that make use of faster cooling schedules. If we have a reasonably good initial configuration x_0 then a rapid convergence can be obtained by the ICM method proposed by

Besag in [14] (it will be extensively employed, in the algorithms presented in this work). The quality of the final result strongly depends on the initialization since ICM realizes only a descent in the nearest energy-valley and energy functionals are generally non-convex. Of course, the obtained minimum is only *local* but convergence towards this minimum is usually obtained in a few number of iterations.

Algorithm 2.1.1 (ICM)

1. *Start at a "good" initial configuration x^0 and set $k = 0$.*
2. *For each configuration which differs at most in one element from the current configuration x^k (they are denoted by $\mathcal{N}_{x^k}$), compute the energy $U(\eta)$ ($\eta \in \mathcal{N}_{x^k}$).*
3. *From the configurations in $\mathcal{N}_{x^k}$, select the one which has a minimal energy:*
$$x^{k+1} = \arg \min_{\eta \in \mathcal{N}_{x^k}} U(\eta) \tag{2.11}$$
4. *Go to Step 2. with $k = k+1$ until convergence is obtained (for example, the energy change is less than a certain threshold).*

Notice that in the ICM algorithm there is no temperature parameter and thus there is no annealing.

Estimation of Parameters

As it should be clear from the description of the framework made in the previous section, the segmentation problem is here characterized by a certain number of important parameters such as the number K of labels/classes in the image, the class-related parameters μ_k and Σ_k of the likelihood term $p(y|x)$ (see Eq. 2.3, and the parameters β_{hk} of the Gibbs prior $p(x)$. In the simplest case where all these parameter values are known in advance, *i.e.*, in a fully *supervised* mode, all we have to do is run the ICM procedure described in the last section to find the segmentation map. Quite often, however, some or all of this parameters are not known, resorting to respectively a *semi-supervised* or *unsupervised* segmentation, and must be estimated from the data together with the segmentation $\widehat{x}$ itself.

The single most critical parameter is by far the number of classes K, since it influences heavily all other aspects of segmentation. The problem of determining the number of classes in a data set, or *cluster validation problem*, has

received a great deal of attention in the literature [49], with mixed and inconclusive results. As a matter of fact, in a real-world image, the number of different segments that can be identified varies wildly according to the user's point of view. In a remote-sensing image, for example, a single segment labeled as "urban area" in one application, could be further partitioned into smaller segments in another application. In the absence of prior information on the application, both solutions are equally reasonable, and both should be preserved to let a human interpreter have a final say.

Although some efficient strategies have been proposed to address the cluster validation problem, this is still one of the main reasons for the increase in complexity going from supervised to unsupervised segmentation.

Another reason is the need to estimate, together with the segmentation, the parameters of the involved distributions, collectively represented by a random vector Θ:

$$(\widehat{x}, \widehat{\theta}) = \arg\max_{x,\theta} p(x, y|\theta). \tag{2.12}$$

Since exact joint optimization is computationally intractable, a two-step procedure is often used. First, the model parameters are estimated from the observed data, following for example an ML approach, then the MAP segmentation is carried out in a second step using the estimated parameter values. A number of techniques can be used to perform the ML parameter estimation, such as the EM algorithm and its numerous variants, or the similar but more general ICE [58]. Except for some simple cases, however, these algorithms do not have an analytical closed form, and are quite computationally expensive. For this reason, we here consider a suboptimal, but much simpler, alternating marginal optimization ($\widehat{x}$ and $\widehat{\theta}$ are alternately optimized given each other) which can be viewed as an approximation of the two step EM-approach [51], and has been observed to provide comparable results in various practical situations [59].

2.2 The Tree Structured MRF Model

As it should be emerged also from the modeling approach presented in the previous section, several issues have to be accurately studied when using MRFs for image segmentation, the most important being:

1. how to define a MRF (through its potentials) that is able to take into

account prior information while remaining mathematically and numerically manageable;

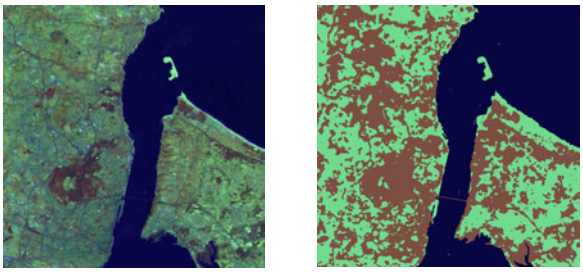
2. how to set/estimate the numerical parameters of such a MRF;

3. how to solve the MAP estimation problem with reasonable computational complexity.

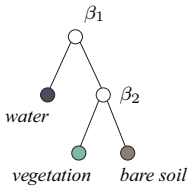
The first problem is certainly the most intriguing, as it amounts to defining an abstract structure of the image that fits well the observed data. The Potts Model (see Eq. 2.5) is of course an easy and effective solution, but in general one could be tempted to define more sophisticated models in order to capture the complex nature of image dependencies. However, model definition cannot overlook the estimation problems (2) and (3). In fact, by increasing the model complexity, for example resorting to the Generalized Potts Model of Sec. 2.1.3, one ends up with a large number of parameters that are more difficult to be reliably estimated; and even neglecting this problem, the subsequent optimization task could be so computationally demanding as to forbid the use of reliable procedures, leading to disappointing results. Indeed, computational complexity remains a major weakness of the MAP/MRF approach, and in developing a real-world MRF-based segmentation algorithm all efforts should be made to keep it under control, without sacrificing fidelity of description.

In the following, we will give motivations and discuss in some detail a new family of MRF models, namely the *Tree Structured Markov Random Field* (TS-MRF) models, that stem from the idea of reducing overall complexity by introducing structural constraints over a MRF-based image modeling, while trying at the same time to preserve the quality of description guaranteed by the adaptivity to "local" characteristics provided by the Generalized Potts Model.

The Tree Structured Markov Random Field modeling has been first introduced in [60], where the authors applied it to the context of unsupervised segmentation and proposed a solution to the cluster validation problem, and was originarily inspired by the work of Fwu and Djuric [61] that proposed a tree structured variant of the ICM algorithm. In further works [18, 1] a deeper insight in the theoretical aspects of the model is provided, along with the presentation of several different application to supervised and unsupervised classification, mainly in the remote sensing domain.



(a) (b)



(c)

Figure 2.4: *TS-MRF motivations* : (a) a simply structured remote sensing image (false color representation), (b) a possible coherent segmentation map, (c) scene desctiption through a hierarchical tree structure.

2.2.1 Structuring a MRF: the Generalized Potts Model case

To better understand the fundamental hypothesis on which the TS-MRF models lie, let us immediately consider the real data example of Fig. 2.4: here, a generic low resolution (around 10 m) remotely sensed scene is presented in (a), evidently characterized at a fine observation degree by three different land cover classes, each one with quite homogeneous spectral properties, namely the *water*, *vegetation* and *bare soil* classes. A possible 3-class segmentation map is depicted in (b). For this image, the hierarchical structurability of data according to the relationships among classes is quite evident if we observe the segmentation map, where it is clearly reasonable to consider a first class coupling, at a "coarser" scale, between the coverage corresponding to the water and the land, and then, within the latter, a "finer" coupling between the two different types of land cover.

More technically speaking, it can be observed that the *vegetation* and *bare*

soil classes share the same spatial interaction with the *water* class, since there's no significant statistical difference between a *green-blue* edge and a *brown-blue* one on the map. Considering now the *Generalized Potts model* framework, where the potentials are expressed as in Eq. 2.6, we easily realize that in this case there is no use taking into account two different estimations for the two parameters associated to the aforementioned couplings. A single estimation could suffice to provide a reliable parametric specification of the MRF model.

This interesting property can be efficiently expressed by means of a hidden hierarchical tree structure, like the one of Fig. 2.4(c) for the example under analysis: the two relevant parameters can here be associated to each inner node t, one (β_1 in figure) at the root level that controls the *split* between water and land, and another one (β_2) at the deeper level controlling the separation between vegtation and soil. This implicitly defines also a strict hierarchical relationship among the different regions of the image identified by class labels.

A *Tree-Structured Markov Random Field* represents a modeling tool that allows for an efficient representation of the image that takes into account of this kind of structural properties: its complete definition is given, in the general case, through a "representative" tree-structure of the image, and a corresponding set of classical (flat) MRFs, each one associated to a specific inner node of the tree, hence *local* to some region of the image and of reduced dimensionality w.r.t. the total number of its classes. Back to the Generalized Potts model framework, reduction in the number of MRF parameters to estimate by imposing the described structural constraint is significant: for a generic K-class segmentation, supposing the use of a simple Potts MRF (with a single parameter to estimate) for each inner node of the tree, in the worst case of a binary tree structure[5] we have to estimate $K-1$ parameters instead of the $\frac{1}{2}K(K-1)$ originarily required by the Generalized Potts model.

More in general, even considering non-isotropic models and/or more sophisticated cliques, one gets the same parameter reduction ratio ($K/2$) between a complete unconstrained model and the "tree-structured" dual one. Moreover, looking at the estimation problem, it is worth considering that if the data can be well represented by this kind of structure, the information available to estimate its few parameters will increase, eventually resulting also in better estimates.

[5] Such case is here considered to be the worst exclusively w.r.t. the number of paramenters to estimate, since binary trees contain the maximum number of inner nodes once fixed the number of leaves.

2.2.2 Theoretical Binary TS-MRF

In its original formulation, the TS-MRF model was introduced with the additional constraint of taking into account only *binary* tree structures. This choice was originally justified by the fact that such structure presents the highest number possible of inner nodes, hence providing the richest parametrization given the structural constraint, under the hypotesis of using simple Potts MRFs (*i.e.*, each one defined using a single edge-penalty parameter) for each inner node. For this reasons, the basic theory behind TS-MRF, discussed in the following, has been developed according to this binary constraint, but can be easily extended to the case of generic tree structures, as it will be outlined briefly in the next chapter.

Let us first define a theoretical tree-structured MRF model, and later the actual implementation of the model. To this end, let us consider a binary tree T, identified by its nodes and their mutual relationships. Except for the root, each node t has one parent $u(t)$, and each internal node has two children $l(t)$ and $r(t)$, with $u[l(t)] = u[r(t)] = t$. We also define $\widetilde{T} = \{t \in T : l(t) = r(t) = \emptyset\}$, the set of terminal nodes or leaves, and $\overline{T} = T - \widetilde{T}$, the set of internal nodes.

Integer numbers are used to index the nodes of the tree, as well as all items associated with them, so that root $= 1$, $l(t) = 2t$, $r(t) = 2t + 1$ and $u(t) = \lfloor t/2 \rfloor$ (see Fig.2.5). Note that each terminal node corresponds to a class, while each internal node corresponds to both a merging class and an edge-penalty parameter. In order to define the model it is helpful to use the binary representation of the indexing integers. Let $\nu(t)$ be the function that converts a non negative integer $t \in \mathbb{N}$ to its corresponding variable-length binary code $c \in \mathcal{B}$, where all leading zeros are discarded (see the balanced tree of Fig.2.5), and let ℓ_t be the corresponding length.

Let us also define the function $\Psi(a, b) : \mathcal{B} \times \mathcal{B} \rightarrow \mathcal{B}$ which returns the longest common prefix of a and b. It's easy to check that $\Psi(a, b)$ gives the nearest common ancestor node of a and b.

We now define a tree-structured MRF through its local characteristics, still expressed by Eq. 2.7, but with the additional $\frac{1}{2}K(K-3)+1$ constraints:

$$\beta_{kh} = \beta_{pq} = \beta_t \tag{2.13}$$

for (k, h) and (p, q) such that

$$\Psi(\nu(k), \nu(h)) = \Psi(\nu(p), \nu(q)) = \nu(t),$$

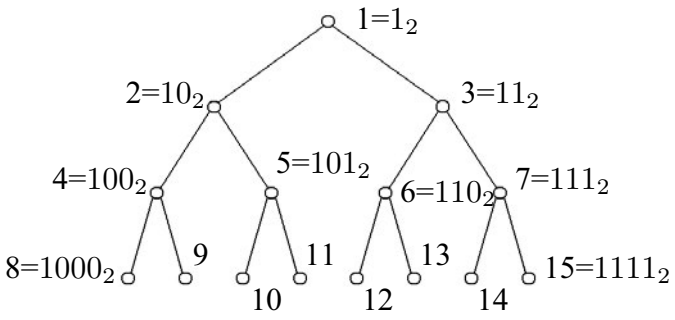


Figure 2.5: Tree indexing.

with $k \neq h$, $p \neq q$, $(k,h) \neq (p,q)$.

Reorganizing the terms in Eq. 2.7 we can explicit the local characteristics with respect to the non-redundant parameter set $\{\beta_t\}_{t\in\overline{T}}$ as follows:

$$p(x_s = k|x_{\eta(s)}) = \frac{1}{Z}\exp[-\sum_{n=1}^{\ell_k-1}\beta_{\nu^{-1}(k_1,\ldots,k_n)}N_{\nu^{-1}(k_1,\ldots,k_n,\overline{k}_{n+1})}],$$

with $\nu(k) = (k_1,\ldots,k_{\ell_k})$. Here, when h corresponds to an internal node, $x_s = h$ means that s belongs to one of the descendant classes of h, and N_h is the number of neighbours of s which belong to one of such classes. For example, with reference to the tree of Fig. 2.4(c), we have

$$\begin{aligned} p(x_s = 10|x_{\eta(s)}) &= \frac{1}{Z}\exp[-\beta_1 N_3 - \beta_2 N_4 - \beta_5 N_{11}] \\ &= \frac{1}{Z}\exp[-\beta_1(N_{12}+N_{13}+N_{14}+N_{15}) + \\ &- \beta_2(N_8+N_9) - \beta_5 N_{11}]. \end{aligned}$$

The clique potentials are expressed by

$$V_c(x_c) = V_c(x_p,x_q) = \begin{cases} \beta_{\nu^{-1}(\Psi(\nu(k),\nu(h)))} & \text{if } x_p = k \neq x_q = h, \quad p,q \in c \\ 0 & \text{otherwise} \end{cases}.$$

Now we define the function $\mathcal{N}_t(x)$ which gives the number of cliques in the map x with edge-penalty β_t. With this position, the joint probability of the TS-MRF becomes simply:

$$p(x) = \frac{1}{Z} \exp[-\sum_{t \in \overline{T}} \beta_t \mathcal{N}_t(x)]. \tag{2.14}$$

The complexity of this model could still seem prohibitive for a practical implementation because of the dimensionality of the parameter space, dependent on the number of classes, that makes very hard the optimization. However, thanks to the structural constraints of the model, a recursive optimization procedure can be used which, although sub-optimal, involves only one edge-penalty at a time.

2.2.3 Recursive Optimization

Let us consider for each node t of a tree T,

- a set of sites $\mathcal{S}^t \subseteq \mathcal{S}$, corresponding to a segment of the image (in particular $\mathcal{S}^{\text{root}} = \mathcal{S}$);
- a binary random field $X^t = \{X_s^t : \ s \in \mathcal{S}^t\}$, with realization x^t where $x_s^t \in \{l(t), r(t)\}$.

Now we impose the additional constraint that the set of sites associated with any given node is obtained from the binary segmentation of the parent set of sites. More formally, for each internal node of the tree $t \in \overline{T}$,

$$\begin{cases} \mathcal{S}^{l(t)} &= \{s \in \mathcal{S}^t : \ x_s^t = l(t)\} \\ \mathcal{S}^{r(t)} &= \{s \in \mathcal{S}^t : \ x_s^t = r(t)\} \end{cases} \tag{2.15}$$

Therefore, the tree-structured MRF X is completely given by the set of binary fields $\{X^t\}_{t \in \overline{T}}$ and *vice-versa*, that is:

$$X = \bigcup_{t \in \overline{T}} X^t.$$

Let us define, now, $\omega(t) = \{h \in \overline{T} - \{t\} : \nu(h) \text{ is a prefix of } \nu(t)\}$, the set of the ancestor nodes of t, and $X^{\omega(t)} = \{X^t\}_{t \in \omega(t)}$, the set of the ancestor fields of t (of course, $\omega(1) = X^{\omega(1)} = \emptyset$). Observe that, except for X^1, each field X^t depends on the ancestor fields $\{X^{\omega(t)}\}$, in particular, the very same

domain of X^t is fixed once the ancestor fields are specified. On the other hand, given a realization $x \equiv \{x^k\}_{k\in\overline{T}}$, the number $\mathcal{N}_t = \mathcal{N}_t(x)$ of cliques with edge-penalty β_t depends only on x^t and, for the above considerations, on $x^{\omega(t)}$, while it is independent of other component binary fields. As a consequence, the joint probability of the overall field (Eq. 2.14) becomes:

$$\begin{aligned} p(x) &= \frac{1}{Z}\exp[-\sum_{t\in\overline{T}}\beta_t\mathcal{N}_t(x^t, x^{\omega(t)})] \\ &= \prod_{t\in\overline{T}}\frac{1}{Z_t}\exp[-\beta_t\mathcal{N}_t(x^t, x^{\omega(t)})]. \end{aligned} \tag{2.16}$$

It is also easy to prove that, for each node in the tree, given X^t and $X^{\omega(t)}$, the set of fields which lie on the left sub-tree stemming from t is independent from the set of fields which lie on the right sub-tree. As an example, for the structure in Fig.2.5 we can write:

$$\begin{aligned} p(x^5, x^4, x^2|x^7, x^6, x^3, x^1) &= \frac{p(x)}{p(x^7, x^6, x^3, x^1)} \\ &= \frac{\frac{1}{Z}\exp[-\sum_{t=1}^{7}\beta_t\mathcal{N}_t]}{\sum_{x^5,x^4,x^2}\frac{1}{Z}\exp[-\sum_{t=1}^{7}\beta_t\mathcal{N}_t]} \\ &= \frac{\frac{1}{Z}\exp[-\beta_2\mathcal{N}_2 - \beta_4\mathcal{N}_4 - \beta_5\mathcal{N}_5]}{\sum_{x^5,x^4,x^2}\frac{1}{Z}\exp[-\beta_2\mathcal{N}_2 - \beta_4\mathcal{N}_4 - \beta_5\mathcal{N}_5]} \\ &= \frac{1}{Z(x^1)}\exp[-\beta_2\mathcal{N}_2 - \beta_4\mathcal{N}_4 - \beta_5\mathcal{N}_5] \quad (2.17) \\ &= p(x^5, x^4, x^2|x^1), \quad (2.18) \end{aligned}$$

which proves the independence. In a similar way, it can be proved that $p(x^4|x^5, x^2, x^1) = p(x^4|x^2, x^1)$ and so on. More in general, thanks to the above property, by a recursive use of the Bayes theorem we have:

$$p(x) = \prod_{t\in\overline{T}} p(x^t|x^{\omega(t)}). \tag{2.19}$$

Note also that, given the ancestor field X^1, the field built on the sub-tree with root in $t = 2$, (X^2, X^4, X^5), is still a TS-MRF (see Eq. 2.17); this property holds for each internal node t as well. As a consequence, given $X^{\omega(t)}$, the

terminal binary fields X^t (associated with terminal splits) are Potts MRFs, that is

$$p(x^t|x^{\omega(t)}) = \frac{1}{Z(x^{\omega(t)})} \exp[-\beta_t \mathcal{N}_t].$$

This property does not hold for non-terminal binary fields, because, in this case, the partition function Z is itself a function of x^t. For example we have:

$$\begin{aligned} p(x^2|x^{\omega(2)}) &= \sum_{x^5,x^4} p(x^5, x^4, x^2|x^1) \\ &= \frac{1}{Z(x^1)} \exp[-\beta_2 \mathcal{N}_2] \sum_{x^5,x^4} \exp[-\beta_4 \mathcal{N}_4 - \beta_5 \mathcal{N}_5] \\ &= \frac{1}{Z(x^1)Z(x^1,x^2)} \exp[-\beta_2 \mathcal{N}_2]. \end{aligned}$$

In other words, not all the terms of Eq. 2.19 are Potts distributions, as one could believe for the similarity between Eq. 2.16 and Eq. 2.19. Nonetheless, in order to find a MAP estimate of a segmentation with TS-MRF prior probability, one can recursively maximize the terms in Eq. 2.16, together with the likelihood parts, starting from the root and descending the tree until all leaves are reached. Each term depends only on a binary field X^t once its ancestor fields $x^{\omega(t)}$ are given and, also, it does have a Potts form.

As a consequence, each one can be maximized, just like with an ordinary Potts MRF, by using simulated annealing, ICM, etc. Note, again, that in the step corresponding to node t, only the parameter β_t must be estimated, and that $\mathcal{N}_t$ is a sufficient statistic for β_t. Therefore, when the prior parameters are unknown, estimation-maximization procedures can be used again following a recursive schedule.

Finally, we underline that each binary field X^t, except for the root field, makes sense only once the realization $x^{\omega(t)}$ of its ancestor fields are given, since it is defined on an irregular (that is, non-rectangular) lattice whose shape is a result of $x^{\omega(t)}$.

2.3 Unsupervised Segmentation using TS-MRFs

In the model-based framework, the unsupervised segmentation task if often split in two part [62]. The former is the *cluster validation problem*, where

the goal is to detect the number K of classes/regions present into the image and, for each of such classes, to provide some features that summarize the region properties. The latter can be seen as a semi-supervised segmentation, where some model parameters may come out from the former step. Besides such point of view, other approaches may follow a joint solution to address an unsupervised segmentation problem [49].

In particular, most of the MRF model-based algorithms refer to a disjoint approach, since K is needed before an optimization procedure could proceed, or else they update progressively the number of classes while the optimization procedure goes on. TS-MRF models represent an exception since their recursive nature, that fits with a recursive step-by-step optimization, naturally allows an incremental update of K that suggests a joint solution for clustering data and segmentation. In fact, the problem of estimating K is strictly related with the problem of finding the structure that supports the model, and they may be addressed in a simple way by controlling the growth of the tree, thanks to a test local to each node that indicates whether or not it must be split.

Now, let us focus on the description of such a TS-MRF unsupervised segmentation algorithm, proposed in its basic form in [60] and later refined in [18].

2.3.1 The *Split Gain* and the Recursive Tree Growth

The unsupervised TS-MRF based algorithm has a recursive nature, it starts with a single-node tree which grows leaf by leaf until a stopping condition is met. Therefore, we first describe the algorithm initialization, focusing on the root (node 1), and then the generic step with reference to a given tree.

At the beginning we consider the following two hypotheses (see Fig. 2.6 for indexing):

$$\begin{cases} H_0: & T = \{1\}, & X = \emptyset \\ H_1: & T' = \{1,2,3\}, & X = \widetilde{x}^1 \end{cases} . \tag{2.20}$$

The first hypothesis corresponds to the case in which the whole image, associated with the root node ($\mathcal{S}^1 = \mathcal{S}, y^1 = y$), is represented as a single region. Therefore, the observed data are described by a single distribution $p(y^1)$, whose model is known but for some parameters ν^1 that must be estimated from the data themselves. Of course, in this case the TS-MRF is empty, and all sites have the same label attached. This is the only possible configuration and in this sense we define $p(x|T = \{1\}) = 1$, and also write the data distri-

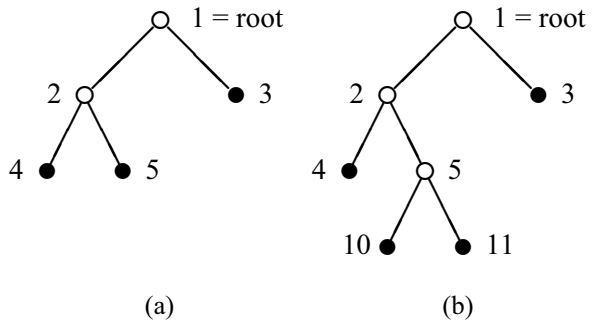


Figure 2.6: A simple binary tree (a), and the tree resulting from splitting node 5 (b).

bution as $p(y^1|\nu^1)$ to make explicit that y^1 is described through the single set of parameters ν^1 attached with node 1.

The second hypothesis corresponds to the case in which the image is represented by two regions. To single out such regions, a binary MRF X^1 is defined on $\mathcal{S}^1$, with a given neighborhood system η^1, and with potentials $V_c^1(\cdot)$ that are completely specified except for some parameters θ^1. The MAP (or any other criterion) estimate of the MRF x^1, with probability $p(x^1)$, divides the image into two new regions, $\mathcal{S}^2 = \{s \in \mathcal{S}^1 : x_s^1 = 2\}$ and $\mathcal{S}^3 = \{s \in \mathcal{S}^1 : x_s^1 = 3\}$, with their associated data y^2 and y^3. Also, since we assumed conditionally independent data, their description factors out as $p(y^1|x^1) = p(y^2|\nu^2)p(y^3|\nu^3)$.

At this point, we compare the two statistical descriptions of the image, based on a single-class model (tree T) or a two-class model (tree T'), by checking the condition

$$G^1 = \frac{p(y, x|T')}{p(y, x|T)} = \frac{p(x|T')p(y|x, T')}{p(x|T)p(y|x, T)} > 1, \tag{2.21}$$

which, specialized for $T = \{1\}$, becomes:

$$G^1 = \frac{p(x^1)}{1} \times \frac{p(y^1|x^1)}{p(y^1|\nu^1)} > 1. \tag{2.22}$$

If the test succeeds, namely the split gain G^1 is greater than 1, the two-region description better fits the data and the procedure goes on, otherwise it stops and the single-region description is accepted.

Let us now consider a generic tree T, that has been temporarily accepted as our structure, with associated TS-MRF X, and let τ be a leaf of T that we are testing for a possible split. The two hypotheses under test are then:

$$\begin{cases} H_0: & T, & X = \{x^t\}_{t\in\overline{T}} \\ H_1: & T' = \text{split}(T,\tau), & X = \{x^t\}_{t\in\overline{T}'} \end{cases}, \tag{2.23}$$

where tree $T' = \text{split}(T,\tau)$ is identical to T except for node τ which generates two new leaves becoming itself an internal node, that is $\overline{T}' = \{\overline{T},\tau\}$ (see Fig. 2.6). To explicit the test of Eq. 2.21 for the general case, remember that $p(x) = \prod_{t\in\overline{T}} p(x^t|x^{\omega(t)})$. Moreover $p(y|x) = \prod_{t\in\Lambda} p(y^t|\nu^t)$. Therefore, we can write

$$\begin{aligned} p(x|T) &= \prod_{t\in\overline{T}} p(x^t|x^{\omega(t)}) \\ p(x|T') &= \prod_{t\in\overline{T}'} p(x^t|x^{\omega(t)}) = p(x^\tau|x^{\omega(\tau)}) \prod_{t\in\overline{T}} p(x^t|x^{\omega(t)}) \\ p(y|x,T) &= \prod_{t\in\Lambda} p(y^t|\nu^t) = p(y^\tau|\nu^\tau) \prod_{t\in\Lambda-\{\tau\}} p(y^t|\nu^t) \\ p(y|x,T') &= \prod_{t\in\Lambda'} p(y^t|\nu^t) = p(y^\tau|x^\tau, x^{\omega(t)}) \prod_{t\in\Lambda-\{\tau\}} p(y^t|\nu^t) \end{aligned} \tag{2.24}$$

and the test becomes simply

$$G^\tau = \frac{p(x^\tau|x^{\omega(\tau)})}{1} \times \frac{p(y^\tau|x^\tau, x^{\omega(\tau)})}{p(y^\tau|\nu^\tau)} > 1. \tag{2.25}$$

It should be noted that the test depends exclusively on region $\mathcal{S}^\tau$. In fact, given $\{\widetilde{x}^t\}_{t\in\overline{T}}$ the maximization process operates only on x^τ, and the MAP problem reduces to :

$$\widetilde{x}^\tau = \arg\max_{x^\tau} p(x^\tau|\widetilde{x}^{\omega(\tau)}) p(y^\tau|x^\tau, \widetilde{x}^{\omega(\tau)}) \tag{2.26}$$

completely local to node τ. If the test succeeds, the growth of the tree and of the associated segmentation continues in a similar way for each newly created leaf, as if each one were the root of a new tree. Therefore, the tree growing process is accurately described by a recursive procedure, which can go on in parallel for each node.

The ratio G^τ, named split gain, accounts for the gain in description efficiency arising from the split of leaf τ. This interpretation becomes more compelling if we take the logarithm of G^τ and regard it as the difference $\log G^\tau = I(T) - I(T')$ between the self-information associated with each of the competing TS-MRF's[6]. If the self-information is a good indicator of the description complexity, then a positive log split gain indicates that the new description of the observed data is "simpler" than the preceding one, and hence preferable (according to Occam's razor). In more detail, a split has always a cost, $p(\widetilde{x}^t) < 1$, due to the need of describing the segmentation $\widetilde{x}^t$, but also a value, $p(y^t|\widetilde{x}^t)/p(y^t|\nu^t) > 1$, because the data are more accurately represented, in each new segment, by their local parameters. A positive $\log G^t$ indicates that the overall benefits outweigh the cost. Analogies can be found in [63] where the *Minimum Description Length* (MDL) criterion is proposed. It must be underlined, however, that the evaluation of the split gain involves an intractable partition function and that only an approximation of it, possibly inaccurate, will be available in any practical implementation.

2.3.2 The Unsupervised TS-MRF Algorithm

Fig. 2.7 shows a high-level flow chart of the TS-MRF model-based unsupervised segmentation algorithm. To improve readability, the procedure is sequential rather than recursive, and only one leaf at a time is split, the one with the largest split gain (the experiments will follow this convention as well).

- In the initialization step, the tree is defined as consisting of the sole root ($T = \{1\}$); the whole image is associated to it ($\mathcal{S}^1 = \mathcal{S}, y^1 = y$), and the vector of parameters $\widehat{\nu}^1$ is estimated; of course, the TS-MRF is empty ($X = \emptyset$).
- In the procedure CheckNode(t), the binary MRF X^t is defined on $\mathcal{S}^t$, the MAP realization $\widetilde{x}^t$ is estimated together with its parameters $\widehat{\theta}^t$, and the split gain G^t is evaluated. If $G^t > 1$ this node will be split sooner or later.
- SplitTree(t) updates the structure of the tree by moving t from Λ to $\overline{T}$, and generating two new leaves $l(t)$ and $r(t)$; to each one of such new nodes the proper quantities ($\mathcal{S}^{l(t)}, y^{l(t)}, \widehat{\nu}^{l(t)}$, etc.) are associated (they were evaluated during the CheckNode step).

[6]This discussion is only to gain insight about the meaning of the split gain, and there is no attempt to be rigorous.

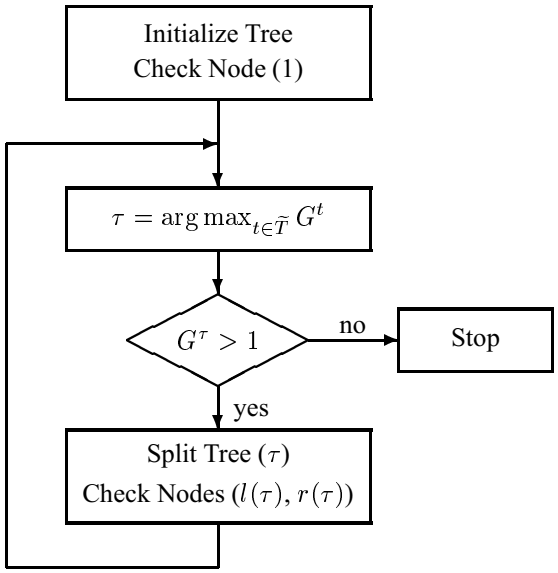


Figure 2.7: High-level flow chart of the unsupervised TS-MRF algorithm.

This procedure provides a fast segmentation of the image, based only on binary decisions, and solves automatically the cluster validation problem.

Finally, an example of how the described algorithm works on real life images is now presented, showing just a simple experiment on a 128×128 remote-sensing GER hyperspectral image composed of 6 bands selected among the 63 of a whole set. In Fig. 2.8 are shown: a band of the selected group; a Potts model-based segmentation as reference; the partial segmentations of the TS-MRF algorithm, whose associated structure is depicted in Fig. 2.9.

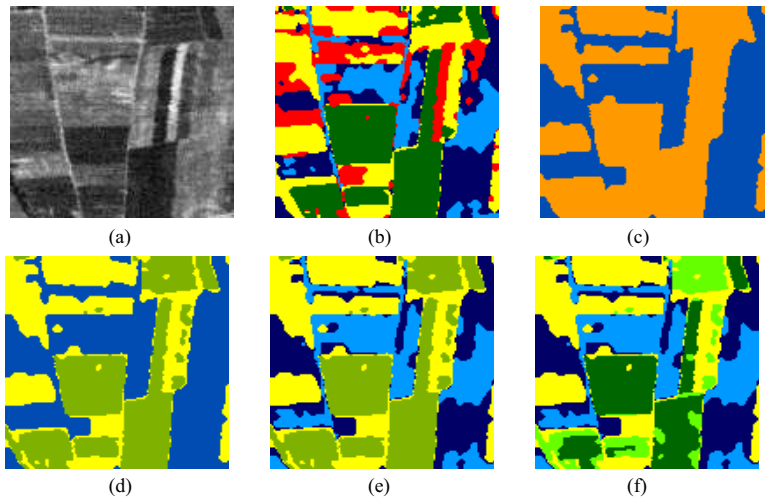


Figure 2.8: An example of unsupervised segmentation by a TS-MRF: (a) band 7 of the GER data; (b) Potts model-based segmentation; (c)-(f) partial segmentations of the TS-MRF algorithm.

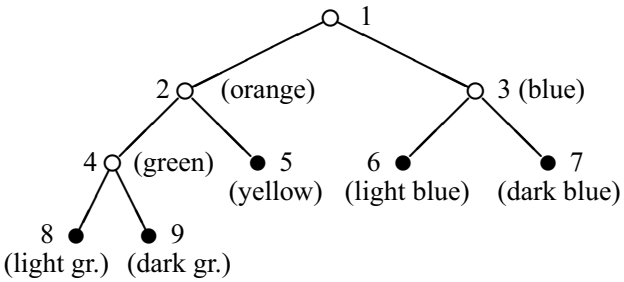


Figure 2.9: Tree structure associated with the experiment of Fig. 2.8

Chapter 3

Mean Shift Clustering applied to Unsupervised TS-MRF

In this chapter, we first recall the basics of Mean-Shift analysis, and then describe the new Fast Mean Shift Clustering (FMSC) algorithm, focusing in turn on the variable-bandwidth strategy, and on the speed-up solutions introduced. Hereinafter, we show how the new clustering tool can be used to improve the performance for unsupervised segmentation tasks and present the modified version of the unsupervised TSMRF algorithm presented in Sec. 2.3. Finally, experimental evidence of the improved performances of the new algorithm is carried out.

3.1 Introduction to Mean Shift

The Mean Shift procedure is a *mode detection* method for density functions that lies on the most popular non-parametric density estimation technique, known in the pattern recognition literature as the *Parzen Window* method [49]. Mean Shift was first introduced in 1975 by Fukunaga and Hostetler [64] as a technique for the estimation of probability density gradients, but only recently [65, 66, 19, 67] the advantages of such approach both in density estimation and clustering has been newly recognized.

As for the non-parametric density estimation techniques, the main idea on which this approach is based lies on the fact that samples in an arbitrary feature space can be seen as an empirical probability density function, that is, local maxima of the probability should be observed in areas that have a dense concentration of data points. Following this rationale, a kernel-based mode

seeking technique is proposed in [19], where the main contribution has been given by showing that such technique is robust, *i.e.* the proposed procedure is demonstrated to converge to some stationary point of the unknown density function, and general, it is applicable for the analysis of complex multimodal feature spaces.

In the next subsections, the fundamentals of such technique are presented and the algorithmic procedure for mode retrieving is finally delineated.

3.1.1 From Kernel Density Estimation to Mean Shift

Let us first recall the theoretical basis below the reference kernel density estimation technique. The basic approach in the Parzen Window technique lies on the observation that, given a d-dimensional feature space and a set of n data points $(s_1, \cdots, s_n)$, the probability density function $p(s)$ can be estimated as

$$\hat{p}_{H,K}(s) = \frac{1}{n}\sum_{i=1}^{n} K_H(s - s_i), \tag{3.1}$$

where, in the most general case, $K_H(s) = |H|^{-1/2}K(H^{-1/2}s)$, with H being a $d \times d$ symmetric and positive definite *bandwidth* matrix, whose meaning will be clarified later, and $K(\cdot)$ being a d-variate kernel function, bounded and with compact support, satisfying the following set of conditions [68]:

$$\begin{gathered}\int_{R^d} K(s)ds = 1, \quad \lim_{\|s\|\to\infty} \|s\|^d K(s) = 0, \\ \int_{R^d} sK(s)ds = 0, \quad \int_{R^d} ss^T K(s)ds = c_K I,\end{gathered} \tag{3.2}$$

where c_K is a constant and I is the identity matrix.

In [19], the author pointed out that a family of kernel functions satisfying the conditions 3.2 and showing the "sufficient" property of radial simmetry can be obtained in the following way:

$$K(s) = c_{k,d}k(\|s\|^2), \tag{3.3}$$

with $c_{k,d}$ normalizing constant, that is to say defining a univariate *kernel profile* $k(x)$ for $x \geq 0$ and rotating it in the space R^d.

It is further observed in [68] that, in order to limit complexity in the density estimation procedure, a common practical choice is to set the bandwidth matrix

H as proportional to the identity matrix, that is $H = h^2 I$, so that only one parameter should be provided in advance. Under this assumption, the formula of the estimator given in 3.1 becomes

$$\hat{p}_{h,K}(s) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{s - s_i}{h}\right), \tag{3.4}$$

and therefore, if we further assume the use of a radially symmetric kernel built as in 3.3, the following expression is obtained:

$$\hat{p}_{h,K}(s) = \frac{c_{k,d}}{nh^d} \sum_{i=1}^{n} k\left(\left\|\frac{s - s_i}{h}\right\|^2\right). \tag{3.5}$$

Applying the gradient operator to both sides of (3.5) yields to the form of the *density gradient estimator*. Using $g(x) = -k'(x)$, we obtain

$$\begin{aligned}\hat{\nabla} p_{h,K}(s) &= \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (s_i - s) g\left(\left\|\frac{s - s_i}{h}\right\|^2\right) \\ &= \frac{2c_{k,d}}{nh^{d+2}} \left[\sum_{i=1}^{n} g\left(\left\|\frac{s - s_i}{h}\right\|^2\right)\right] \left[\frac{\sum_{i=1}^{n} s_i g\left(\left\|\frac{s-s_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{s-s_i}{h}\right\|^2\right)} - s\right].\end{aligned} \tag{3.6}$$

Observe that the density estimate $\hat{p}(s)$ evaluated using the function $G(s) = c_{g,d} g(\|s\|^2)$ as a kernel (also called the *shadow* of kernel $K(s)$) is given by

$$\hat{p}_{h,G}(s) = \frac{c_{g,d}}{nh^d} \sum_{i=1}^{n} g\left(\left\|\frac{s - s_i}{h}\right\|^2\right), \tag{3.7}$$

therefore it is possible to rewrite Eq. 3.6 as

$$\hat{\nabla} p_{h,K}(s) = \frac{2c_{k,d}}{h^2 c_{g,d}} \hat{p}_{h,G}(s) \mathbf{m}_{h,G}(s), \tag{3.8}$$

with the term

$$\mathbf{m}_{h,G}(s) = \frac{\sum_{i=1}^{n} s_i g\left(\left\|\frac{s-s_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{s-s_i}{h}\right\|^2\right)} - s, \tag{3.9}$$

being called the *mean shift vector*.

3.1.2 Mean Shift Procedure for Mode Detection

Observing Eq. 3.9, it is possible to give the following intuitive interpretation: for each *kernel center* s, the mean shift vector points to the *local weighted mean*, whose weights are computed using the kernel G; therefore, starting from any center s it is possible to find the direction to the area where most of the data points are (locally) concentrated, that is, the direction of the maximum increase in the density. The same conclusions are drawn if we observe that (from Eq. 3.8)

$$\mathbf{m}_{h,G}(s) = \frac{1}{2} h^2 \frac{c_{g,d}}{c_{k,d}} \frac{\hat{\nabla} p_{h,K}(s)}{\hat{p}_{h,G}(s)}, \tag{3.10}$$

since the latter equation shows that the mean shift is proportional to the density gradient estimation computed using kernel $K(\cdot)$, normalized by the density estimation computed using the shadow of $K(\cdot)$. Such a normalization induces an interesting property, since the mean shift vector will be smaller for points close to a local maximum and larger for points in non-dense areas.

Based on these properties, an *iterative mode-seeking procedure* is introduced, aimed at tracking a path from a starting center kernel "up" to a mode of the probability density function; once a starting kernel center s is assigned, the procedure consists of two iterative steps:

1. compute the mean shift vector $\mathbf{m}_{h,g}(s)$,
2. update the kernel center $s = s + \mathbf{m}_{h,g}(s)$.

Since the normalization underlined in Eq. 3.10 implies an adaptive step size selection, the described procedure is in fact an adaptive gradient ascent method.

In [19] a proof of convergence of such procedure is given under some mild conditions for the kernel profile, assuring that the procedure will lead to a stationary point in the density function following a monotonically increasing sequence of density values. Notice that such proof does not eliminate the possibility that some non-maximum stationary point could be reached (e.g. *plateaus*).

Starting from this basic iterative procedure, a complete algorithm for mode detection is provided by simply running it many times, with different inizializations, in order to cover most of the feature space. To avoid that some non-maxima stationary points are detected, each time the procedure converges to a new point, this one is properly perturbed using a small norm vector and the

basic procedure is run again using the biased point as starting kernel center; if it converges to the same point, then it is for sure a local maxima.

A remarkable property of this algorithm is that it provides a data clustering as a by-product, since each data point converges only to one mode. This allows to subdivide the original sample set in different subsets of points associated with different modes; such subsets are usually called *basins of attraction* of the corresponding modes.

In the next section, a fast implementation of the mean shift clustering algorithm is proposed and its different issues are discussed in details.

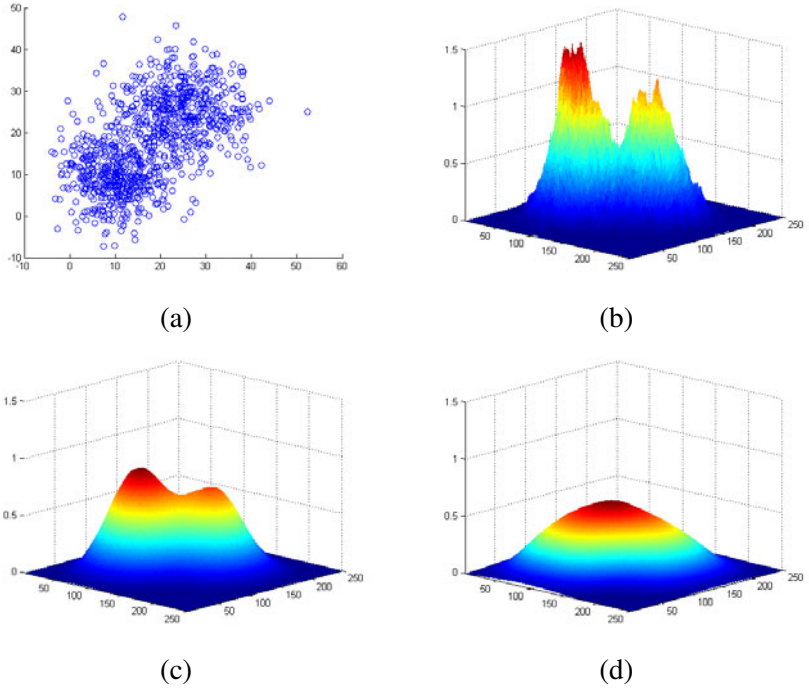
3.2 The Fast Mean Shift Clustering Algorithm

As already remarked above, the detection of modes through the Mean-Shift procedure determines an implicit clustering strategy over the feature space, since all the points of a basin of attraction form a well defined cluster.

However, this would require running the Mean-Shift procedure for each point of the feature space, so as to identify the basin of attraction of all modes as clusters. Of course, this is unfeasible in practice, since for sample sets larger than several hundreds of data points computational time becomes extremely large for most of the possible applications. Hence, an efficient implementation is usually required, especially for data-intensive cases.

Another critical implementation issue is the choice of the kernel size, or *bandwidth* parameter, which plays a central role for density estimation since it determines the smoothness of the pdf and, consequently, the number of modes that the algorithm singles out. Using a too large bandwidth leads to underestimating the number of modes, and the opposite for too small a value.

Let us further clarify this fundamental point observing that the *Parzen Window* basic equation (see Eq. 3.4) represents a direct way to build a *smoothed histogram* of the image: infact, rather than grouping points of the feature space together in bins, the kernel density estimator can be thought to place small "bumps" at each point, whose shape is determined by the kernel function $K(\cdot)$, and sum all of them together. The effect of this smoothing procedure can be observed in Fig. 3.1 for the bimodal sample set in (a): in case a too small kernel size is adopted, the underlying estimate of the density function will suffer from overfitting, finally leading to the detection of a too large number of modes. In (c), a reasonable value of the kernel size has been chosen, such that the density estimate presents the two meaningful modes one expects, while in (d) a too large value of the kernel size clearly caused underfitting.



(a) (b)

(c) (d)

Figure 3.1: Role of the *bandwidth* parameter: a random bimodal sample set (a) and three different kernel density estimates using a too small (b), reasonable (c) and too large (d) kernel size.

In general, the correct choice of h is all but a simple task, in many cases being really critical, meaning that small "changes" of the value can significantly alter the outcome of the mode detection procedure. This is the case when dealing with image segmentation, as observed since our preliminary work presented in [69], making the proper choice of h a critical issue for the deployment of a robust clustering technique.

We propose here an implementation of Mean-Shift clustering which addresses the two problems outlined above. In particular, the new algorithm is based on:

- a data-dependent adaptive kernel size h that overcomes the instability of the typical fixed strategy;
- a fast clustering technique that enables its use for real-world applications.

3.2.1 The Adaptive Kernel Size Strategy

Selection of Kernel Shape

First of all, let us discuss about the shape of the kernel function $K(\cdot)$ to use in the implementation of the Mean Shift procedure. As far as the conditions for convergence about the kernel shape are not so strict, it is possible to choose among a wide range of possibilities. As stated above, a class of particularly interesting kernels, above all for their mathematical tractability, are the radially symmetric kernels. In literature, mainly two of these kernels have been used: the *Epanechnikov* kernel [70], having good properties both in minimizing the overall error between a density and its estimate [71] and in terms of computational manageability, and the *Normal* kernel, that is proved to provide a strict gradient ascent when applied to the mean shift procedure and gives in general better results, even if the number of iteration needed for convergence is higher then in the Epanechnikov case [19].

However, selection of kernel shape is by no means critical. A common choice widespread in literature is the use of a Normal kernel, generated using the method of Eq. 3.3 starting from the following kernel profile function

$$k_N(x) = \exp\left(-\frac{x}{2}\right) \quad x \geq 0, \tag{3.11}$$

that infact leads to the multivariate kernel

$$K_N(s) = (2\pi)^{-d/2} \exp\left(-\frac{\|s\|^2}{2}\right) \tag{3.12}$$

that evidently has a Normal shape.

In practice, there is no need to compute the values of the kernel function or the profile function, since the quantity actually computed is the mean shift vector by means of Eq. 3.9. The only function that we need to evaluate is g, whose expression is

$$g_N(x) = -k'_N(x) = \frac{1}{2} \exp\left(-\frac{x}{2}\right). \tag{3.13}$$

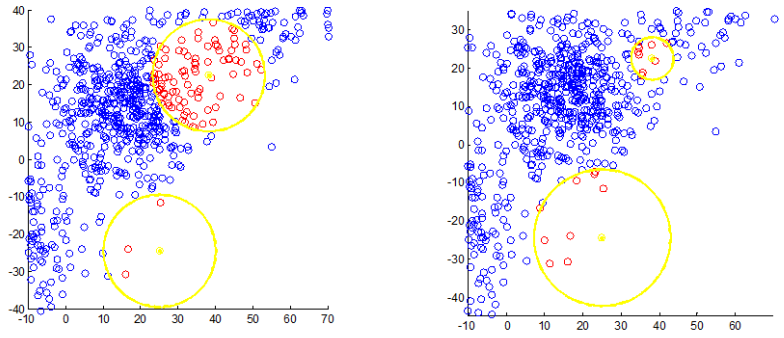
k-nn based Adaptive Kernel Size Selection

The original Mean-Shift procedure proposed by Comaniciu [19] uses a fixed bandwidth parameter h, but this is clearly inappropriate when the density of points in the feature space varies wildly. In such cases, in fact, no value can be well suited for both high- and low-density areas.

To face this problem, we adapt the bandwidth parameter locally in the feature space by taking into account only the first k-Nearest Neighbors in the computation of the Mean Shift vector. This amounts to truncating the kernel at some distance from the center but, if k is not too small, this truncation will take place when the kernel has already a negligible value, independent of the local density. The bandwidth, instead, will clearly depend on the local density, being larger in low-density areas and smaller in high density ones.

The difference between the fixed and variable bandwidth approaches can be better appreciated in Fig. 3.2, where an example is shown about how the bandwidth parameter h can be selected (a) by means of a fixed choice and (b) taking into account only a fixed number of nearest neighbours from the kernel center. In the first case, starting from a sparse area of the feature space, the reduced number of neighbours "captured" by the kernel can easily compromise reliability in the computation of the mean shift vector, while in dense areas the risk of an underfitting of the density function increases. Such effect is not present with the proposed alternative strategy, as can be observed in Fig. 3.2(b), where the kernel size adapts itself to the local density of points in the feature space.

In more detail, given a suitable value of k, at each step of the procedure the set $NN(s)$ of k points closest to s is singled out, and the kernel size is calculated as:



(a) (b)

Figure 3.2: Adaptive bandwidth selection: (a) a fixed kernel size strategy, (b) a variable kernel size strategy obtained using a fixed number of nearest neighbours.

$$h(s) = \sqrt{\frac{1}{k} \sum_{i \in NN(s)} \|s - s_i\|^2}, \tag{3.14}$$

This value is then used in (3.9) for the computation of the mean-shift vector where the summation is again restricted to the points in $NN(s)$.

It could also be observed that this solution moves the problem from the estimation of parameter h, to that of parameter k, but it is well-known [71] that k-*NN* estimation is quite robust w.r.t. its parameter, and works quite well also in high dimensional spaces, which are instead quite challenging for the Mean-Shift. In next subsection, we propose a data dependent procedure for obtaining a stable estimate of the k parameter.

3.2.2 Fast Mean Shift based Clustering

Our speed-up strategy is based on the obvious consideration that all points that lie on the trajectory that goes from the starting point to the corresponding mode belong necessarily to the same basin of attraction. Therefore, they could all be attributed, without error, to the same cluster.

Although it is extremely difficult that any sample point will coincide *exactly* with a point of this path, one can reasonably assume that sample points

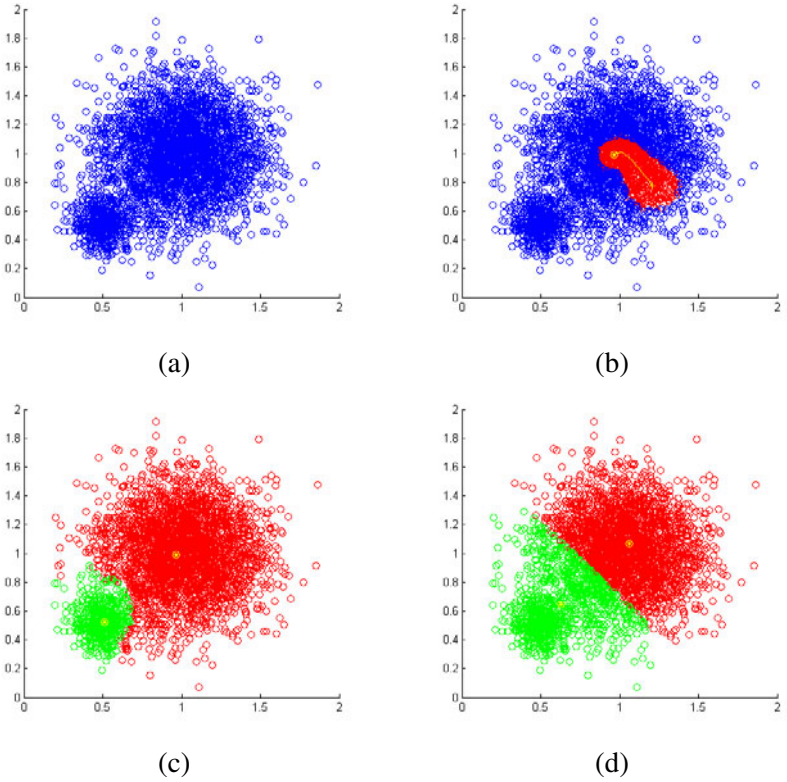


Figure 3.3: (a) bi-modal sample set, (b) Mean Shift trajectory with the corresponding "voting" points, (c) final clustering, (d) GLA-based clustering for comparison.

that are *close* to the trajectory belong very likely to the same basin. By clustering all such points at once we drastically reduce the complexity, but also risk to cause some errors, especially for data points that are close to the watershed between two basins of attraction. Hence, in order to preserve the accuracy of clustering, we do not assign sample points on the fly, but rather implement a voting mechanism and decide only a posteriori, with a majority rule, when all sample points have been touched by at least one trajectory.

The modified procedure can be summarized as follows:

Algorithm 3.2.1 (Fast Mean Shift Clustering (FMSC))

1. Initialization*: set all sample points as* non visited*.*
2. Mean-Shift*: run the procedure starting from a randomly selected non visited point: at each step along the trajectory, mark as* visited *all points* s_i *such that* $\|s - s_i\| < h(s)$*, and for each of them add a vote for the "final" mode.*
3. Mode validation*: once convergence is reached, compute the distance* $d_{\min}$ *between the new tentative mode and the closest mode already detected:*
   - *if* $d_{\min} < h/2$ *reject the new mode, and mark the closest mode as final;*
   - *otherwise accept the new mode, and mark it as final.*
4. Test*: if there are still non visited points, go to step 2.*
5. Clustering*: assign each visited point to the mode (and cluster) with the most votes.*

An example of clustering provided by the described procedure is presented in Fig. 3.3: the bivariate sample set of part (a), obtained as a mixture of two normally distributed data sets, is given as input to the clustering algorithm. In part (b) the effect of a single modified Mean-Shift procedure is represented, where all the points in red are "giving a vote" to the final mode. Part (c) shows the final clustering, which appears to follow quite faithfully the underlying distribution and is certainly much better than the clustering based on the Generalized Lloyd Algorithm shown in part (d) where, in addition, the correct number of clusters had to be provided as a further input.

3.3 The Unsupervised TS-MRF/MS Algorithm

Turning to the unsupervised TS-MRF based segmentation techique, the general segmentation strategy discussed in Sec. 2.3 must be translated into a real-world functioning algorithm, where a number of implementation choices, sometimes driven by complexity concerns, might have a critical impact on the overall performance.

One such choice, made in [18] to simplify the local optimization task, is to consider only binary tree structures, reducing the segmentation process to a sequence of nested binary splits controlled by a suitable stopping criterion. Such a constraint, however, might cause the detection of false contours as can happen when three or more balanced classes are present in the same region. In [69] we removed this constraint and resorted to the Mean-Shift procedure to detect the number of pdf modes in a class, and hence the number of children at a given node.

Another critical choice is the use of the Generalized Lloyd Algorithm to carry out the initial segmentation needed to perform the MRF optimization at each node. In fact, image pixels are often described by a complex and generally unbalanced probability distribution in the spectral domain, in which case the GLA can easily provide inaccurate results, as in the example of Fig. 3.3(d).

Here we propose a modification of the original unsupervised TS-MRF algorithm, aimed at increasing both the flexibility of the process, by removing the aforementioned binary constraint, and the quality of initial segmentation at each step of the recursive procedure, making use of a finer pixel-wise clustering technique.

3.3.1 Proposed Modification to the Unsupervised TS-MRF

The fundamental modification to the original algorithm consists in the replacement of the GLA based segmentation originarily proposed for the inizialization of MRFs locally to each node of the tree (that is, at each step of the recursive segmentation process) with the more accurate segmentation technique relying on the variable-bandwidth Mean-Shift based clustering described in the previous section. With respect to the flowchart depicted in Fig. 2.7, we basically operate on the block containing the *SplitTree* function, moving it from the logical scheme of Fig. 3.4(a) to the one in (b).

A first immediate consequence of the proposed solution concerns the elimination of the binary constraint that characterized the original technique. Infact, the proposed Mean Shift based procedure is able to automatically determine

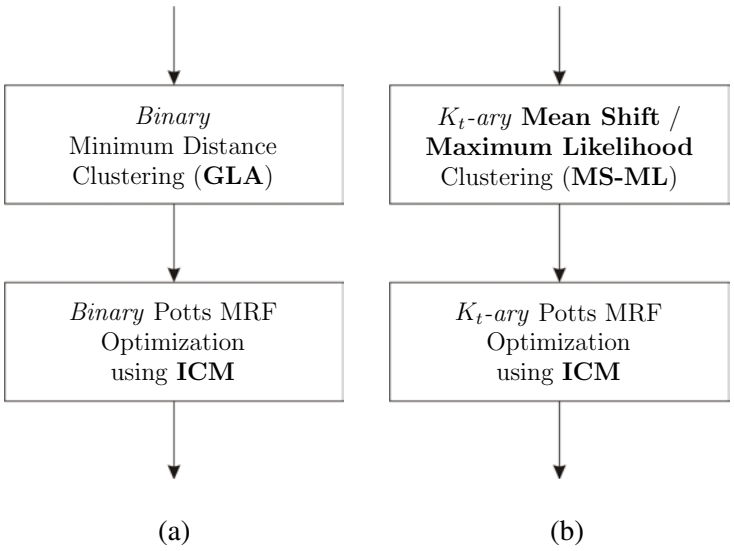


Figure 3.4: Modification to the original Unsupervised TS-MRF algorithm: high-level flowchart of the old (a) and new (b) *Split Tree* function (see Fig. 2.7).

the number of cluster in the subset under analysis, meaning that, during the unsupervised segmentation process, the growth of the tree at each node is no longer guaranteed to be binary. Therefore, at the end of the process a generic tree structure can be eventually retrieved.

Extension to Generic Tree Structures

Since the original formulation of the TS-MRF modeling framework relies on binary tree structures, further discussion on the properties introduced in the pevious chapter is now made necessary to validate the theoretical robustness of the proposed method. However, this turns out to be not a difficult task, as we will briefly outline in the following.

To generalize TS-MRF model properties to generic tree structures, let us consider the non-binary tree of Fig. 3.5, where from each node t a non-constant number of children K_t originates. Following the same path of Sec. 2.2.2, ob-

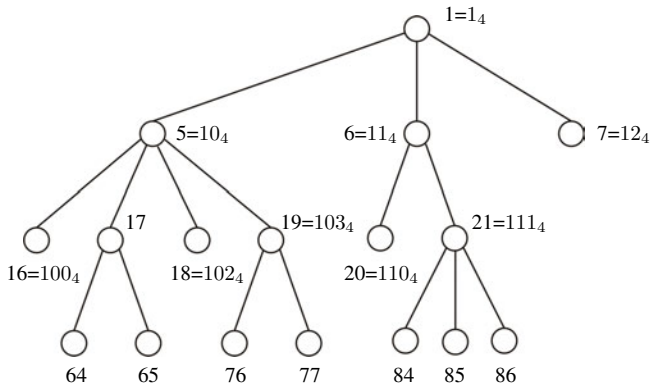


Figure 3.5: Tree indexing for generic tree structures.

serve that:

- concerning *tree indexing*, we can simply extend the introduced formalism by assigning to each node a base-$\tilde{K}$ number, in place of a binary string, where $\tilde{K} = \max_t K_t$, along with the corresponding base-10 integer. The children of a generic inner node t are indicated as $c_1(t), \ldots, c_{K_t}(t)$ and, given the father label, their base-$\tilde{K}$ identifier are obtained from it appending different digits from 0 to K_t. With this new convention, redefinitions of the indexing functions $\nu(t)$ and $\Psi(a, b)$ are straightforward.
- The number of "structural" constraints of the type in Eq. 2.13 is now in general $\leq \frac{1}{2}K(K-3)+1$.

All the considerations leading to the joint probability of Eq. 2.14 remains exactly the same, as shown in the following "updated" example relative to the tree of Fig. 3.5:

$$
\begin{aligned}
& p(x_s = 84 | x_{\eta(s)}) = \\
= \; & \frac{1}{Z} \exp[-\beta_1(N_5 + N_7) - \beta_6 N_{20} - \beta_{21}(N_{85} + N_{86})] = \\
= \; & \frac{1}{Z} \exp[-\beta_1(N_{16} + N_{64} + N_{65} + N_{18} + N_{76} + N_{77} + N_7) + \\
- \; & \beta_6 N_{20} - \beta_{21}(N_{85} + N_{86})].
\end{aligned}
$$

Similarly, concerning the recursive optimization procedure, now we simply have to consider that each random field X^t is now K_t-ary, and its realization x^t is such that $x_s^t \in \{c_1(t), \ldots, c_{K_t}(t)\}$. The generalization of the segmentation constraint of Eq. 2.15 is also considered, thus for each internal node of the tree $t \in \bar{T}$:

$$\begin{cases} \mathcal{S}^{c_1(t)} &= \{s \in \mathcal{S}^t : \; x_s^t = c_1(t)\} \\ \quad \vdots & \\ \mathcal{S}^{c_{K_t}(t)} &= \{s \in \mathcal{S}^t : \; x_s^t = c_{K_t}(t)\} \end{cases}$$

Under these conditions, independence among disjoint subtrees still holds, as we can see from the following example, always referring to the tree of Fig. 3.5:

$$\begin{aligned} & p(x^{17}, x^{19}, x^5 | x^6, x^{21}, x^1) = \\ = \; & \frac{p(x)}{p(x^6, x^{21}, x^1)} \\ = \; & \frac{\frac{1}{Z}\exp[-\sum_{t\in\bar{T}} \beta_t \mathcal{N}_t]}{\sum_{x^{17},x^{19},x^5} \frac{1}{Z}\exp[-\sum_{t\in\bar{T}} \beta_t \mathcal{N}_t]} \\ = \; & \frac{\frac{1}{Z}\exp[-\beta_{17}\mathcal{N}_{17} - \beta_{19}\mathcal{N}_{19} - \beta_5\mathcal{N}_5]}{\sum_{x^{17},x^{19},x^5} \frac{1}{Z}\exp[-\beta_{17}\mathcal{N}_{17} - \beta_{19}\mathcal{N}_{19} - \beta_5\mathcal{N}_5]} \\ = \; & \frac{1}{Z(x^1)}\exp[-\beta_{17}\mathcal{N}_{17} - \beta_{19}\mathcal{N}_{19} - \beta_5\mathcal{N}_5] \\ = \; & p(x^{17}, x^{19}, x^5 | x^1). \end{aligned} \tag{3.15}$$

The Mean Shift/Maximum Likelihood (MS-ML) Classifier

Even though our fast implementation helps limiting the processing burden, plain Mean-Shift clustering would have an exceedingly high computational complexity for the very large images we usually deal with, and hence we will eventually resort to a hybrid *Mean-Shift/Maximum Likelihood* (MS-ML) classifier. In more details, for each region $\mathcal{S}^t$ the following unsupervised segmentation procedure is run:

Algorithm 3.3.1 (MS-ML classifier)

1. *a sufficiently large random subset of pixels $y_s^t : s \in \mathcal{S}^t$, say $\tilde{y}^t$, is extracted (from 1% of the region area $|\mathcal{S}^t|$ to the entire region, depending on its size);*

2. *the Mean-Shift clustering described in Sec. 3.2 is then applied to this sample set, obtaining its clustering in K_t subsets;*

3. *each subset is used to characterize a corresponding class, using the mean vectors μ_i and covariance matrices Σ_i, $i = 1, \ldots, K_t$;*

4. *the initial segmentation map of region $\mathcal{S}^t$, that is, the initial field x_0^t, is then obtained by means of a* Maximum Likelihood *(ML) classification using the previously computed statistics (each class is modeled using a multivariate Gaussian):*

$$x_0^t = \arg\max_{x^t} p(y^t|x^t)$$

Concerning the point 2 of this procedure, in Sec. 3.2.1, we did not address the problem of selecting a suitable value of k for the k-*NN* based bandwidth estimation of (3.14). A typical choice is to set k to a fraction, *e.g.* 10%, of the sample set cardinality, which, given the robustness of k-*NN*, provides usually good results. For some nodes, however, this simple choice turned out to be unsatisfactory, causing a proliferation of modes in the Mean-Shift clustering and a certain instability in the segmentation. This is not surprising, after all, given that the same algorithms are used at all nodes, from the root, corresponding to the whole image, to terminal leaves corresponding sometimes to much smaller and much more fragmented regions.

Therefore, we use a simple heuristic procedure that adapts the value of k to minimize such unlikely behaviors. Our underlying assumption is that, most of the times, the data structure can be well described through one or more binary splits, hence the procedure is based on quantifying the "stability" of the Mean Shift procedure in detecting a number of modes equal to 2: starting from an initial guess of k, namely $k_0 = \text{round}(\alpha_0|\tilde{y}^t|)$, with $\tilde{y}^t$ being the current sample set under analysis and $0 < \alpha_0 < 1$ being the desired fraction of $|\tilde{y}^t|$, *e.g.* 0.1, the basic Mean Shift procedure described in Sec. 3.1.2 is run multiple times (at most C_1 times), each with a different initialization, while the number of deteted modes is kept under observation. A good value of k is the one that allow the stable detection of 2 modes (within a certain number C_2 of subsequent iterations), and from the initial value k_0 it can be modified as follows:

Algorithm 3.3.2 (Automatic *k* refinement)

1. *Set the detected number of modes $D = 0$, the current $k = k_0$, and the total number of iterations $it = 0$;*

2. *Set $seq = 0$, being the current count of subsequent iterations where the number of detected modes remains unchanged;*

3. *repeat the following steps while $it < C_1$ and $seq < C_2$:*

   - *if the current k is outside of the range $[\alpha_1 k_0, \alpha_2 k_0]$, with $\alpha_1 < \alpha_0 < \alpha_2$, accept it and exit;*
   - *select a random "unused" starting center from the sample set and mark it as "used";*
   - *run the basic Mean Shift procedure: if a new mode is detected, update the total number of modes $D = D + 1$ and set $seq = 0$, else set $seq = seq + 1$;*
   - *if $D > 2$, update the current value of k = round$(k + \Delta k)$ and return to step 2;*
   - *if $D < 2$ and $seq > \frac{N_2}{2}$, update the current value of k = round$(k - \Delta k)$ and return to step 2;*
   - *Set $it = it + 1$;*

4. *accept the current value of k and exit.*

This procedure also provides a solid criterion to decide whether to split a node or not, since the stable detection of a single mode qualifies the corresponding region as elementary.

Using a more reliable technique to carry out the initial clustering does certainly improve the subsequent MRF optimization, but there is a more subtle and important consequence in the context of hierarchical segmentation. In fact, the MS-ML clustering provides a quite reliable segmentation in the spectral domain, while the MRF model allows to take into account contextual information to regularize the final map. The points that change label during MRF optimization turn out to be "outliers" in the spectral domain for the final class ω, that is, their statistics will be far apart from those of points originally attributed to ω by the MS-ML technique. If class ω is segmented again, such outliers can give origin to one or more separate clusters, leading to critical over-segmentation errors. We are now in the position to solve this unwanted phenomenon, by simply erasing such points from the new sample set. Notice that this was not possible with a GLA initialization, since the initial segmentations were in general so far from the final segmentation (compare again Fig. 3.3) that such erasure would amount to eliminate large valid chunks of data.

	DB.	B.	LB.	C.	G.	O.	R.	Br.	*u.a.*
D.Blue	**49083**	2749	0	0	0	0	0	0	*94.7%*
Blue	2467	**44573**	0	0	0	0	0	0	*94.7%*
L.Blue	0	0	**20922**	0	6	0	0	0	*99.9%*
Cyan	0	0	8	**29944**	20361	0	0	0	*59.5%*
Green	0	0	14866	11383	**5472**	0	0	0	*17.2%*
Orange	0	0	6	0	0	**13129**	1436	6574	*62%*
Red	0	0	0	0	0	2	**14987**	6863	*68.6%*
Brown	0	0	0	0	0	5	0	**17308**	*99.9%*
p.a.	*95.2%*	*94.2%*	*58.4%*	*72.5%*	*21.2%*	*99.9%*	*91.2%*	*56.3%*	***74.5%***

Table 3.1: Confusion matrix for the segmentation of Fig. 3.6(c). In bold, correct assignments. (*p.a.* is the producer's accuracy, *u.a.* the user's accuracy, as defined in Sec. 3.4.1).

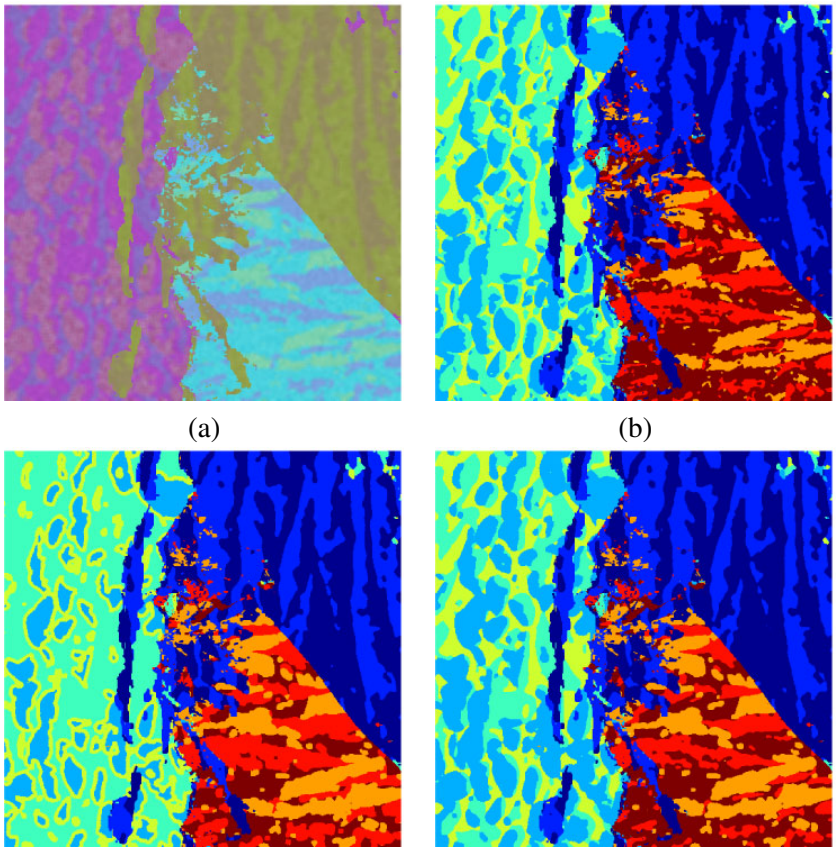
3.3.2 Preliminary Experimental Results

In order to validate the ideas that have led to the proposed technique and to provide a first ispection in its potential, several tests of the algorithm have been performed on synthetic data.

The three-band synthetic image, shown in Fig. 3.6(a), has been obtained by projecting the ground truth of Fig. 3.6(b) on the data space, adding white noise, and finally performing a light spatial filtering. The reference algorithm generates the tree structure shown in Fig. 3.7(a) and the segmentation map of Fig. 3.6(c), while the new algorithm generates the tree structure of Fig. 3.7(b) and the segmentation map of Fig. 3.6(d).

For this experiment, we set the main parameters of the automatic procedure for the selection of k as $\alpha_0 = 0.1, \Delta = 0.05, C_1 = 100, C_2 = 5$.

It is clear that the old technique has a hard time fitting the intrinsic structure of the data, that has been willingly chosen as non-binary. In some cases, infact, a ternary split is needed, for example in the root node: it can be noticed infact that, at a coarser scale of observation, three spectrally coherent macroregions are present that are also almost "equally spaced" in the spectral domain. Here, the algorithm must simulate it by means of a sequence of two binary splits. Sometimes, this has no detrimental effect, like in the root itself, where the dark-blue, light-blue and orange macroregions are correctly singled out, but in at least one instance, the split of the light-blue regions, this leads to a grossly



(a) (b)

(c) (d)

Figure 3.6: Testing the new TS-MRF/MS algorithm on synthetic data: test image (a), ground truth (b), 8-class segmentation with the classical unsupervised TS-MRF (c) and the proposed method (d).

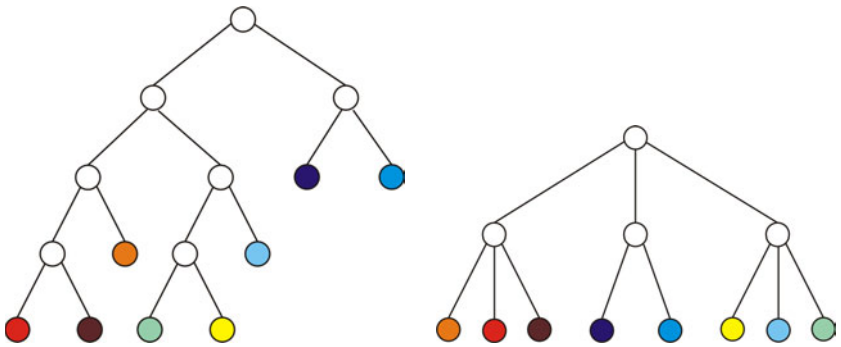


Figure 3.7: Testing the new TS-MRF/MS algorithm on synthetic data: tree structures for the experiment of Fig. 3.6 retrieved respectively using the old unsupervised TS-MRF (a) and the new TSMRS/MS algorithm (b).

	DB.	B.	LB.	C.	G.	O.	R.	Br.	*u.a.*
D.Blue	**50429**	4590	0	0	0	0	0	0	*94.7%*
Blue	1121	**42732**	0	0	0	0	0	0	*94.7%*
L.Blue	0	0	**35724**	9224	2634	0	0	0	*99.9%*
Cyan	0	0	16	**27990**	275	0	0	0	*59.5%*
Green	0	0	55	4113	**22930**	0	0	0	*17.2%*
Orange	0	0	7	0	0	**13108**	736	5993	*62%*
Red	0	0	0	0	0	16	**14266**	903	*68.6%*
Brown	0	0	0	0	0	12	1421	**23849**	*99.9%*
p.a.	*97.8%*	*90.3%*	*99.7%*	*77.7%*	*88.7%*	*99.8%*	*86.8%*	*77.5%*	***88.1%***

Table 3.2: Confusion matrix for the segmentation of Fig. 3.6(d). In bold, correct assignments.

inaccurate segmentation, as also testified by the confusion matrix[1] reported in Table 3.1. From another point of view, this inaccuracy can be seen as the detection of a false intermediate contour: a further binary split of the cyan region using the old technique succeeds in revealing the correct missing contour, but the overall result will be an obvious oversplitting of the macroregion.

On the contrary, the proposed TS-MRF/MS provides the correct (or *a* correct) tree structure for the test image, with a first ternary split at the root node that singles out the correct macroregions, each of which is then split in two or three regions, following their actual composition. As a consequence, the segmentation map is globally more accurate, with overall accuracy jumping from 74.5% to 88.1%, but for some random sparse errors, as obvious from the analysis of the corresponding confusion matrix of Table 3.2. Major improvements have been obtained on the light-blue macroregion, due to the direct ternary split, and on the orange one. For the latter case, we observed in particular that the new split initialization method leads to a more accurate contour detection, thus pointing out again the limits of the old GLA-based algorithm.

For this preliminary experiment, an interesting result concerns also the total number of classes detected by the two algorithms, that is, the cluster validation. Using the old version of the algorithm, the segmentation process does not stop until the maximum (overestimated) number of classes is reached, while with TS-MRF/MS, it stops automatically when 9 classes are detected, with only one elementary region oversplit, thus resulting, for the case, in a drastic reduction of oversegmentation phenomena.

3.4 Application to Remote Sensing

3.4.1 Classification of Multispectral SPOT Data

Spot Image of Lannion Bay (France)

The unsupervised TS-MRF/MS algorithm has been applied to SPOT satellite images. The scene (Fig. 3.8 - 3.10) is composed of three 1480×1024 images with different wavelengths in the visible spectrum and represents the Bay of Lannion in France in August 1997. The goal of this study was to determine the land cover of this area. So as to reach this aim, the geographers of the Costel laboratory (University of Rennes 2) built a list of eight classification

[1] Details on the confusion matrices and on the general accuracy assessment framework will be discussed in Sec. 3.4.1.

categories: *sea and water*, *sand and bare soil*, *urban areas*, *forests and heath*, *temporary meadows*, *permanent meadows*, *vegetables*, *corn*.

Thanks to both tests on the land and photointerpretation, they were also able to provide samples of these eight categories on the multispectral SPOT image of the scene. The resulting *ground truth* (Fig. 3.11), has been here used to assess the accuracy of the classifications.

Accuracy Assessment Method

By the use of the ground truth, the accuracy of the old and new TS-MRF based classification methods is assessed based on its *confusion matrix*. Recall that the entry of ith row and jth column of this matrix is the number of sample pixels from jth class that have been classified as belonging to the ith class. Since the tested methods are unsupervised, associations of retrieved labels with actual ground truth classes is made by selecting the configuration that gives the best *overall accuracy*.

Various indicators are derived from this matrix. First, two error assessments can be computed for each class: the *user's accuracy* of class i is defined as a_{ii}/a_{i+}, where a_{i+} is the ith row marginal (sum of row entries); conversely, the *producer's accuracy* of this class is defined as a_{ii}/a_{+i}, where a_{+i} is the ith column marginal.

Beside these two class-based parameters, three global quality indicators are also computed. The overall accuracy of the method defined as $\tau = \sum_i a_{ii}/N$, is the percentage of sample pixels that are well classified. Another common indicator is the so-called Kappa parameter, defined as $\kappa = (N\sum_i a_{ii} - \sum_i a_{i+}a_{+i})/(N^2 - \sum_i a_{i+}a_{+i})$, which discounts successes obtained by chance and is therefore more conservative (it can be also negative). Finally, in order to give the same weight to all classes' contributions to the accuracy, irrespective of the number of samples in each one, the confusion matrix can be normalized with the iterative proportional fitting algorithm [72], so that all column and row marginals sum up to unity. The overall accuracy τ^{norm} computed on such a modified matrix is called normalized accuracy.

3.4.2 Experimental Results for Unsupervised Classification

For both the original TS-MRF algorithm and the new version proposed here we use the same settings for the MRF optimization part, and stop the tree growth manually at 8 classes in order to allow rigorous assessment through the available ground truth. This choice is justified by the fact that, unfortunately, for this

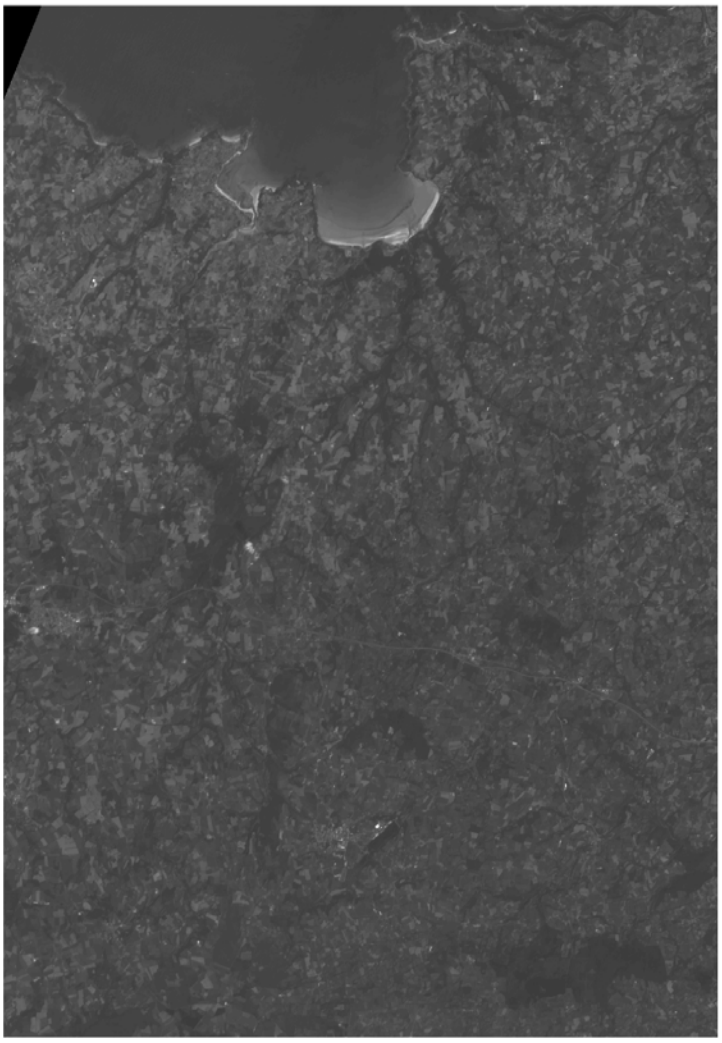


Figure 3.8: SPOT multispectral image of Lannion Bay: channel XS1 (©SPOTImage/CNES).

Figure 3.9: SPOT multispectral image of Lannion Bay: channel XS2 (©SPOTImage/CNES).

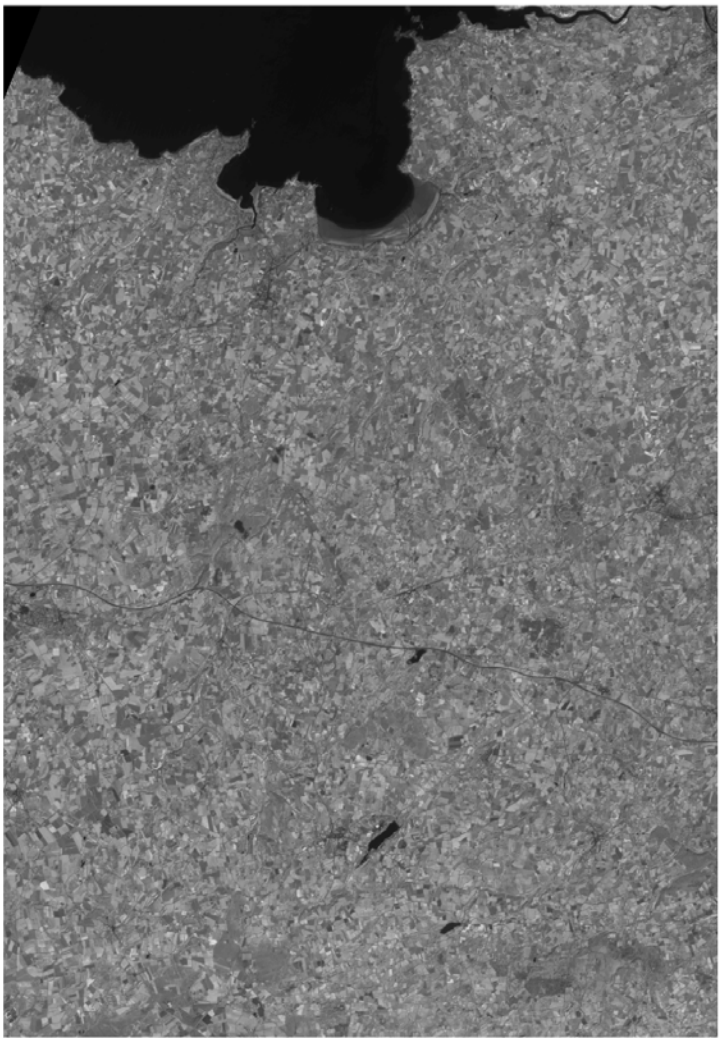


Figure 3.10: SPOT multispectral image of Lannion Bay: channel XS3 (©SPOTImage/CNES).

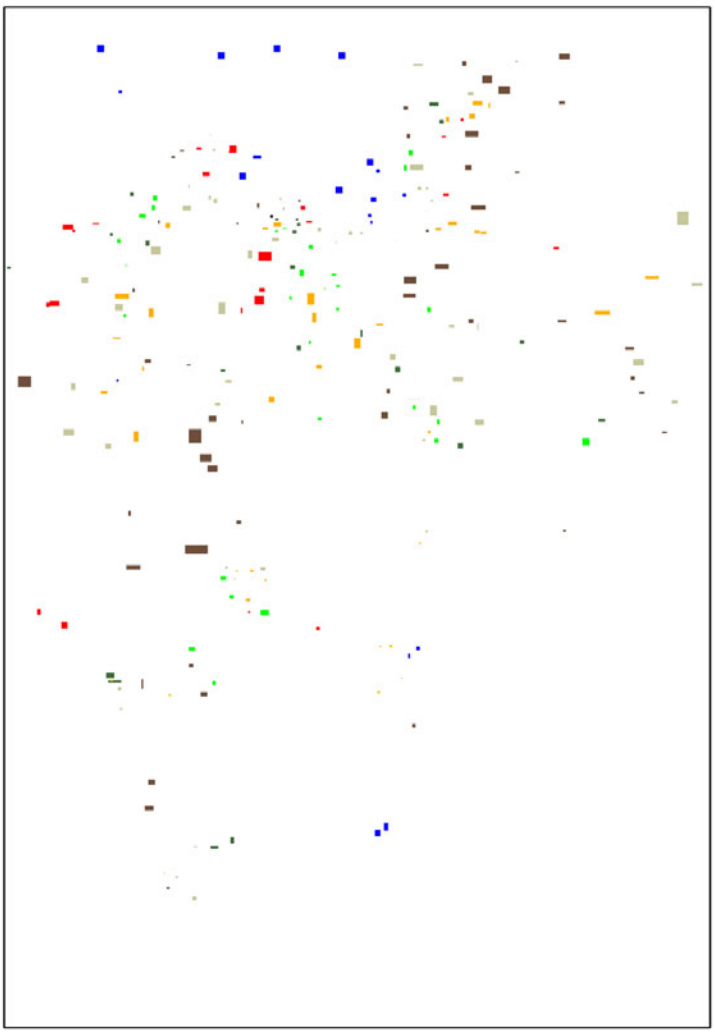


Figure 3.11: Ground-truth of the SPOT image of Lannion Bay: legend in Fig. 3.12. ©COSTEL.

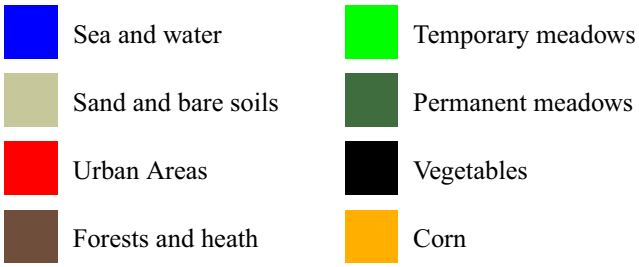


Figure 3.12: SPOT image: legend of land-cover classes.

kind of data both the tested algorithms are generally unsuccessful in resolving the cluster validation; however, it is worth reminding that a general solution to this problem is really far away to come, thus making this lack irrelevant if compared with the other qualifying points that characterize the techniques.

Anyway, as an alternative validation method, we could let the segmentation process evolve until the automatic stop and compare the obtained map with some "reliable" one, *e.g.* obtained by means of a supervised process as in [1], using some more general indicator like the *Local* or *Global Consistency Errors* [73] that can also compare maps with a different number of classes. For this application, we decide to use the rigorous assessment described earlier in this section, renouncing to test the cluster validation and resorting to a manual stopping criterion in order to fully highlight the remaining potentials of the proposed technique.

Turning back to the experiments, the mode detection procedure uses here $\alpha_0 = 0.1, \alpha_1 = 0.08, \alpha_2 = 0.12, \Delta = 0.05, C_1 = 100$ and $C_2 = 10$.

The improvements due to the use of the MS-ML are quite clear since the first stages of segmentation. In Fig. 3.13(a) we show a detail of the source image, along with two maps that, for both the original (b) and new version (c) of the algorithm, show the "sea" class (in white) as identified by the top-level clustering, before any MRF regularization. The errors introduced by the GLA are quite evident in Fig. 3.13(b), as well as the very high accuracy of the MS-ML classification of Fig. 3.13(c). Such a good initialization will likely improve, and certainly simplify the subsequent optimization process (making up for the increased complexity of the MS-ML clustering). Moreover, it will allow to single out easily the few label-switching points to eliminate in further

(a) (b) (c)

Figure 3.13: Detail of the XS3 channel (©SPOTImage/CNES) (a), initial *sea class* split using GLA (b), and MS-ML (c).

	W.	B.S.	U.	F.	T.M.	P.M.	V.	C.	*u.a.*
Water	**847**	0	0	0	0	0	0	0	*100%*
B. Soil	0	**3003**	937	3	0	19	12	3	*75.5%*
Urban	45	47	**82**	1828	41	6	0	8	*4%*
Forests	2	9	4	**1944**	14	6	0	23	*97.1%*
Temp. M.	0	46	252	11	**292**	152	1	5	*38.5%*
Perm. M.	0	5	6	358	284	**209**	3	41	*23.1%*
Veget.	226	50	10	11	0	0	**0**	0	*0%*
Corn	0	0	8	55	337	557	9	**1710**	*63.9%*
p.a.	*75.6%*	*95%*	*6.3%*	*46.2%*	*30.2%*	*22%*	*0%*	*95.5%*	**59.8%**

Table 3.3: *Confusion Matrix* corresponding to the segmentation map of Fig. 3.14.

	W.	B.S.	U.	F.	T.M.	P.M.	V.	C.	*u.a.*
Water	**1055**	0	0	0	0	0	0	0	*100%*
B. Soil	0	**2183**	55	0	0	0	0	0	*97.5%*
Urban	0	494	**530**	29	0	12	11	1	*49.2%*
Forests	65	13	0	**4023**	1	0	0	41	*97.1%*
T. M.	0	142	634	27	**321**	121	2	19	*25.3%*
P. M.	0	29	38	99	322	**273**	11	38	*33.7%*
Veget.	0	298	41	0	0	0	**1**	0	*3.4%*
Corn	0	1	1	32	324	543	0	**1691**	*65.2%*
p.a.	*94.2%*	*69.1%*	*40.8%*	*95.6%*	*33.2%*	*28.8%*	*4%*	*94.5%*	**74.4%**

Table 3.4: *Confusion Matrix* corresponding to the segmentation map of Fig. 3.16.

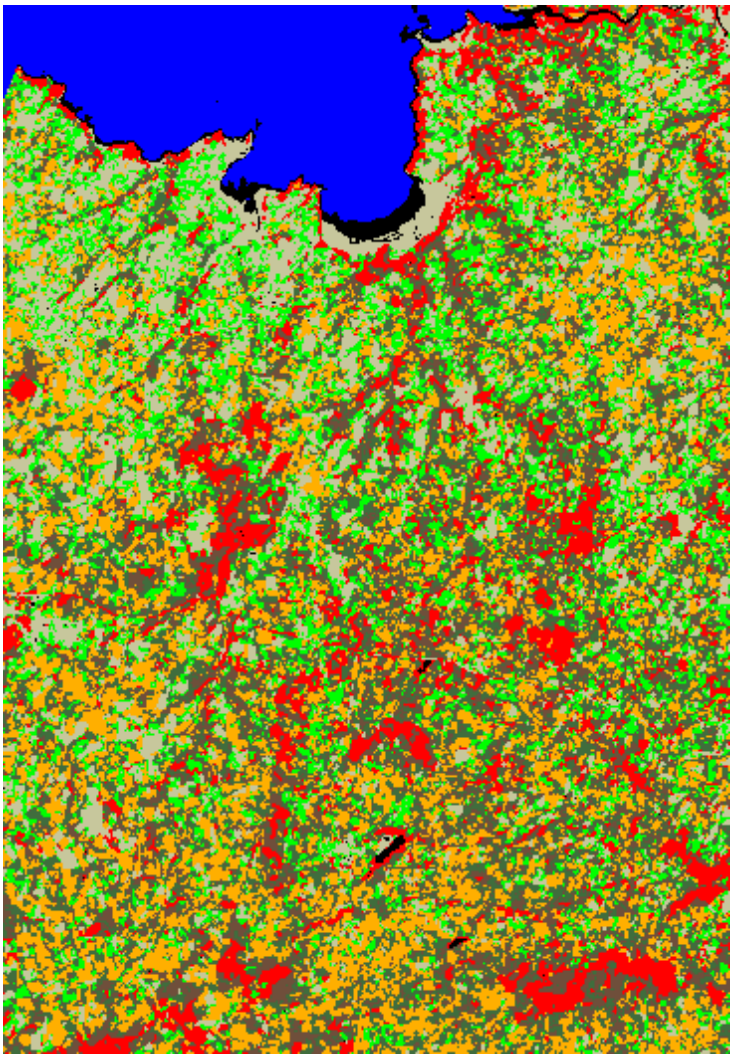


Figure 3.14: Unsupervised segmentation of the SPOT image obtained using the original TS-MRF algorithm.

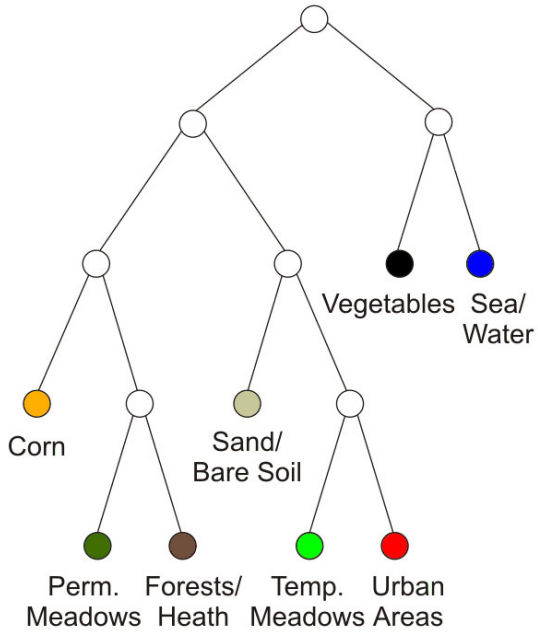


Figure 3.15: Tree structure retrieved for the map of Fig. 3.14.

	W.	B.S.	U.	F.	T.M.	P.M.	V.	C.	*u.a.*
Water	**847**	0	0	0	0	0	0	0	*100%*
B. Soil	0	**3003**	937	3	0	19	12	3	*75.5%*
Urban	0	46	**252**	11	292	152	1	5	*33.2%*
Forests	47	56	86	**3772**	55	12	0	31	*92.9%*
T. M.	0	5	6	358	**284**	209	3	41	*31.3%*
P. M.	0	0	6	19	129	**260**	3	526	*27.6%*
Veget.	226	50	10	11	0	0	**0**	0	*0%*
Corn	0	0	2	36	208	297	6	**1184**	*68.3%*
p.a.	*75.6%*	*95%*	*19.4%*	*89.6%*	*29.3%*	*27.4%*	*0%*	*66.1%*	**71%**

Table 3.5: *Confusion Matrix* corresponding to the segmentation maps of Fig. 3.18.

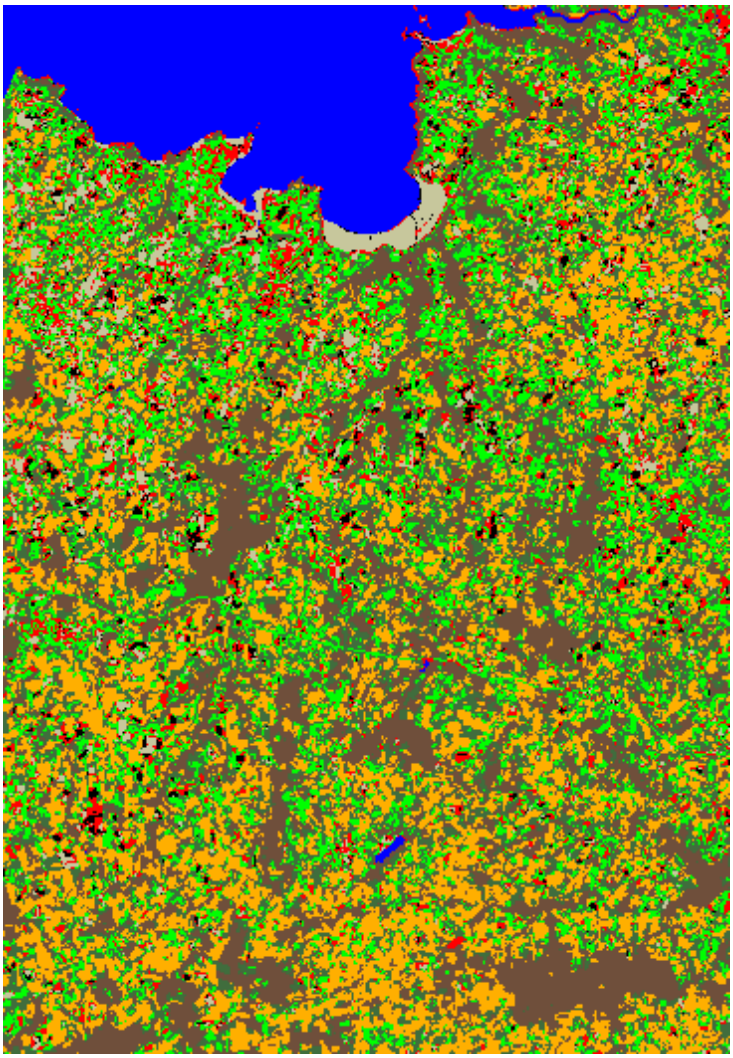


Figure 3.16: Unsupervised segmentation of the SPOT image obtained using the proposed TS-MRF/MS algorithm.

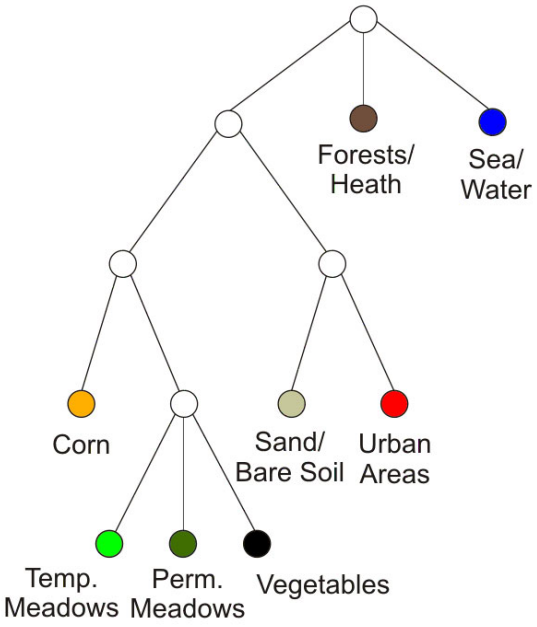


Figure 3.17: Tree structure retrieved for the map of Fig. 3.16.

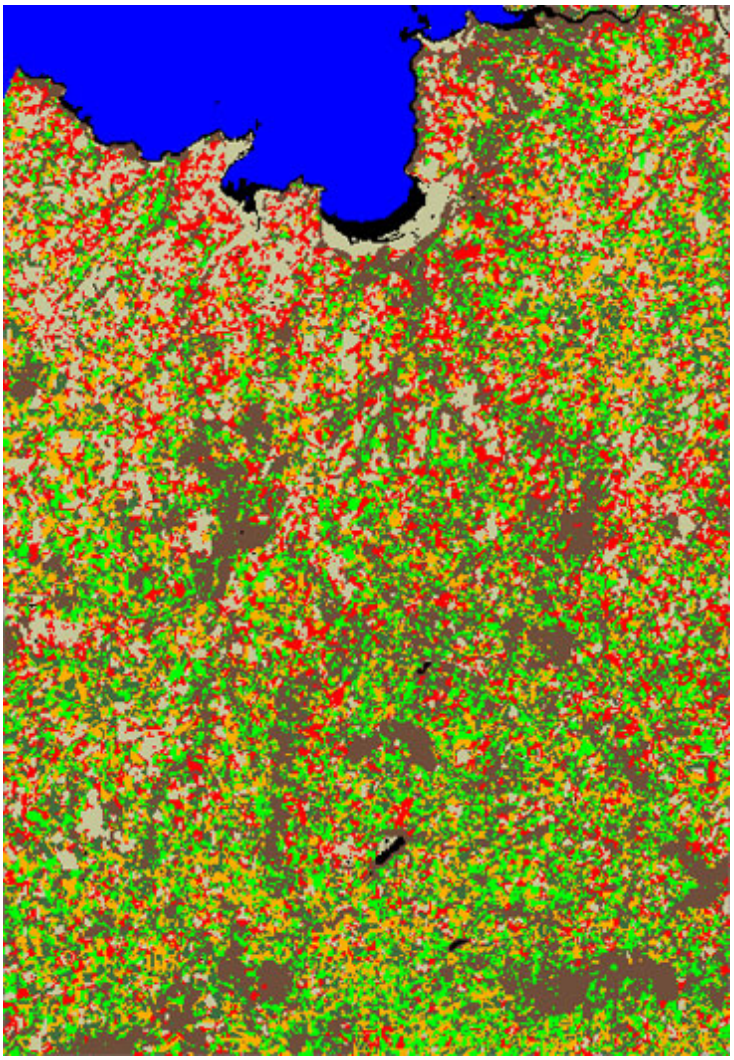


Figure 3.18: 8-class segmentation map obtained through the original TS-MRF (up to 10 classes) with two subsequent manual merging (semi-supervised *split and merge*).

spectral clustering steps.

The final segmentation maps obtained with the original and improved TS-MRF algorithms are reported in Fig. 3.14 and Fig. 3.16, respectively. Fig. 3.15 and Fig. 3.17 instead, show the tree structures detected by both algorithms. Already at a visual analysis, results provided by the proposed version are much more accurate than those of the original algorithm: no major losses are noticeable, at least on top level classes, unlike in the map of Fig. 3.14 where a serious oversplitting of the "forests" class sticks out. Numerical results confirm such empirical observations: the overall classification rate goes from around 60% to 74.4% mainly due to the more precise detection of some large classes, such as the "forests" and "urban areas" classes, as appears from the confusion matrices reported in Tab. 3.3 and 3.4. Improvements are confirmed also by the other overall accuracy figures, being $\kappa = 68.7\%, \tau^{norm} = 65\%$ for the map obtained with new technique, largely outperforming the old one that provides instead $\kappa = 51.6\%, \tau^{norm} = 41.6\%$.

Such an improvement can be likely ascribed to the better segmentation accuracy obtained in the first steps, also due to the more flexible tree structure. As can be seen in Fig. 3.17, in fact, the new technique, by resorting directly to a 3-class top-level split, immediately detects and validates the "forests" class, preventing it from being oversplit in later stages.

To definitely assess the new results, we also decided to compare the new map with the one of Fig. 3.18, obtained using the old technique and arresting the segmentation process when the 10-class map is retrieved, and then manually canceling the two most evident oversplits. This accounts the use of a semi-supervised split-and-merge procedure similar to the one discussed in [74]. Even in this case, the new algorithm outperforms the method described above, as the latter only provides a 71% overall accuracy (see the confusion matrix of Tab. 3.5).

3.4.3 Retrieving the Tree Structure for the Supervised Case

Finally, we present an interesting result concerning the TS-MRF based *supervised* segmentation technique described in [1] and referred to therein as TS/U.

The Supervised TS-MRF Algorithm

The aforementioned technique implements a supervised TS-MRF based segmentation that makes use of some necessary prior information: the number K of classes, statistics on the class-wise spectral distributions of pixels and, of

course, an appropriate tree structure to fit the data into a TS-MRF model. The basic difference between the unsupervised and supervised TS-MRF algorithms stands on the fact that in the latter case no tree structure have to be discovered during the segmentation process, as we dispose of a given one.

To briefly describe the basics of the supervised procedure, let us start from the posterior model introduced in Sec. 2.1.4. Considering a TS-MRF modeling, let X^t be the K_t-ary field associated with node t and $Y^t = \{Y_s : s \in \mathcal{S}, X_s = t\} \subseteq Y$ be the set of data whose labels belong to some descendant class of t, which is known given $x^{\omega(t)}$. As we said before, each X^t can be considered as a K_t-ary Potts field in order to implement the recursive maximization procedure. Indeed, we have to consider a posterior distribution written as:

$$\begin{aligned} p(x^t|x^{\omega(t)}, y^t) \quad &\propto \exp[-\beta_t \mathcal{N}_t] p(y^t|x^t, x^{\omega(t)}) = \\ &= \exp[-\beta_t \mathcal{N}_t] \textstyle\prod_s p(y_s^t|x_s^t, x_s^{\omega(t)}). \end{aligned} \tag{3.16}$$

Here, the likelihood term $p(y^t|x^t, x^{\omega(t)})$ needs to be better defined. In fact, since the descendant fields of node t are unknown for the time being, we are only deciding, for each site, if it belongs to some of the left or right descendant classes, without exactly specifying which one. Therefore, we don't know which normal distributions to use to carry out the test.

To solve this problem, we propose the following "natural" solution for the supervised case. Let us consider the set of children $\{c_i(t)\}_{i=1,\dots,K_t}$ of an internal node t, and define $\gamma(h) = \{t \in \widetilde{T} : \nu(h) \text{ is a prefix of } \nu(t)\}$, the set of the descendant leaves of h. Now we can define the likelihood terms of Eq. 3.16 as:

$$p(y_s^t|x_s^t, x_s^{\omega(t)}) = \max_{k \in \gamma(x_s^t)} p(y_s|x_s = k), \tag{3.17}$$

where $x_s^t \in \{c_i(t)\}_{i=1,\dots,K_t}$ and $p(y_s|x_s = k)$ are the normal densities given in (2.3). In other words, to decide which children node the current site should belong to, the best K_t Gaussian distributions corresponding to "true" classes are considered, being the most likely respectively in $\gamma(c_1(t)), \dots, \gamma(c_{K_t}(t))$. By proceeding so, the tree-structure involves only the prior MRF model while no structural constraints are transferred on the likelihood term $p(y|x)$.

Note that the best fitting Gaussian chosen at this point is only a temporary choice, taken to well fit the data during this intermediate split, but further splits can change such decision based on newly available contextual information.

	W.	B.S.	U.	F.	T.M.	P.M.	V.	C.	*u.a.*
Water	**527**	0	0	1	0	0	0	0	*99,8%*
B. Soil		**1343**	18	0	0	0	0	1	*98,6%*
Urban	0	94	**416**	1	7	17	0	2	*77,5%*
Forests	13	0	0	**1518**	0	0	0	27	*97,4%*
T. M.	0	6	17	1	**221**	117	5	10	*58,6%*
P. M.	0	4	0	11	66	**63**	0	30	*36,2%*
Veget.	15	44	17	11	19	0	**0**	0	*0%*
Corn	0	0	0	48	77	197	0	**436**	*57,5%*
p.a.	*95%*	*90%*	*89%*	*95,4%*	*56.7%*	*16%*	*0%*	*86%*	**83.7%**

Table 3.6: *Confusion Matrix* corresponding to the segmentation map of Fig. 3.19.

	W.	B.S.	U.	F.	T.M.	P.M.	V.	C.	*u.a.*
Water	**544**	0	0	3	0	0	0	0	*99.4%*
B. Soil	0	**1369**	10	0	0	0	0	0	*99.3%*
Urban	0	65	**440**	8	51	10	0	6	*75.8%*
Forests	11	1	0	**1548**	7	1	0	28	*97%*
T. M.	0	9	18	11	**105**	102	5	33	*37.1%*
P. M.	0	12	0	0	101	**78**	0	66	*30.3%*
Veget.	0	43	0	0	33	33	**0**	1	*0%*
Corn	0	0	0	21	93	170	0	**364**	*56.2%*
p.a.	*98%*	*91.3%*	*94%*	*97.3%*	*27%*	*20%*	*0%*	*73%*	**82.3%**

Table 3.7: *Confusion Matrix* corresponding to the segmentation map of Fig. 3.21.

Unsupervised TS-MRF/MS based Tree Building

The supervised procedure has been run here replacing the original binary tree-structure of Fig. 3.19, obtained by visual inspection, with the tree structure of Fig. 3.17 detected by the unsupervised technique proposed in this paper.

For this particular experiment, the labeled pixels available on the ground truth of Fig. 3.11 are divided into two disjoint subsets: the $learningset$ is used to learn the mean spectral response and the inter-band covariance matrix of each category, so that we could perform supervised classifications according to the Gaussian assumption discussed in Sec. 2.1.1, while the remaining samples, the *test set*, is kept to assess the accuracy of the classifications.

The quite accurate segmentation map obtained is reported in Fig. 3.21, showing an overall accuracy of 82.3%, that is only 1.5 points less than the one obtained using the hand-picked tree structure as input of the supervised

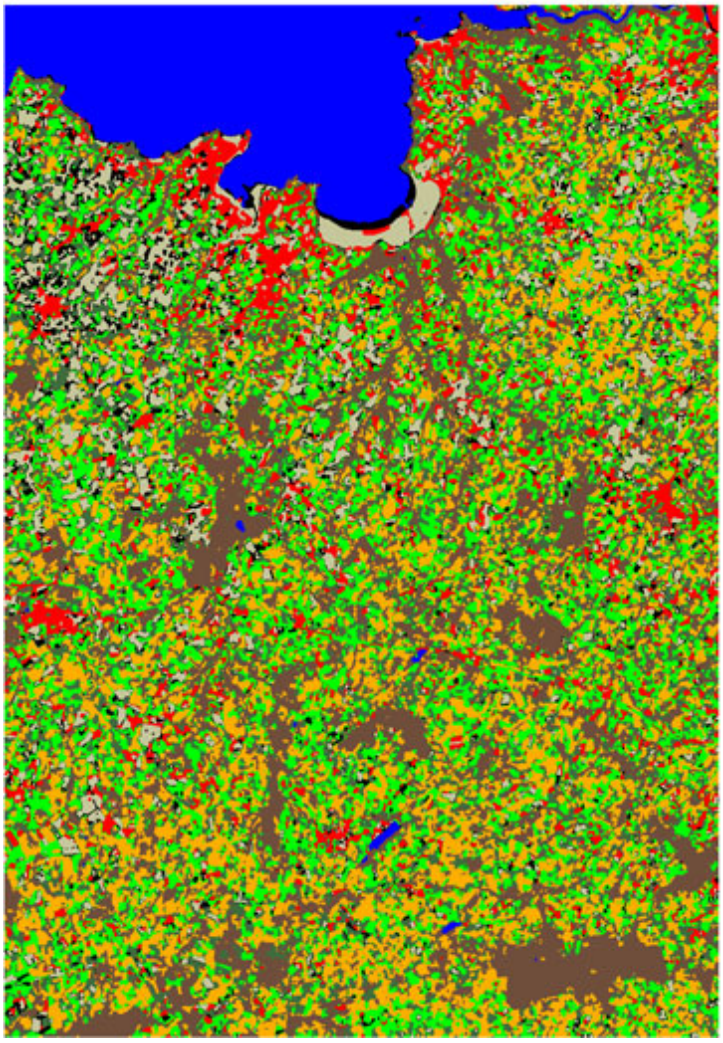


Figure 3.19: Supervised segmentation of the SPOT image obtained using the TS-MRF based algorithm and the tree structure of Fig. 3.20.

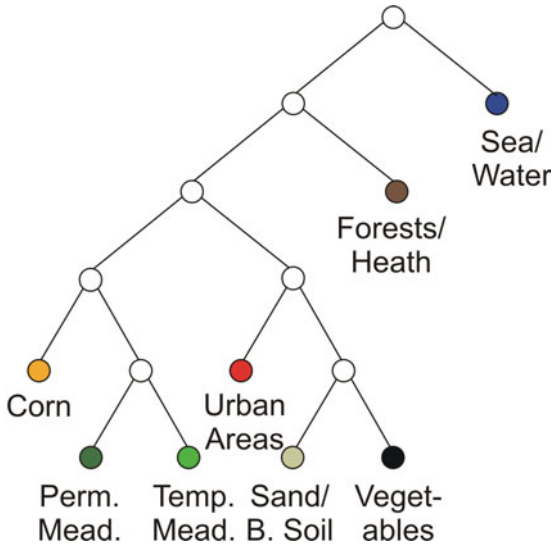


Figure 3.20: Hand picked tree structure used for the original supervised segmentation of Fig. 3.19.

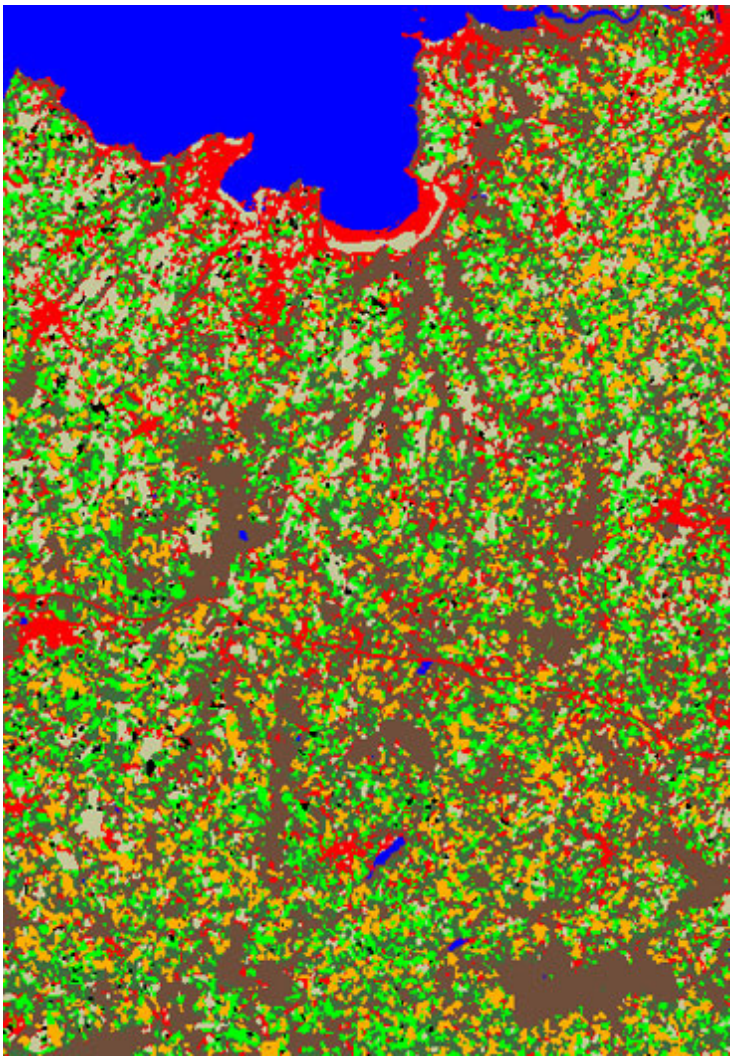


Figure 3.21: Supervised segmentation of the SPOT image obtained using the TS-MRF based algorithm and the tree structure of Fig. 3.17, discovered automatically using the unsupervised TS-MRF/MS algorithm.

process. Confusion matrices of Tab. 3.7 confirm the comparability of the two results also in terms of per-class figures, with some losses occurring only for the *temporary meadows* class.

This seems to show that the tree-structure detected here does fit well the source data and could well be used as a preliminary tool in supervised TS-MRF segmentation, eliminating the need for such a heavy user intervention like providing a tree description of the source data.

Chapter 4

Hierarchical Multiple Markov Chain Models for Texture Segmentation

In this chapter, attention is moved on the problem of texture segmentation*. A new hierarchical model that makes use of a set of Markov chains to describe spatial interactions among texture elements at multiple scales, namely the Hierarchical Multiple Markov Chain (H-MMC) model, is introduced and its properties discussed in detail. The corresponding segmentation algorithm, called* Texture Fragmentation and Reconstruction *(TFR), is therefore presented and its performaces assessed using two different segmentation benchmarks.*

4.1 Hierarchical Texture Modeling

4.1.1 Hierarchical Representation of Textures

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.

4.1.2 The *Hierarchical Multiple Markov Chain* Model

The starting point for the construction of the proposed 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. 4.1 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 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

[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.

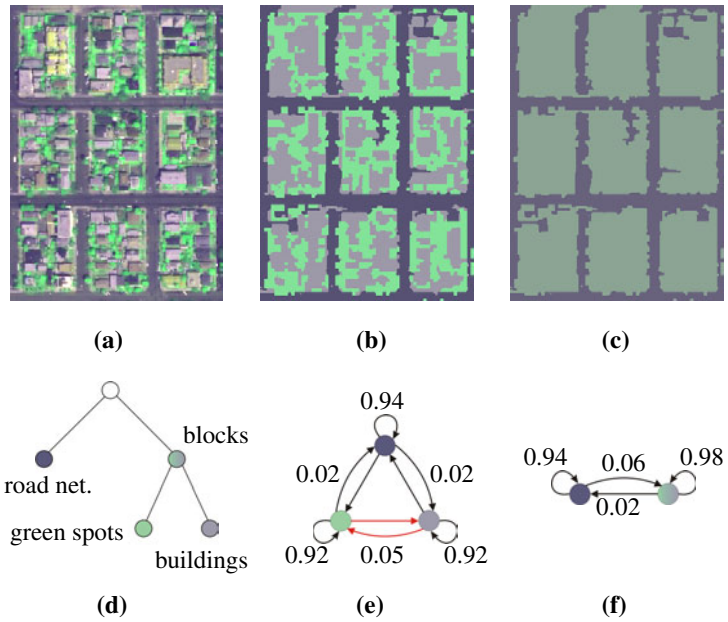


Figure 4.1: H-MMC model: *urban area* sample (a); *3-state* (b) and *2-state* (c) maps; states hierarchy (d); *3-state* (e) and *2-state* (f) Markov chains for the north direction.

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

$$p_j^{(n)}(\omega'|\omega) \stackrel{\triangle}{=} \Pr(x_{s+1} = \omega' | x_s = \omega, \text{ chain} = j) \; \forall \omega, \omega' \in \Omega^{(n)}, \qquad (4.1)$$

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

$$p_j^{(n)}(\omega'|\omega) = \frac{|\mathcal{S}_{\omega \stackrel{j}{\rightarrow} \omega'}|}{|\mathcal{S}_\omega|},$$

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

$$\mathbf{P} = \{\mathbf{P}_j^{(n)} \, : \, 1 \leq j \leq 8, \, 1 \leq n \leq L\}, \qquad (4.2)$$

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. 4.1 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]:

$$\begin{aligned} p(\omega'|\omega) &= \Pr(\omega'_a \cup \omega'_b|\omega_a \cup \omega_b) = \\ &\frac{p(\omega_a)}{p(\omega)}[p(\omega'_a|\omega_a) + p(\omega'_b|\omega_a)] + \\ &+\frac{p(\omega_b)}{p(\omega)}[p(\omega'_a|\omega_b) + p(\omega'_b|\omega_b)], \end{aligned} \tag{4.3}$$

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 (see Eq. 4.4). 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

$$p_j^{(L)}(\omega'|\omega) = \sum_{k=1}^{N_\omega} p_j^{(L)}(\omega'|\omega_k) p(\omega_k), \tag{4.4}$$

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

[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 $\mathcal{T}$.

$$p_j^{(L)}(\omega'|\omega_k) = \frac{|\mathcal{S}_{\omega_k \dashrightarrow j \rightarrow \omega'}|}{|\mathcal{S}_{\omega_k}|} \triangleq A_{\omega_k}(\omega', j) \tag{4.5}$$

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 Eq. 4.5, 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 $\Omega^{(L)}$.

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. 4.2. 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

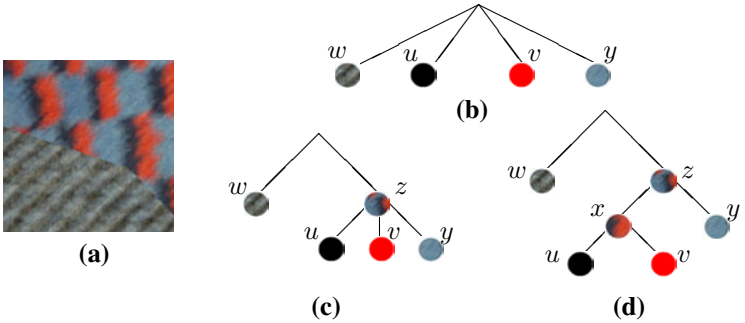


Figure 4.2: Image structure ambiguity. A texture mosaic (a) and several binary (d) and non-binary (b)-(c) hierarchical trees.

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.

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

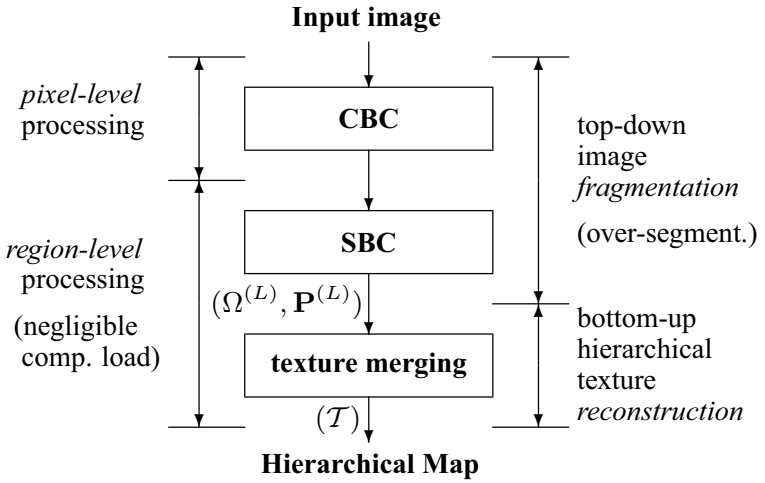


Figure 4.3: TFR flow chart.

4.2 *Texture Fragmentation and Reconstruction*

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. 4.3.

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 interaction with adjacent states) conveyed by the MMC feature set (Eq. 4.5). In order to perform such a classification task, the first CBC block outputs a pixel-by-pixel "color" classification (see Sec. 4.2.1) in K_c color states, also referred to as *partial* (MMC)

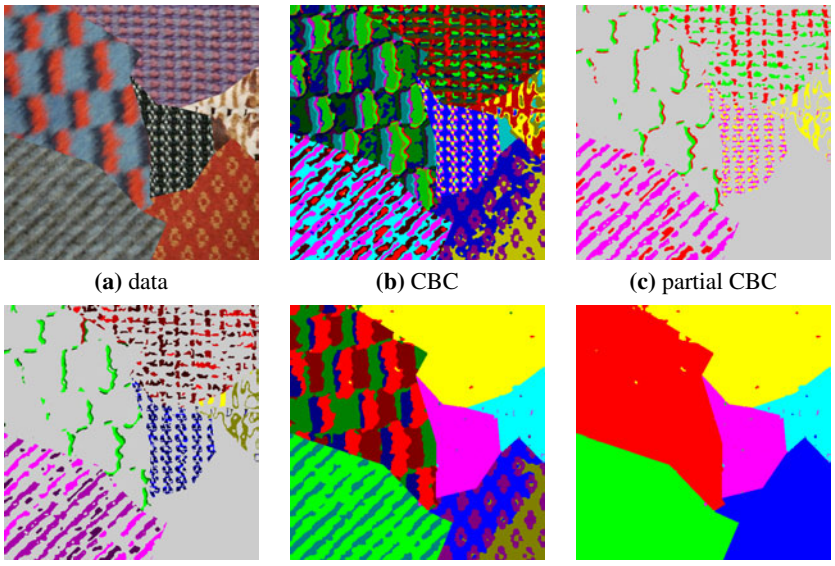
states. At this level each group of adjacent pixels having a same label are assigned to an image "fragment" and all subsequent TFR processing is made considering fragments (rather than pixels) as atomic elements. All contours are therefore fixed in the CBC step, and later, in case, they can only disappear because of region merging. Each color state is therefore further split in K_s (full-defined) states by the SBC block (see Sec. 4.2.2) which operates a clustering aimed at putting together fragments with similar MMC features (Eq. 4.5). 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. 4.2-C/D) can recursively retrieve textures at larger and larger scale.

In order to clarify the overall process an experiment is detailed in Fig. 4.4. 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 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



(a) data **(b)** CBC **(c)** partial CBC

(d) partial SBC **(e)-(f)** hierarchical map (12 and 6 classes)

Figure 4.4: TFR process evolution

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.

4.2.1 Color based Clustering

The color segmentation task (CBC) is here achieved by means of the original version of the TS-MRF model-based unsupervised algorithm presented in Sec. 2.3, because of 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, similar to that of the tree-structured vector quantization algorithm [20], speeds up the processing, ensures convergence to the desired number of classes, and reduces large-scale effects thanks to its progressive localization.

To cope with the specific needs of the TFR algorithm, a cost-free variation of the *splitgain* introduced before is here used to define priorities in the recursive splitting procedure, that only takes into account the largest decrease of overall distortion when fitting data with two local likelihoods instead of one. The only stopping condition lies on the achievement of the desired (*a priori* fixed) number of classes K_c.

4.2.2 Spatial based Clustering

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. 4.5.

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. 4.5 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:

$$F_{\omega_k}(\omega', j) \stackrel{\triangle}{=} \begin{cases} \log[1 - p_j^{(K_c)}(\omega'|\omega_k)], & \omega' = \omega \\ \log[\frac{p_j^{(K_c)}(\omega'|\omega_k)}{(1-p_j^{(K_c)}(\omega|\omega_k))}], & \omega' \neq \omega \end{cases} . \tag{4.6}$$

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 Eq. 4.6 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.

4.2.3 Region Merging

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. 4.2 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.

[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.

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

$$\mathrm{TS}^{\omega} = \frac{p(\omega)}{\max_{\omega' \neq \omega} p(\omega'|\omega)}, \tag{4.7}$$

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).

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

$$\mathrm{TS}^{\omega} = p(\omega) \cdot \frac{1}{p(\bar{\omega}|\omega)} \cdot \frac{p(\bar{\omega}|\omega)}{\max_{\omega' \neq \omega} p(\omega'|\omega)}, \tag{4.8}$$

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

$$\omega^* = \arg\max_{\omega \neq \widehat{\omega}} p(\omega|\widehat{\omega}). \tag{4.9}$$

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 [75] which refers directly to the spatial scale of the classes by defining a suitable threshold for the texture score.

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:

$$D(p\|q) \stackrel{\triangle}{=} E_p\left[\log\frac{p(\mathbf{x})}{q(\mathbf{x})}\right] = \int p(\mathbf{x})\log\frac{p(\mathbf{x})}{q(\mathbf{x})}d\mathbf{x}, \tag{4.10}$$

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:

$$\log \mathrm{TS}^\omega_{\mathrm{KL}} \stackrel{\triangle}{=} \min_{\omega'\neq\omega}\left\{\log\frac{p(\omega)}{p(\omega'|\omega)} + D(q_\omega\|q_{\omega'})\right\}, \tag{4.11}$$

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 [76]:

$$D(p\|q) = \frac{1}{2}(\log\frac{|\Sigma_q|}{|\Sigma_p|} + \mathrm{tr}(\Sigma_q^{-1}\Sigma_p) + (\mu_p - \mu_q)^T\Sigma_q^{-1}(\mu_p - \mu_q) - d) \quad (4.12)$$

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.

4.3 Benchmarking TFR

4.3.1 Application to the Prague Segmentation Benchmark

The Prague segmentation benchmark [77], 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 [78, 73, 77] 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.

Accuracy assessment criteria

The region-based criteria [78] compare the machine segmented regions $\mathcal{R}_i, i = 1, \ldots, M$ with the correct ground truth regions $\bar{\mathcal{R}}_j, j = 1, \ldots, N$. Two regions of different maps correspond to each other depending on their overlapping degree. In particular, when this degree is larger than a fixed threshold $k \in [0.5, 1]$ (0.75 by default, but also full sensitivity curves and their integrals are

available), a correspondence between the regions is assumed. Based on this region matching principle the following *region-based criteria* are defined:

- CS, rate of correct (region) detection;
- OS, over-segmentation;
- US, under-segmentation;
- ME, missed regions;
- NE, noise region.

Other criteria are, instead, based on pixel-wise accuracy indicators. Normally these indexes are applicable to the supervised case, where the correspondence between the classes of the machine segmentation map and those in the ground-truth is fixed a priori, so that the accuracy indicators can be properly computed. However, they can be used in the unsupervised case as well, if a correspondence between the classes of the two maps can be established, even when the number of classes is not the same. In particular, to achieve this goal the benchmark system applies the Munkres assignment algorithm [79]. The following *pixel-wise criteria* are implemented:

- O, omission error, the overall ratio of wrongly interpreted pixels;
- C, commission error, the overall ratio of wrongly assigned pixels;
- CA, weighted average class accuracy;
- CO, recall, the weighted average correct assignment;
- CC, precision, object accuracy, overall accuracy;
- I, type I error, the weighted probability of wrong assignment of classes pixels;
- II, type II error, the weighted probability of commission error;
- EA, mean class accuracy estimate;
- MS, mapping score, emphasizes the error of not recognizing the test data;
- RM, root mean square proportion estimation error, indicates unbalance between omission and commission errors;

- CI comparison index, includes both object precision and recall, and reaches its maximum either for the ideal segmentation or for equal commission and omission errors for every region (class).

A potential problem for a measure of consistency between segmentations is that there is no unique segmentation of an image. For example, two people may segment an image differently because either they perceive the scene differently, or they segment at different granularities. If two different segmentations arise from different perceptual organizations of the scene, then it is fair to declare the segmentations inconsistent. If, however, one segmentation is simply a refinement of the other, then the error should be small, or even zero. Based on this consideration some *consistency measures* were defined in [73], which are:

- GCE, the global consistency error;
- LCE, the local consistency error.

Reference segmentation algorithms

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

GMRF/EM (Gaussian MRF model with EM) [34]. 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.

AR3D/EM (3-D Auto Regressive model with EM) [80]. This algorithm is similar to the previous one, but the GMRF model is replaced by a 3-D autoregressive model, thus spectral space correlations can be modeled without approximating the spectral information.

JSEG [44]. 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.

SWA (Segmentation by Weighted Aggregation) [28]. 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.

Blobworld [81, 82]. This is the basic segmentation tool used in the content-based image retrieval system *blobworld* [82]. 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.

EDISON (Edge Detection and Image SegmentatiON system) [83]. 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 [19] with a generalization of the traditional Canny edge detection procedure [84], which employs the confidence in the presence of an edge [85].

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. 4.7 and Eq. 4.11 respectively.

The benchmark data set is composed of twenty different 512×512 texture

	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 4.1: 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.

mosaics, seven of which are shown in Fig. 4.5 - 4.14 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.4.1.

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 Sec. 4.2. 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 [75] 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

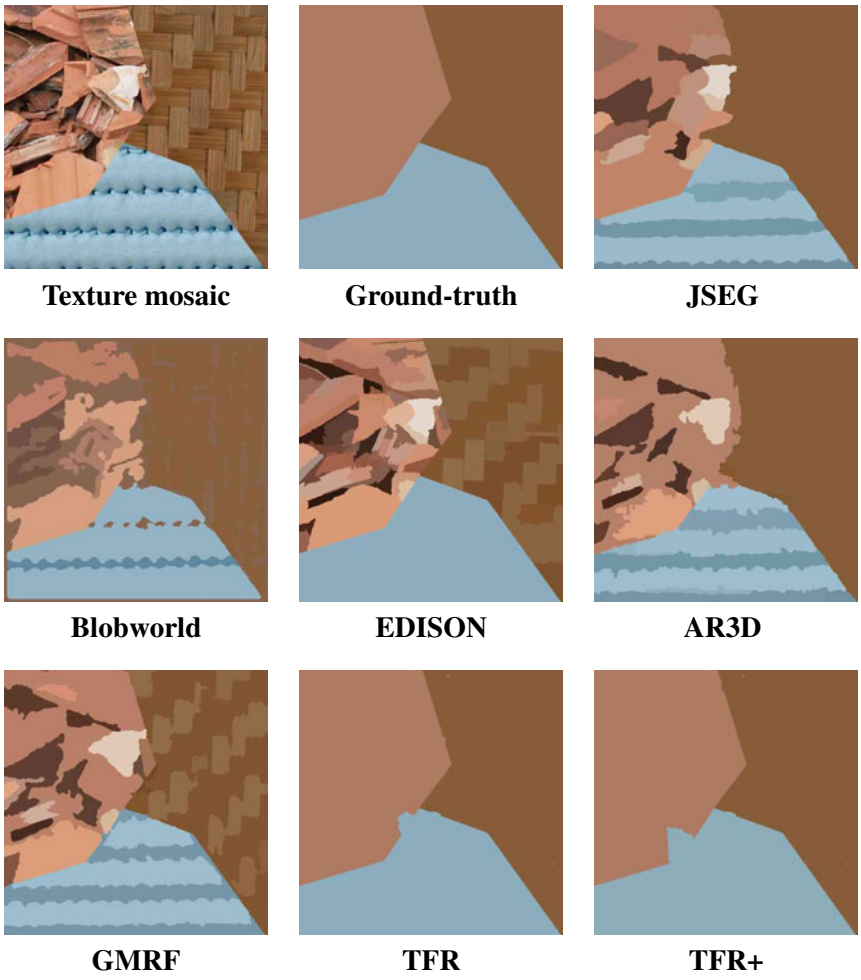


Figure 4.5: Texture mosaic No.1: data, ground-truth and segmentations.

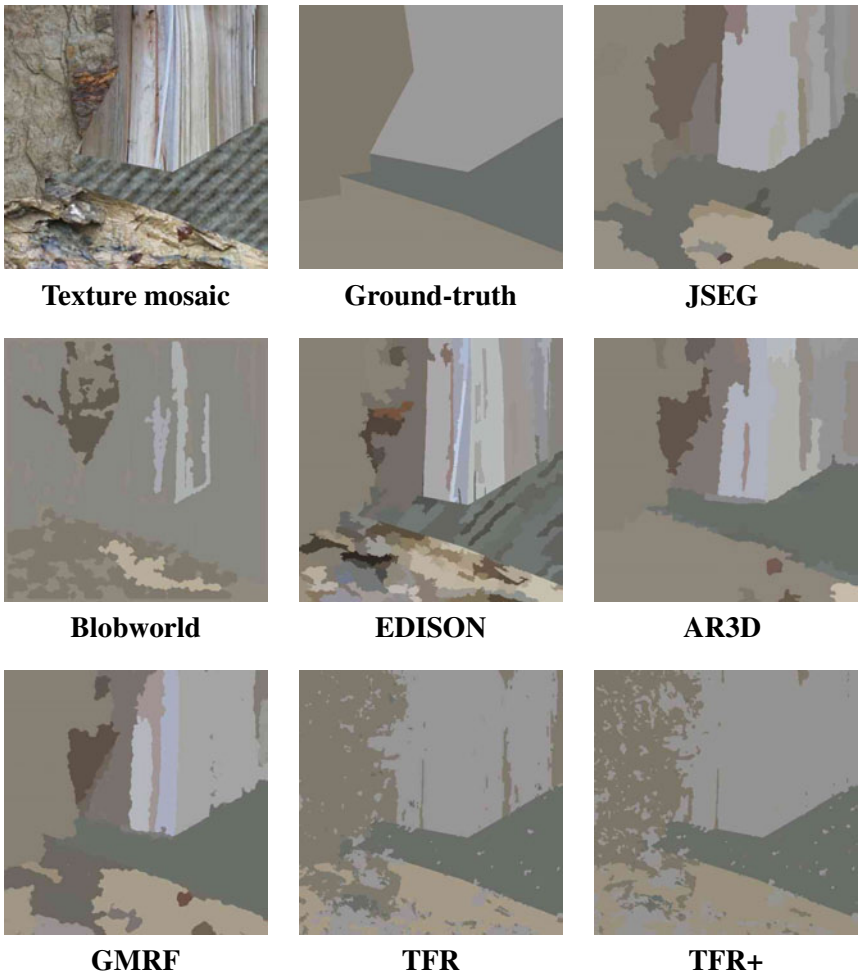


Figure 4.6: Texture mosaic No.2: data, ground-truth and segmentations.

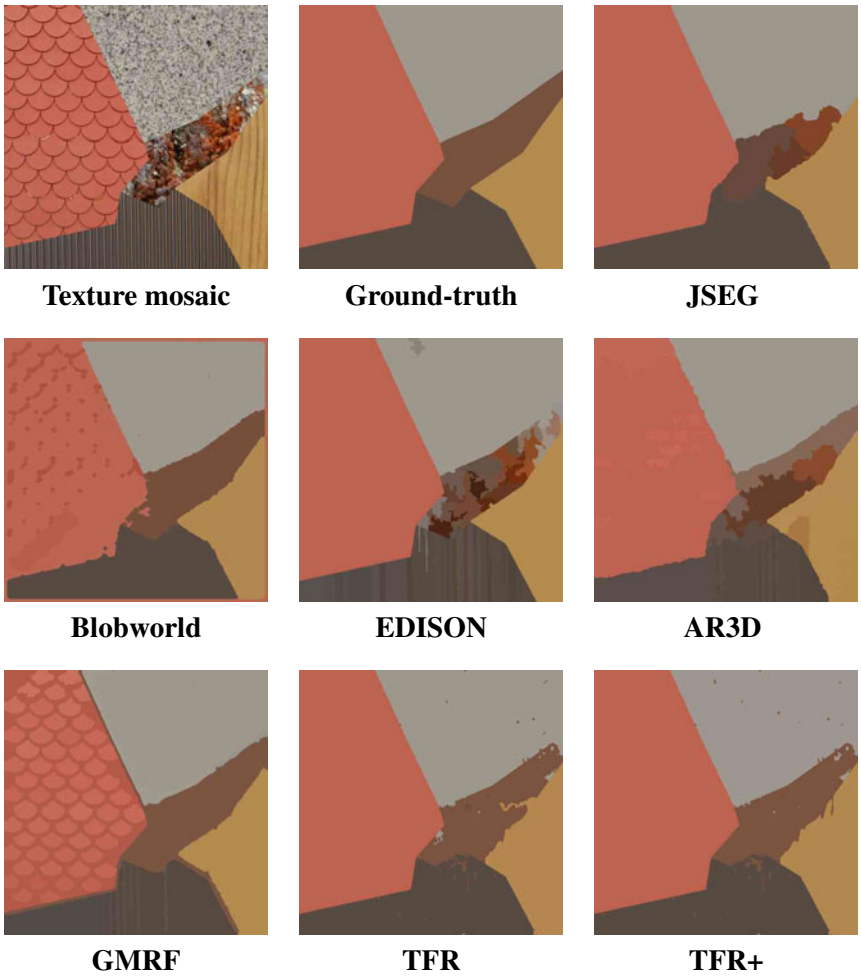


Figure 4.7: Texture mosaic No.3: data, ground-truth and segmentations.

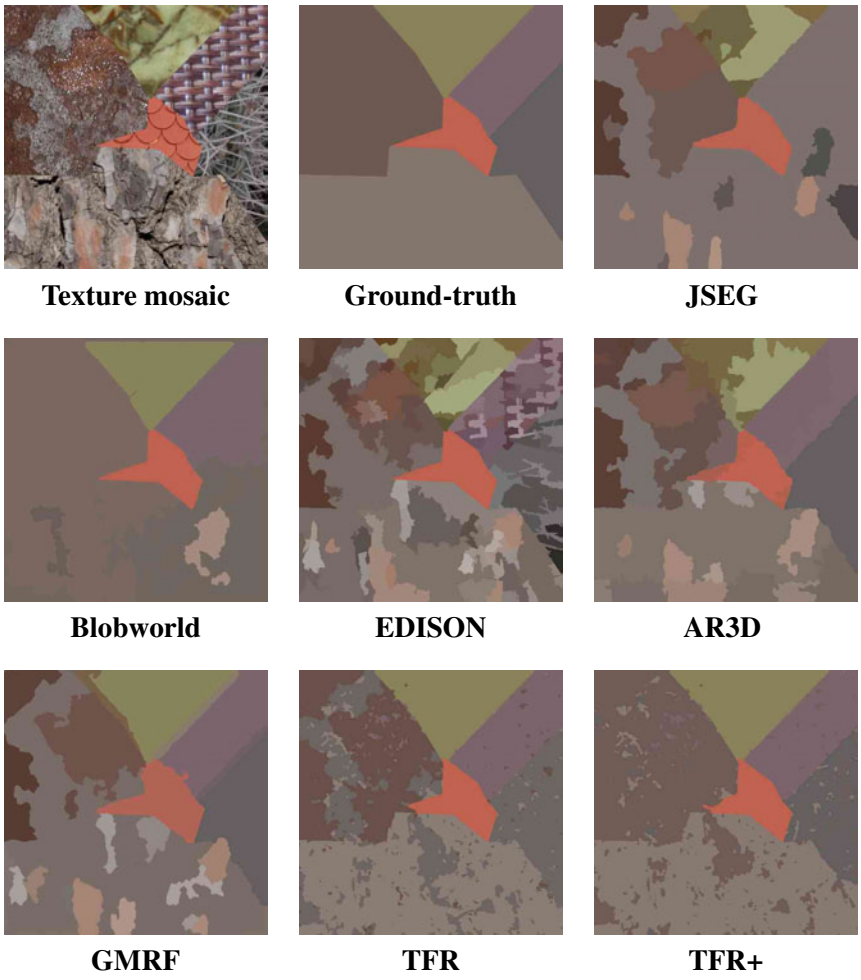


Figure 4.8: Texture mosaic No.4: data, ground-truth and segmentations.

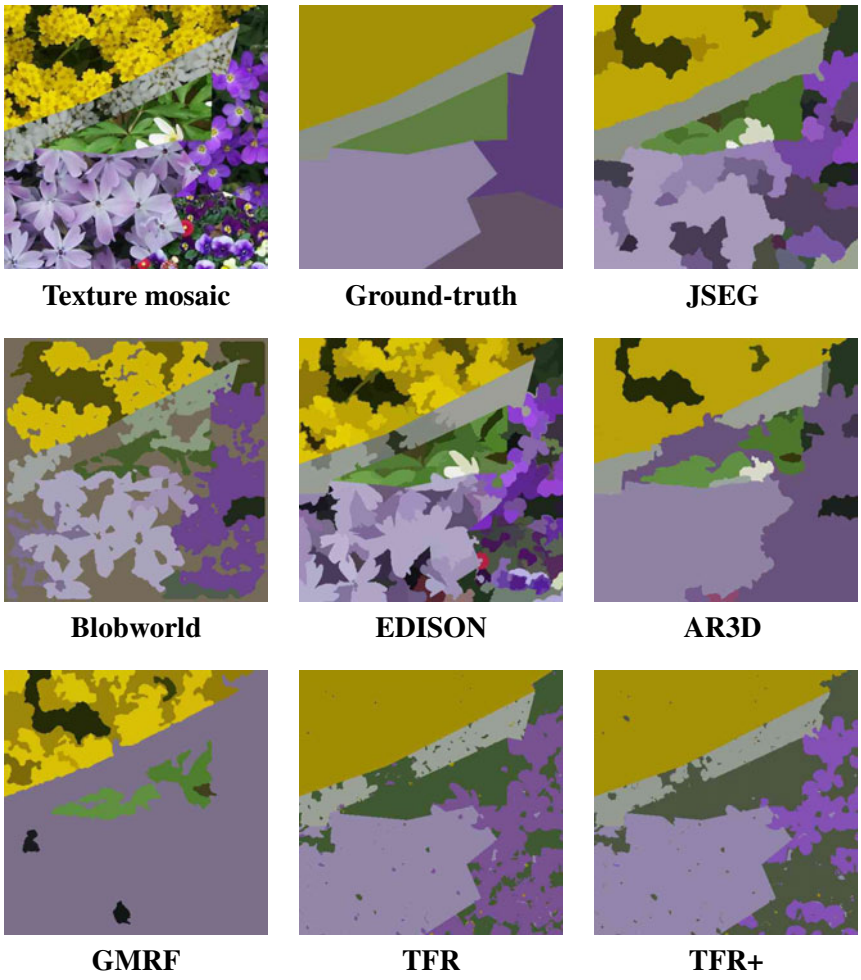


Figure 4.9: Texture mosaic No.12: data, ground-truth and segmentations.

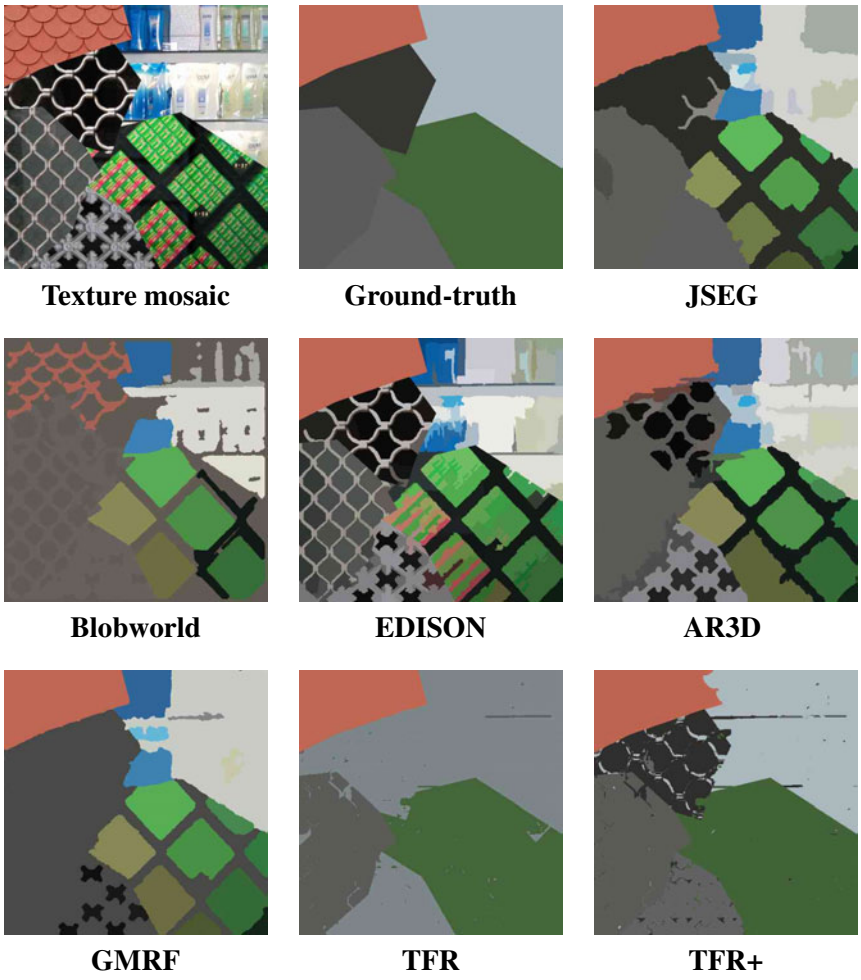


Figure 4.10: Texture mosaic No.14: data, ground-truth and segmentations.

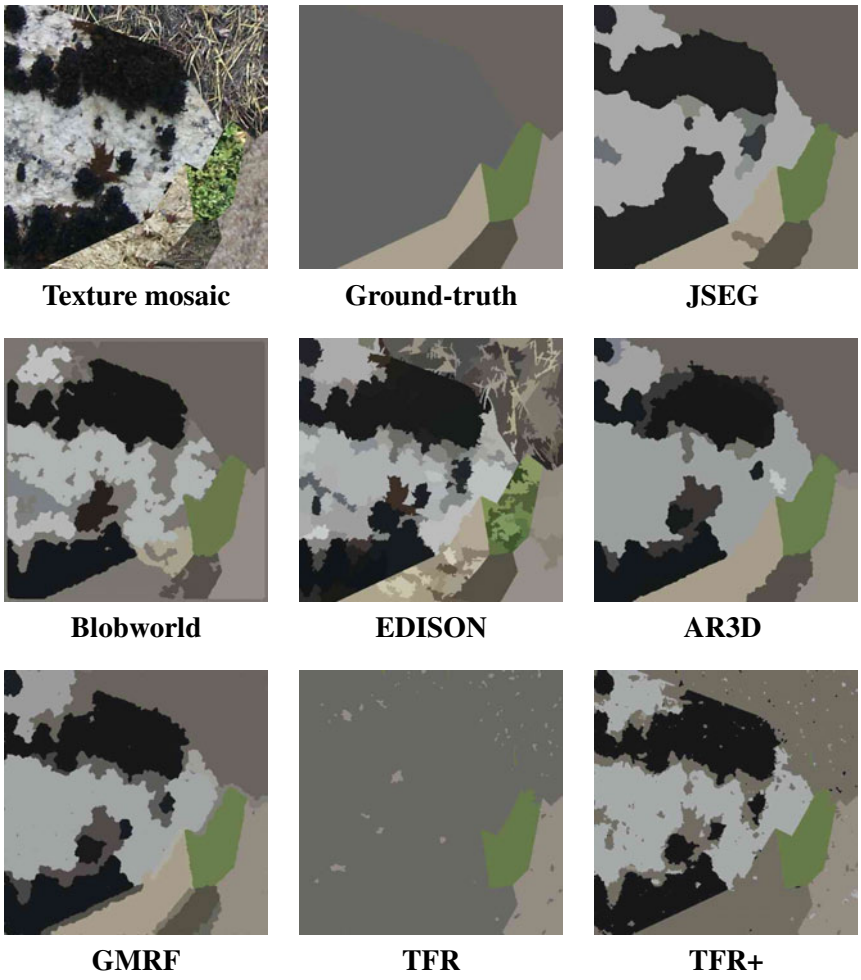


Figure 4.11: Texture mosaic No.15: data, ground-truth and segmentations.

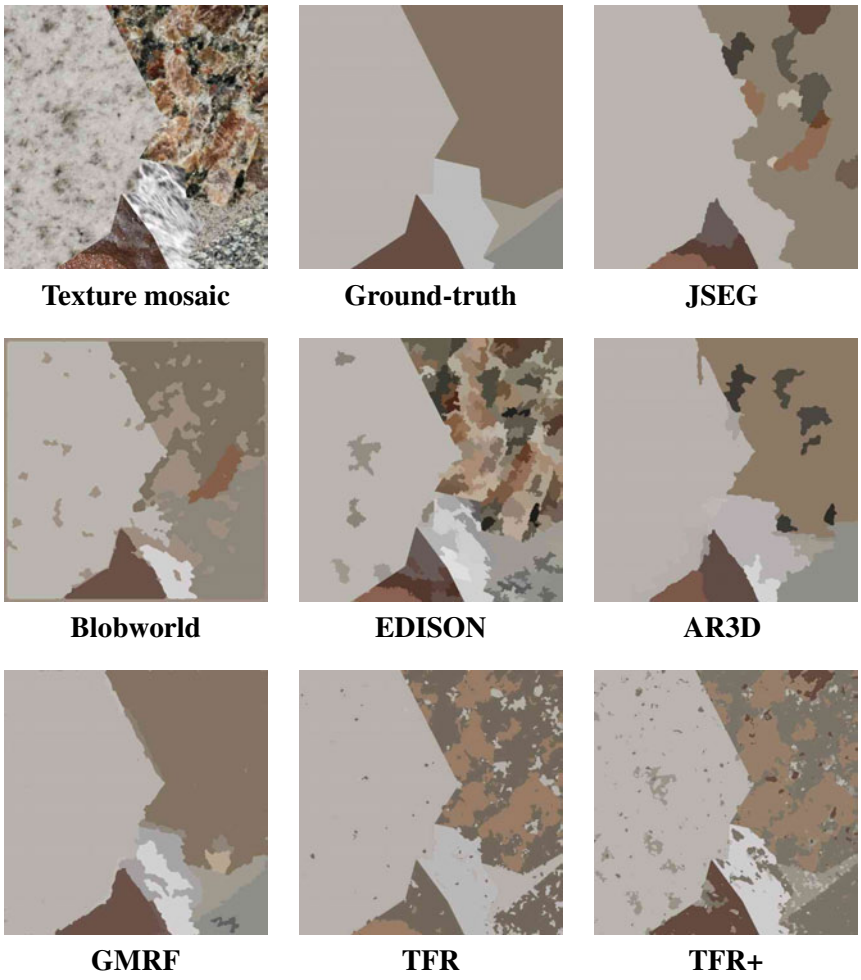


Figure 4.12: Texture mosaic No.18: data, ground-truth and segmentations.

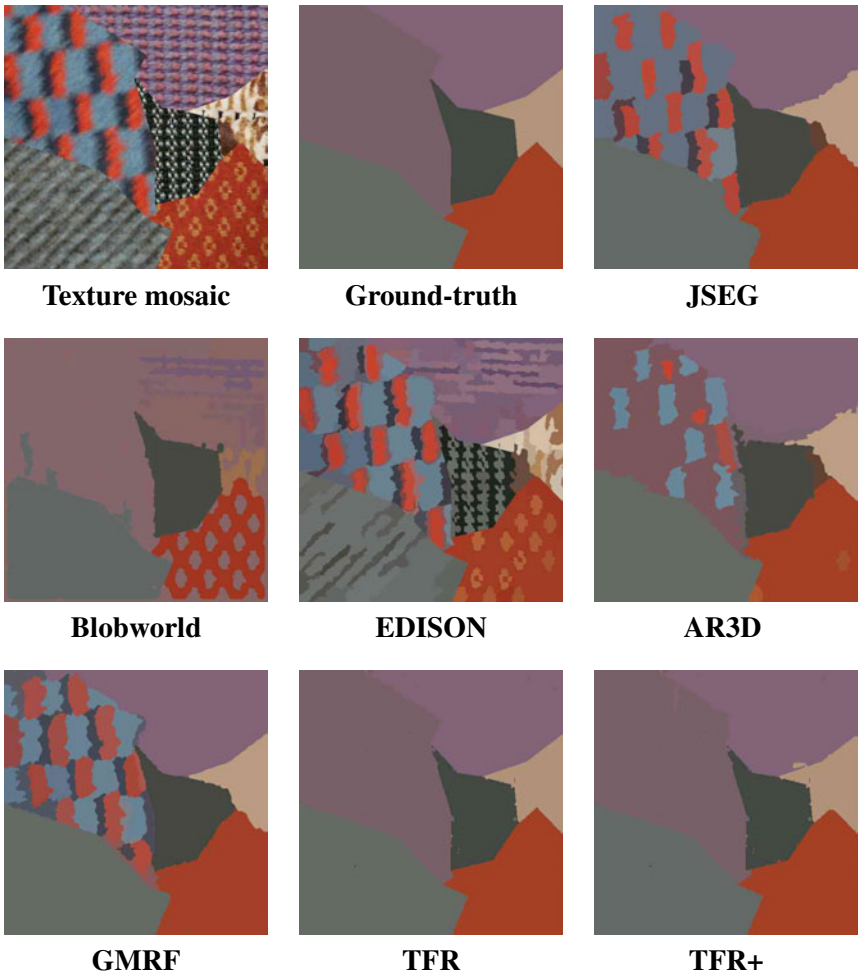


Figure 4.13: Texture mosaic No.19: data, ground-truth and segmentations.

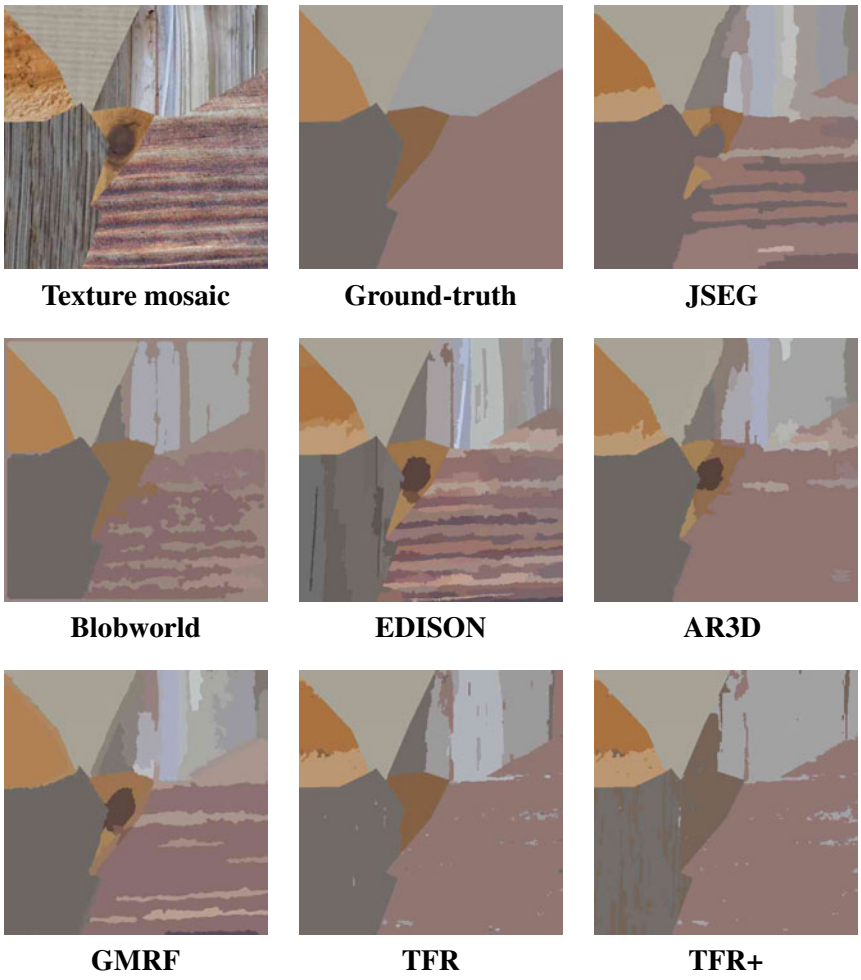


Figure 4.14: Texture mosaic No.20: data, ground-truth and segmentations.

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.

The visual inspection of the segmentation maps shown in Fig. 4.5 - 4.14 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. 4.1).

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. 4.5 - 4.14, consider the 6th mosaic, where the green blocks on a black background are separated by all reference methods.[6] 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. 4.1, 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 ($OS = 86.91$). 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, the highest under-segmentation index $US = 23.99$

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

is achieved by TFR (see also the texture mosaic nr. 14, Fig. 4.10, 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. 4.1 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.

4.3.2 Application to the Berkeley 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 [73].

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 Sec. 4.2, we set $K_c = 12$ and $K_s = 6$.

Experimental results for some test images are reported in Fig. 4.15 - 4.18. 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 [73].

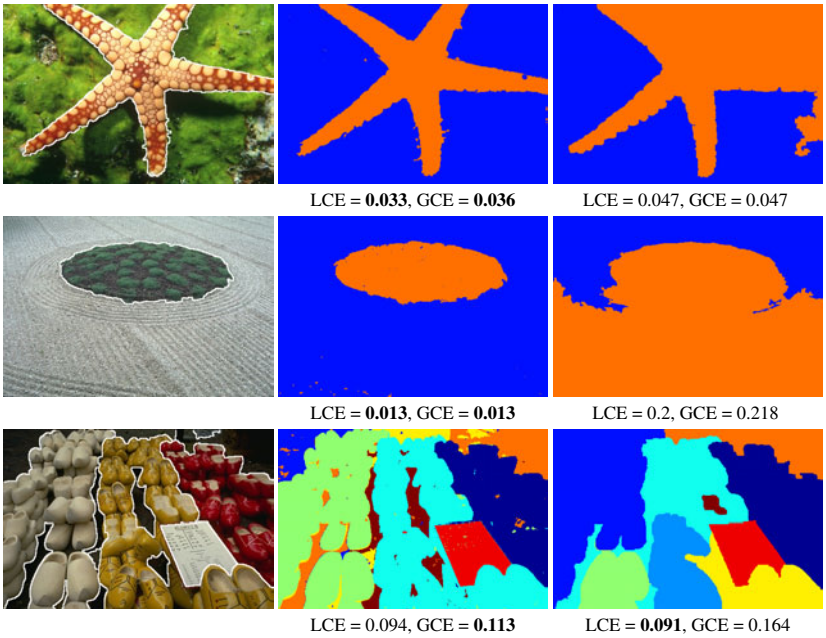


Figure 4.15: Segmentation of natural images #12003, #86016 and #140075 taken from the Berkeley Segmentation Dataset: original image (left), best result obtained using the TFR algorithm (middle) and the SWA algorithm (right). Below each image the mean *Local* and *Global Consistency Errors* (LCE and GCE) are reported (in bold, the best values for each experiment).

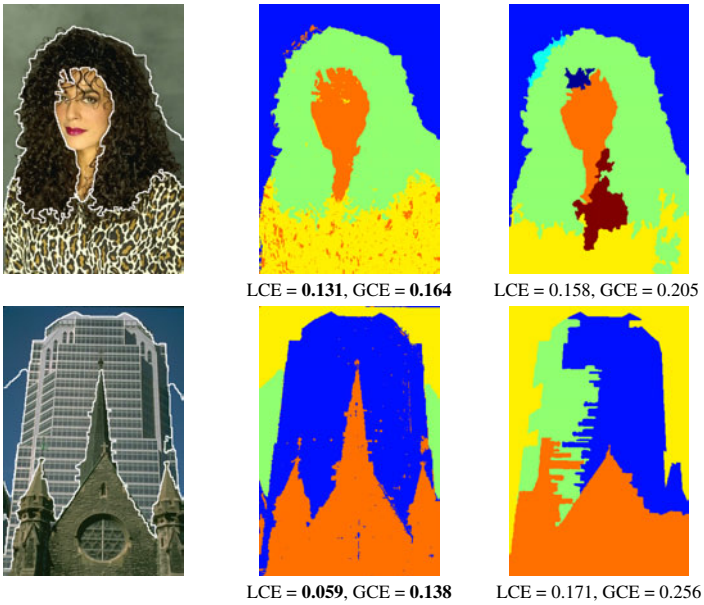


Figure 4.16: Segmentation of natural images #198054 and #277095: original image (left), best result obtained using the TFR algorithm (middle) and the SWA algorithm (right).

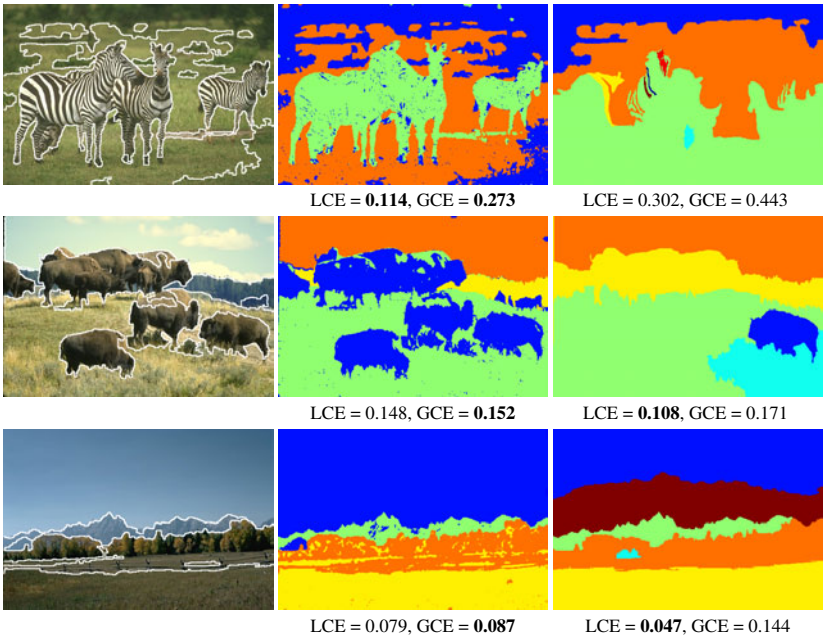


Figure 4.17: Segmentation of natural images #253027, #38092 and #2092: original image (left), best result obtained using the TFR algorithm (middle) and the SWA algorithm (right).

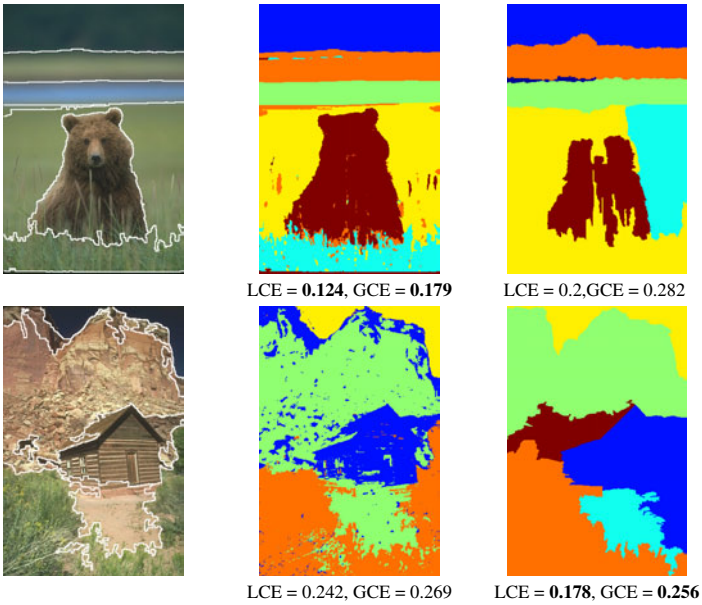


Figure 4.18: Segmentation of natural images #100080 and #254054: original image (left), best result obtained using the TFR algorithm (middle) and the SWA algorithm (right).

Chapter 5

Hierarchical Segmentation of Multiresolution Remote Sensing Images

The Texture Fragmentation and Reconstruction algorithm introduced in the last chapter has proved to achieve good performances in segmenting images with a rich textural content. In this chapter, a real-life application of TFR is presented in the domain of high-resolution remote sensing images, with specific reference to multi-resolution data provided by the new generation Ikonos sensors. Issues concerning the modification to the original TFR algorithm to operate on this kind of data are here discussed, and results obtained in the unsupervised classification of urban/peripheral scenes are finally presented.

5.1 Advances in Remote Sensing Image Segmentation

Sensors of the last generation, with spatial resolution as high as 0.6 m, are giving new impulse to standard applications concerning the analysis, interpretation and classification of remote sensing imagery. Land classification and change detection, for instance, are now typically aimed at providing thematic maps very rich in detail; the extraction of specific land coverage information, such as roads and buildings in urban areas [86, 2], or vegetation classification in rural and non-urban contexts [87, 88, 89], is often required for monitoring purposes or for the updating of Geographical Information Systems. In addition, several new applications directly stem from the availability of highly

detailed optical data, like the extraction and counting of tree crowns in plantations for forest inventory purposes [90], or the detection of urban structures in remotely sensed scenes as in [91].

Many of these image analysis applications require a prior segmentation process which provide a conceptual object-based or class-based map. For segmentation, as for most remote sensing image analysis problems, the availability of high-resolution images has changed both the expectations on the nature and quality of the results and the approaches and tools used to deal with the problem. In fact, the high resolution allows for a more precise detection of boundaries, and hence a finer definition of the regions of interest, possibly at multiple scales of observation, but, on the other hand, calls for new solutions to cope with the increased complexity and new peculiarities of these data.

A particularly relevant problem to deal with is the reduced spectral resolution exhibited by this new generation of sensors, a technological limitation that would bar the use of classical spectral-based segmentation approaches, *e.g.,* [1, 92, 93], highly successful with low- and mid-resolution data. This problem can be circumvented by resorting to a single system which provides both a high-resolution single-band, or "panchromatic" (PAN) image, and a low-resolution multispectral (MS) image. Notable examples are the Ikonos satellite, with a 1 m resolution panchromatic image complemented by a 4 m resolution multispectral image, and the Quickbird satellite, with even higher resolutions (0.6 m and 2.4 m). It is worth underlining the shift of paradigm implied by this solution: since no instrument is able to provide data with the desired resolution *both* in the spatial and spectral domains, the task is passed on to subsequent signal processing steps that are asked to improve the data usability

As far as segmentation is concerned, the goal is to obtain a map with the high geometric resolution of the panchromatic image, but also with the reliability guaranteed by the richer spectral information of the multispectral data, so the problem is how to perform an intelligent fusion among the available multiresolution data.

5.1.1 Exploiting Multiresolution Data for Segmentation

In principle, the problem should be addressed by using jointly all available data, that is, by resorting to a truly multiresolution segmentation algorithm, such as those proposed in [94], where observable data are associated with the various layers of a tree-structured Markov random field, and all tree nodes are then labeled at once. Besides the obvious modeling hurdles, the optimization

task in such a setting is exceedingly complex, and in fact it is usually tackled by means of strictly causal models on quadtrees, such as those proposed in [95, 96] which typically produce some blocking artifacts.

Indeed, computational complexity is a major discriminant, together with accuracy, when designing a real-world segmentation algorithm. For this reason, most segmentation/classification techniques for multiresolution data resort to some structural simplifications. A first approach is to use a *pansharpening* technique, such as those proposed in [97] or [98], followed by some well-known segmentation method for full-resolution multispectral images, like JSEG [44], the FNEA algorithm embedded in eCognition [99], or the TS-MRF algorithm [18]. Examples of this approach can be found in [86] and [2]. Of course, this additional processing phase may introduce new errors, that will inevitably affect the final segmentation accuracy; in addition, the computational complexity may increase significantly since a complete high-resolution datacube must be dealt with.

Another approach is to use a two-step procedure in which the low-resolution multispectral image is segmented first, while the panchromatic image is used in a second stage to refine the initial coarse segmentation map. In [88], for example, the low-resolution thematic map obtained by working on the multispectral image is later refined by incorporating a detailed edge map extracted by the panchromatic image. By so doing, however, the spectral information is given priority w.r.t. the spatial information, since high-resolution panchromatic data are used only to refine a coarse segmentation obtained on multispectral data. Such an approach is effective when the spectral information is actually more relevant than the spatial one, namely, when typical objects in the image are large enough w.r.t. the data resolution, much less effective when the original image is very rich in fine details (think of urban areas). In [100], for example, where contour refinement is carried out with the help of geometrical constraints specific for urban areas, it is pointed out that *"When the input to the geometric refinement is not accurate, the output improves only partially."* This problem is also recognized in [101] and [102] where, in the context of multiresolution processing of single-level data, the problem of adaptively selecting the best resolution for feature extraction is addressed.

Because of these observations the solution that we propose here follows the opposite path. It operates first on the high-resolution panchromatic data, decomposing the image into a collection of elementary regions, and subsequently enriches the region description by adjoining spectral features extracted by the multispectral data. By so doing, we aim at fully exploiting the high-

resolution of data because boundaries are detected with high geometrical precision, spatial features, such as the shape and orientation of regions, can be easily extracted, and relationships among regions can be analyzed in order to identify and use textural properties. Our choice is also justified by complexity concerns, since the segmentation of such large images becomes heavier if vector-valued data are involved. The initial segmentation is carried out by means of a recently proposed [18] low-complexity contextual algorithm, based on the tree-structured Markov random field (TS-MRF) model [60, 18], and a straightforward technique is then used to associate spectral information with each segment.

5.1.2 Providing a Multiscale Segmentation

Although the low-level segmentation map obtained at this point can be already of interest, more refined products are often required for actual use in high-end applications.

When the application is known in advance, one can easily single out some features of special relevance for the problem, such as object shapes, repeated patterns, or other geometrical properties, that drive the image processing scheme. Notable examples are [103], where a urban-vs-rural classification technique is proposed based on the joint use of features like length and orientation of straight line edges and spectral-based vegetation indexes, or [104], where urban areas classification is carried out based on the spectral coherence of groups of pixels along selected directions, or again [89], where the identification of forest density (dense, sparse and empty) is pursued by defining a suitable set of morphological operators that enhance specific textural properties.

However, segmentation techniques conceived to work for a variety of different tasks, image sources, and scenes, cannot rely on such specific features, and hence a more general approach must be considered. To this end, some techniques have been recently proposed which try to model the problem of segmentation in a hierarchical fashion. By looking at the scene under multiple scales of observation, different objects and features can emerge at the various scales, and be related with one another according to some suitable criteria in a hierarchical structure. Dealing with high-resolution satellite images, for example, the main environments, such as urban areas, rural zones, or forests, can be identified at coarser levels, while more detailed structures, such as buildings and roads in urban areas or trees in forests, will emerge at finer levels.

Hierarchical segmentation is certainly not a new idea. Back in the sev-

enties, for example, Horowitz and Pavlidis [105] began to combine recursive splitting with region merging. In [106], segmentation is performed through a region merging process carried out by hierarchical stepwise optimization. Likewise, the algorithm proposed in [107] integrates information from edges and regions in the framework of a hierarchical image partition. It also worth mentioning that the concept hierarchical segmentation does not apply only to regions but also to objects, as shown in a very recent work [108] where the goal is to detect complex urban structures.

These techniques show very clearly the potential of a multiscale approach in the segmentation of high resolution remote sensing images. However, they all aim at retrieving the largest possible *homogeneous* regions (including those characterized by homogeneous micro-textures) present in the image. Hence, they are unable to recognize and extract more complex regions, characterized by large-scale textures, which appear quite frequently in remote-sensing as well as in other images.

The use of the *Hierarchical Multiple Markov Chain* model introduced in the previous chapter, along with the deriving *Texture Fragmentation and Reconstruction* framework, appears to be a reasonable and natural solution to cope with these limitation and provide a rich multiscale description of high resolution remotely sensed scenes. In the following of this chapter, the application of TFR to this domain is therefore discussed, focusing on the necessary modifications to the basic algorithm to deal with data at multiple resolutions, and some results are presented that show the potential of the proposed solution, both in terms of richness of the description and segmentation accuracy.

5.2 The modified TFR Algorithm

The texture-based image model and segmentation algorithm described in Chapter 4 rely on quite general properties, and hence can be applied to a wide variety of images. Multiresolution remote-sensing images, with their wealth of fine details and textures, appear as the perfect candidates for using these tools. Before doing this, however, we must address the key issue of how to exploit jointly the various sets of data available, characterized by different resolutions and different spectral contents, in order to devise a viable and reliable segmentation algorithm.

As already discussed in Sec. 5.1.1, we avoid the use of any prior pansharpening step in order not to introduce artifacts that could affect the quality of the overall segmentation. For the very same reason, the first piece of informa-

tion taken into account in our processing scheme will be the high-resolution panchromatic image: its over-segmentation will hopefully preserve all image contours, providing a preliminary map containing all the elementary fragments of the scene. Only in a later stage, the spectral information from the low-resolution multispectral image will be injected onto this map, by means of a region-level data fusion, providing a full region-based characterization of the segmented image.

The overall segmentation algorithm can be summarized by the block diagram shown in Fig. 5.1. The first three steps of the procedure basically replace the CBC block of the original TFR algorithm (see Fig. 4.3): after the initial gray-level based segmentation of the panchromatic image, the fusion with the multispectral data takes place, followed by a spectral clustering phase based on the enriched features by now available. The final spatial clustering and merging processes, which are not peculiar of multiresolution images, are the same outlined in Sec. 4.2. As a by-product, a simple "color" segmentation map is also available, which could be used, for example, as a support for a possible region-based adaptive pansharpening.

It has to be noticed that the choice of using only the panchromatic (scalar) data in the first step has the important effect of keeping limited the computational complexity of the new CBC block, where pixel-wise processing is performed on the source, often quite significant especially in this domain where very large images can be taken under analysis.

In the following subsections we describe the algorithm in detail, with special attention for the first three steps which are peculiar of multiresolution images.

5.2.1 Segmentation of the Panchromatic Image

As in the original TFR algorithm, segmentation of the panchromatic image is here performed by means of the unsupervised TS-MRF algorithm introduced in Sec. 2.3. Motivations remain the same introduced in Sec. 4.2.1 for the general TFR framework. Concerning our choice to use only the scalar panchromatic image to derive the elementary fragments of the scene, it is motivated by the fact that resorting to the data with the highest resolution, unaltered by any pansharpening procedure, helps preserving fine object contours and, as a consequence, correctly detecting the elementary structures of the image.

It goes by itself that the limited spectral content of the panchromatic data increases the risk of not distinguishing regions of different nature but with close gray levels. We reduce this risk by resorting to a moderate over-

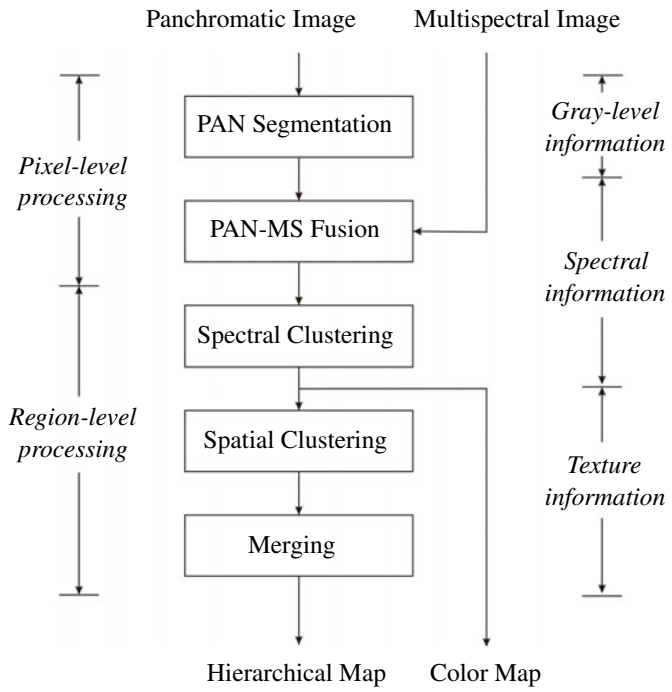


Figure 5.1: Block diagram of the proposed segmentation technique, with current processing level (left), and current source information (right).

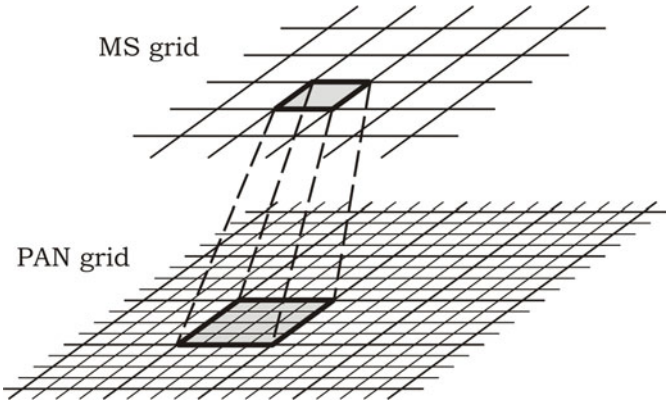


Figure 5.2: Relationship between the multispectral (MS) and panchromatic (PAN) image grids, under the hypothesis of perfect source registration.

segmentation, and take care of the remaining errors after the PAN-MS fusion, by detecting them and carrying out a local refinement, as explained in the next subsection.

The only relevant input parameter of this stage is the number of initial "gray" classes, say K_g.

5.2.2 Fusion of High Resolution Map with Multispectral Data

Once the elementary fragments are singled out, we enrich their characterization by means of information drawn by the low-resolution multispectral image. This will allow us to obtain a larger and more finely featured set of classes to be used as initial states for the merging process.

Assuming perfect registration, each pixel of the low-resolution MS image can be put in correspondence with a rectangular set of "children" pixels in the PAN image (*e.g.,* a 4×4 square, for the Ikonos and Quickbird images), as shown in Fig. 5.2. We compute the region spectral signature as an average of the spectral responses of all multispectral pixels that overlap the region of interest, with weights proportional to the extent of overlap with the region. The spectral signature μ_k of region R_k is therefore computed as

$$\mu_k = \frac{1}{|R_k|} \sum_{s \in R_k} y^{(\mathrm{MS})}_{\rho(s)}, \tag{5.1}$$

(a) (b) (c)

Figure 5.3: Example of PAN-MS fusion: fragments obtained after the PAN segmentation step (a), corresponding regions of interest in the MS image (b), and featured fragments obtained using the spectral signature of Eq. 5.1.

where $\rho(s)$ is the low-resolution "father" of pixel s, $y_{\rho(s)}^{(\mathrm{MS})}$ its spectral response vector, and $|R_k|$ is the size of region R_k. A graphical example is shown in Fig. 5.3.

Of course, such a straightforward procedure is somewhat arbitrary and will produce some errors that a more sophisticated unmixing procedure [109] might probably avoid. On the other hand, since we characterize regions, rather than pixels, such problems are relatively unimportant. In fact, the MRF-based segmentation produces many large regions for which only a fraction of the interested MS pixels overlap the border, leading to quite reliable spectral signatures. On the contrary, smaller fragments might be inaccurately featured, but they are readily absorbed by larger regions in the merging process, as explained in Sec. 4.2.3, carrying a negligible effect on the final high-level segmentation.

A more serious problem, instead, is the unwanted fusion of same gray-level regions mentioned before. After the PAN-MS fusion, however, such phenomena are easily detected through a threshold test on the region total distortion:[1]

$$D_k = \sum_{s \in R_k} \| \, y_{\rho(s)}^{(\mathrm{MS})} - \mu_k \, \|^2 \, . \tag{5.2}$$

For the mixed regions, a further *local* TS-MRF segmentation is then carried out. This refinement step increases only marginally the overall complexity, because the TS-MRF algorithm works locally on each region. After each con-

[1] The threshold itself is a non-critical parameter as mixed and ordinary regions are form well separated groups.

nected fragment has been associated with a single spectral signature, the processing scale moves once and for all to the region level, making computational complexity all but irrelevant from this point on.

5.2.3 Spectral Clustering

Once obtained the spectral signatures of the regions, we refine the initial segmentation by carrying out a clustering in the spectral domain, so as to separate different semantic classes, with different spectral signatures, pooled together in the first step because of their close gray levels.

We carry out a different clustering on each gray-level class, using always the same number of clusters, K_{sp}, set heuristically in advance as the largest number of semantic classes expected in any gray-level class. Many of such classes are actually uniform, and would not need any further split, but here, as in other steps of the proposed technique, we accept a certain degree of oversegmentation in order to be sure to detect all significant classes in the image. Excessive fragmentation will be eventually made up for in the merging phase, as explained in Sec. 4.2.3.

The clustering algorithm is a weighted version of the K-means, with weights equal to the fragment sizes. By so doing, we minimize the disturbance produced by small fragments, poorly characterized in the spectral domain because of their reduced size, which could lead to inconsistent results.

At the end of this process we obtain the K_c-class color segmentation map, where $K_c = K_g \times K_{sp}$, that will be the starting point for the subsequent spatial-based analysis and hierarchical merging step (the SBC and Merging blocks of Sec. 4.2). Such a map, though not accounting for textural properties, represents by itself a valuable byproduct of the process, that could serve, for example, as a support for a possible region-based adaptive pansharpening.

5.3 Experimental Results

5.3.1 Ikonos Satellite Data

In order to gain better insight on how the proposed technique works and to provide a first evidence of its performance, we present here the results of a segmentation experiment carried out on a two-resolution Ikonos image, a 2km $\times$ 2km section of the city of San Diego (USA), containing both dense and residential urban areas, as well as a significant area covered with vegetation. The 2004×2004 pixel panchromatic image, shown in Fig. 5.4, has a spatial

Figure 5.4: IKONOS imagery used in the experiments: 1m-resolution *panchromatic* image with size 2004×2004.

Figure 5.5: IKONOS imagery used in the experiments: 4m-resolution *blue* channel of the multispectral image with size 501×501.

Figure 5.6: IKONOS imagery used in the experiments: 4m-resolution *green* channel of the multispectral image with size 501×501.

Figure 5.7: IKONOS imagery used in the experiments: 4m-resolution *red* channel of the multispectral image with size 501×501.

Figure 5.8: IKONOS imagery used in the experiments: 4m-resolution *near-infrared* channel of the multispectral image with size 501×501.

Figure 5.9: IKONOS imagery used in the experiments: false color representation of the *multispectral* image (size 2004×2004) using the red, near infrared and blue composite.

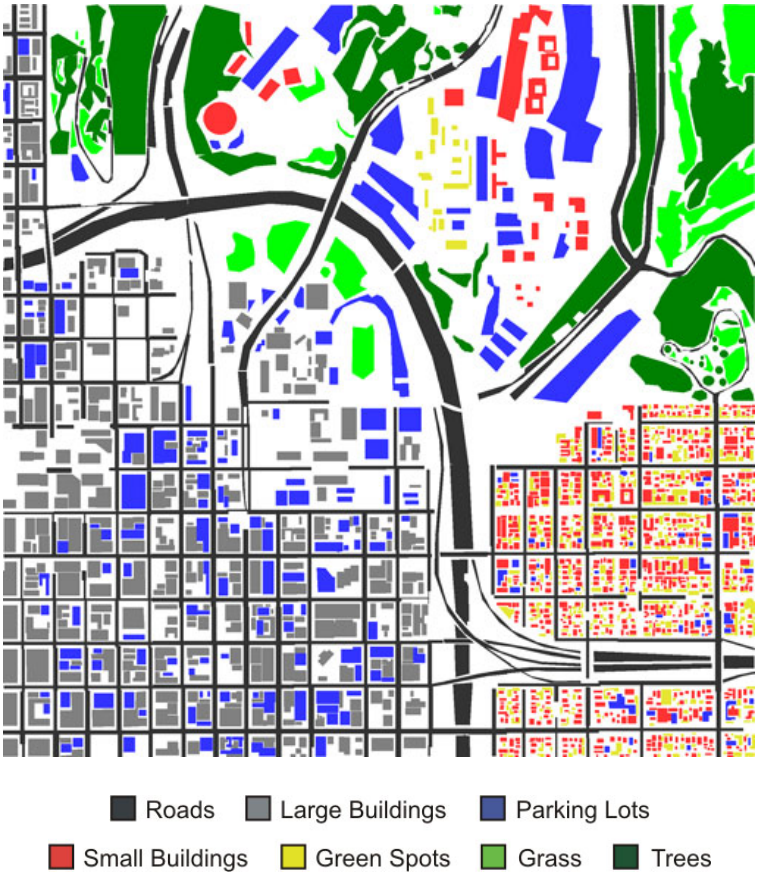


Figure 5.10: IKONOS imagery used in the experiments: manual *ground-truth* with legend.

resolution of 1 meter, while the 501×501 pixel multispectral image, composed of four spectral bands (red, green, blue and near infrared), has a resolution of 4 meters and is perfectly registered with the PAN. In Fig. 5.5 - 5.8 we show each of the four channels of the MS image, while in Fig. 5.9 a false color representation of the MS image is shown, using the red, near infrared and blue bands, that enhances the difference between urban areas and vegetation. The effective radiometric precision is 11 bits per pixel for all components.

Lacking a certified ground truth for performance assessment, we created an *ad hoc* one, reported in Fig. 5.10, by visually inspecting the image, also with the help of the Google Earth maps, and selecting a large number of easily identifiable regions to which we assigned the semantic labels reported at the bottom of Fig. 5.10 for ease of description. Note that, consistent with our multiscale approach, such a 7-class ground truth gives rise automatically, through merging, to other ground truths with fewer classes. As an example, merging the first five classes on one side, and the remaining two on the other side, gives rise to a 2-class urban-areas/vegetation ground truth.

5.3.2 Classification Results

Preliminary Color Segmentation

The only free parameters to set prior of the segmentation procedure are the number of classes used in the TS-MRF segmentation of the PAN image, and in the spectral and spatial clustering phases. After a few preliminary trials, we have selected $K_g = 7$, $K_{sp} = 3$, and $K_s = 5$ respectively; later on we will briefly discuss the robustness of the technique w.r.t. such parameters.

After the segmentation of the PAN image, the PAN-MS fusion, and the subsequent spectral clustering, we obtain a segmentation map composed of many thousands of fragments, grouped in 21 spectral classes. The map is reported in Fig. 5.11 using averaged false colors for each class. This is an intermediate product, to be further processed, nonetheless it deserves some comments. Hence, in Fig. 5.12(a) we show a 270 $\times$ 270 pixel detail of the panchromatic image of Fig. 5.4), together with the corresponding false-color multispectral data (b), and with the (labeled) 21-class segmentation map (c). It is clear that the map catches most if not all image details, retaining a good level of spatial accuracy as testified by the rounded corners of the gardens or the shapes of the trees. To allow an easier interpretation of results, in Fig. 5.12(d) we show again the same map where, however, each class is represented with its average false color. It is apparent that the color segmentation map provides

Figure 5.11: 21-class segmentation map obtained after the spectral clustering. Each color class is represented using its average false color.

a spectral characterization of the image that is completely coherent with the original multispectral component of Fig. 5.9, although it has, in fact, a much higher resolution.

Following the basic TFR data flow, each of these 21 classes is further split into 5 different clusters based on spatial properties. It is worth underlining once more that we will now make a very specific use of these clusters of segments, looking for the detection and recovery of complex textures. If we were interested in reconstructing elementary objects, or solving some other specific problems, *e.g.,* true classification, the first mandatory step would be to disgregate such clusters and handle each fragment by itself, without unnecessary

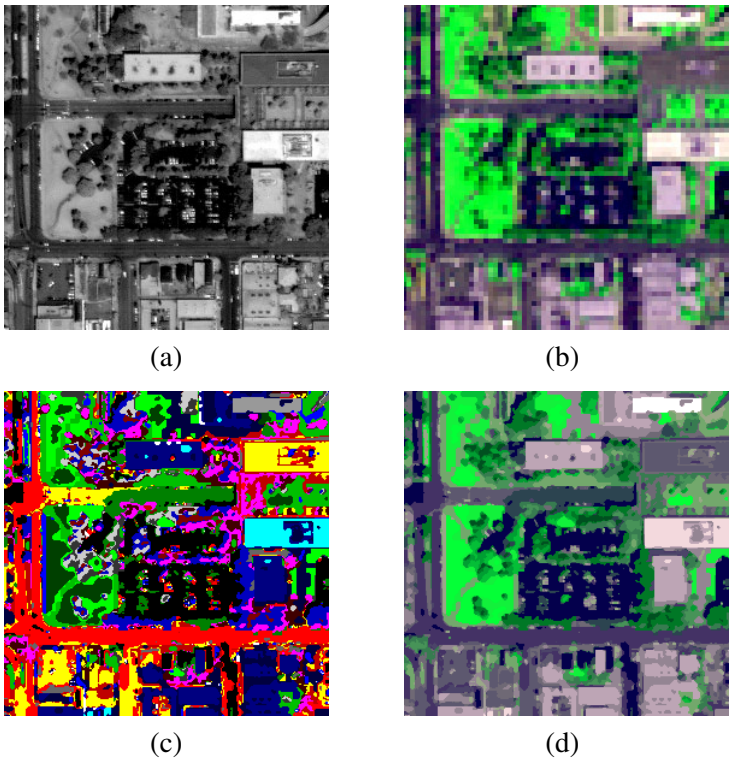


Figure 5.12: A detail of the panchromatic image (a), the corresponding area in the multispectral image (b), the 21-class segmentation map (c), the same map with colors drawn from the MS data (d).

	roads	pk lots	lg blds	sm blds	gr. spots	trees	grass	u. a.
roads	**269471**	113825	21018	14102	2102	1053	89	*63.9%*
park. lots	156841	**109289**	47652	28766	1096	73	3	*31.8%*
large bdg.	27816	43119	**292692**	67945	1432	30	2	*67.6%*
small bdg.	83241	12549	3397	**7287**	3795	1243	0	*6.5%*
green sp.	18862	19616	8134	37622	**40052**	17200	872	*28.1%*
trees	4245	726	232	647	12237	**279043**	37820	*83.3%*
grass	1130	175	57	165	6339	18619	**76387**	*74.2%*
p. a.	*48.0%*	*36.5%*	*78.4%*	*4.7%*	*59.7%*	*87.9%*	*66.3%*	***56.8%***

Table 5.1: Confusion matrix for a 7-class pruning of the segmentation tree. In bold, correct classifications.

constraints bound to undermine the effort. However we do not consider these other applications in this work. The sequential binary merging procedure finally will complete the execution.

Final Multiscale Classification

Once the hierarchical stack is provided, one could browse through the sequence of the corresponding segmentation maps in search of structures of interest that emerge gradually as the result of the merging of neighboring regions.

By selecting a suitable 7-leaf pruning of the tree, and matching the resulting classes with those of the ground truth, we obtain the confusion matrix shown In Table 5.1. The overall accuracy of the technique at this level is fairly good ($\tau = 56.8\%$) considering the total lack of supervision. However, the errors are not evenly distributed among the classes: in particular, the "small building" class singled out in the ground truth does not emerge at all, and its regions are mostly associated with the "large building" class. Likewise, there is a large cross-classification between the "roads" and "parking lots" classes. In both cases, the spatial context has not been strong enough to tell apart such classes, which are very homogeneous spectrally.

The performance improves significantly if we select the 5-leaf pruning shown in Fig. 5.13 since the "roads" are now merged with the "parking lots" while the "small buildings" are merged with the "green spots". This later merging is quite interesting, since it shows that the merging process privileges the emergence of meaningful textures (what we now call "residential" class) rather than the reduction of the classification error. The overall accuracy goes up

	roads	lg blds	sm blds	trees	grass	u. a.
roads	**649426**	68670	46066	1126	92	*84.8%*
large bdg.	70935	**292692**	69377	30	2	*67.6%*
small bdg.	134268	11531	**88756**	18443	872	*34.9%*
trees	4971	232	12884	**279043**	37820	*83.3%*
grass	1305	57	6504	18619	**76387**	*74.2%*
p. a.	*75.4%*	*78.4%*	*39.7%*	*87.9%*	*66.3%*	***73.4%***

Table 5.2: Confusion matrix for the 5-class pruning of Fig. 5.13.

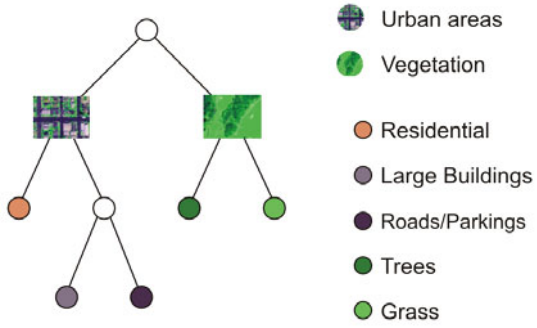


Figure 5.13: Results of the hierarchical segmentation process: a 5-class pruning of the retrieved tree structure.

($\tau = 73.4\%$), and in particular the class accuracies grow to about 80% for the roads and almost 40% for the residential areas (the full confusion matrix is reported in Fig. 5.2).

In the corresponding segmentation map, shown in Fig. 5.14, all major areas of the image are clearly recognizable. Although the wide road network represents by itself an important structure of the image, and its preservation is a success of the algorithm, it also prevents the formation of two distinct urban regions in the downtown and residential areas, which should each include a part of the network.

Going on with the pruning, we obtain eventually the two-class segmentation associated with the top-level nodes, corresponding to the "urban" and "vegetation" macro-textures. To allow for an accurate analysis of this segmentation, in Fig. 5.15 and 5.16 we show a separate image for each class,

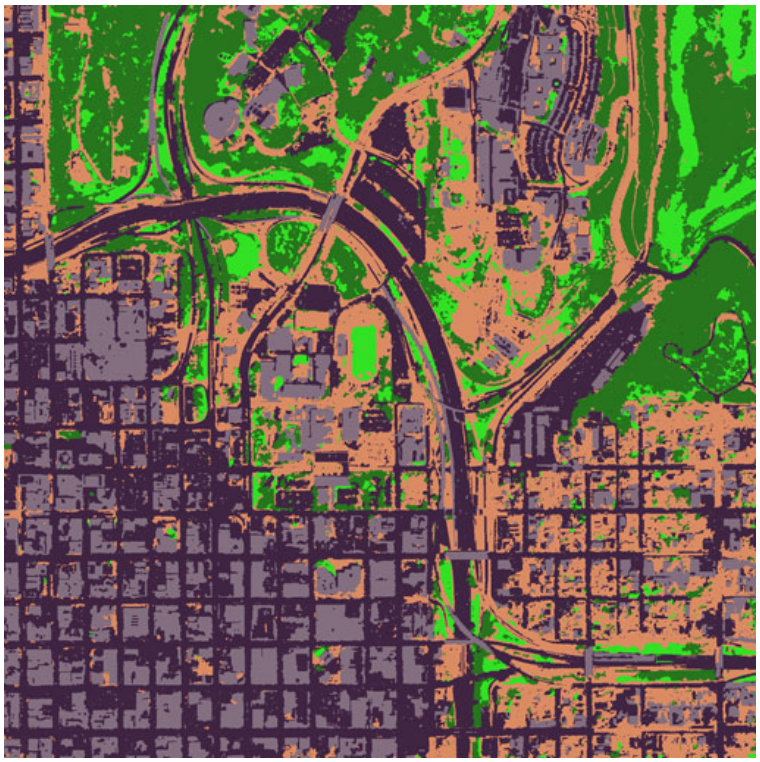


Figure 5.14: The 5-class map corresponding to the tree of Fig.5.13.

Figure 5.15: Top-level segmentation of the test image: *urban areas*. The class of interest is in false colors, the other in black.

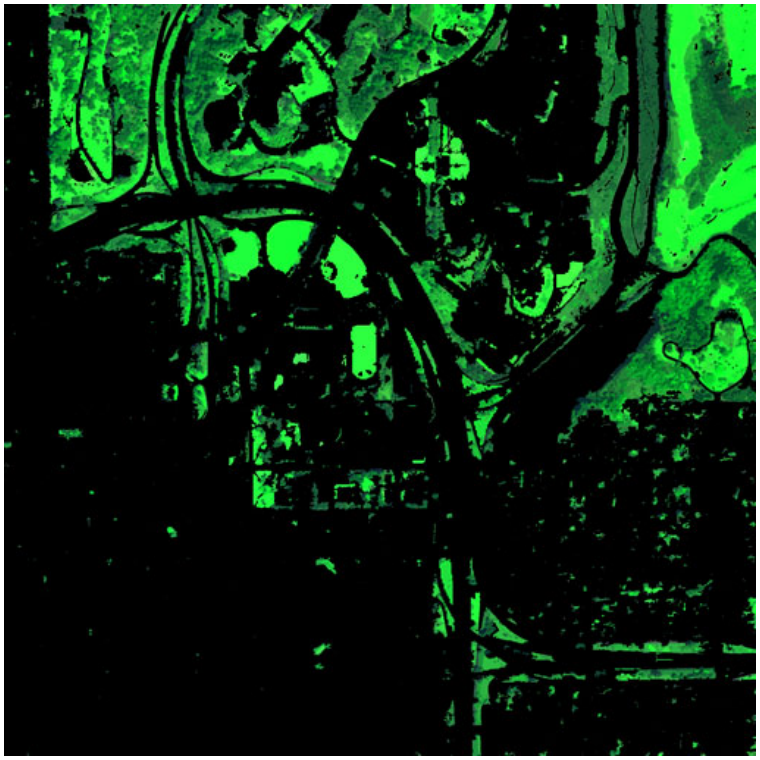


Figure 5.16: Top-level segmentation of the test image: *vegetation*. The class of interest is in false colors, the other in black.

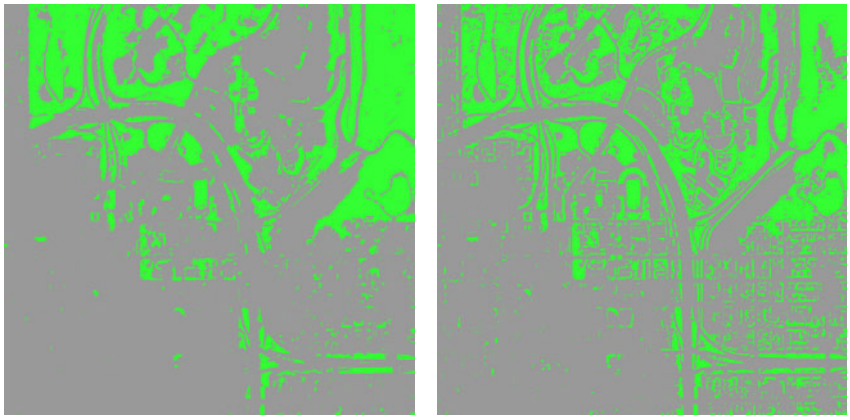
obtained by blackening the other class and showing the high-resolution false-color map for the class of interest. The detection of the two macro-textures is quite accurate ($\tau = 97.5\%$) especially if one considers that some quite complex subtextures of the image, like the residential area in the lower right part, have been uniformly included in the "urban" class, as clear in Fig. 5.15, despite the 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.

Robustness Analysis and Comparisons

As we said before, the performance of the proposed technique is not much sensitive to the exact values of the parameters K_g, K_{sp}, and K_s. To support experimentally such statement, the test image was segmented varying the parameters in the ranges $[5-9]$, $[2-5]$, and $[3-7]$, respectively. As a result, the accuracy with 5 classes varied from a worst case of 65.8% to a maximum of 79.0%, mostly for changes in the residential urban area, remaining quite stable, between 95.5% and 97.8%, for the 2-class case. Note that the maxima were assumed for different combinations of the parameters, none of which corresponds to our compromise choice.

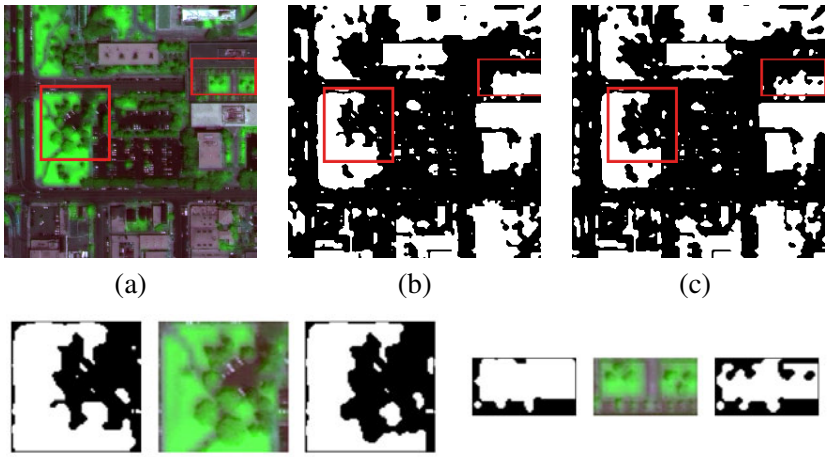
As for the computation time, the experiment described above takes, on the average, 250 seconds on a HP notebook equipped with an Intel Core 2 Duo 1.66 GHz processor. The most time-consuming step, as expected, is the initial TS-MRF segmentation that accounts for about 70% of the whole CPU time, while the post-fusion refinement, takes an additional 10%. Region-level operations have a relatively small cost, less than 20% of the total.

Finally, we compare here the results of the proposed technique with some alternative solutions. This is not an easy task, because the vast majority of techniques proposed in the literature rely on spectral and microtextural properties, and hence they are very good at detecting and possibly classifying objects [91, 100], also using hierarchical multi-scale approaches [18, 2, 110], but fail to detect large-scale complex textures as individual entities, which is the major strength of the proposed technique. As an example, only by means of heavy user interaction the FNEA algorithm [99] could provide products similar to the maps of Fig. 5.14 or Fig. 5.15 and 5.16, but with unsatisfactory results. To gain insight about the different behaviors of spectral-based and texture-based segmentation, let us consider the results obtained by using the TS-MRF, which is a



(a) (b)

Figure 5.17: The 2-class maps obtained using the proposed algorithm (a) and the TS-MRF with supervised split and merge strategy (b).



(a) (b) (c)

(d) (e)

Figure 5.18: Pansharpened detail (a), first binary split obtained by working on pansharpened (b) and on PAN (c) data; enlarged critical areas (d)-(e).

tree-structured spectral-based segmenter, directly on the pansharpened image, obtained using the Gram-Schmidt orthogonalization technique [111]. Pruning the tree at 5 nodes, and matching classes with the ground-truth, the overall accuracy is just 64.7%, with very bad user's and producer's accuracies on the "residential" class, 14.5% and 16.9%, respectively. Then, if we prune back the tree at 2 nodes, the overall accuracy drops down to 54.5% because the first split, on the basis of spectral information only, tells apart just dark and light areas. To obtain a product comparable with our 2-class segmentation, instead, we pooled together optimally (w.r.t. ground truth) some of the classes found at the 5-node level without following the actual tree structure, that is, mimicking the behavior of the merging process proposed here. Even so, the overall accuracy reaches only 91.7% as opposed to the 97.5% of the proposed technique. Looking at the synthetic maps of Fig. 5.17, the main reason for such a disappointing result becomes obvious, since the residential urban area is now split between the "urban" and the "vegetation" classes. Note that pansharpening and TS-MRF segmentation alone take more than 300 seconds of CPU time.

In the end, to test the effectiveness of our choice to work only on the PAN image in the first step, we compare our results with those obtained by growing a large segmentation tree with the pansharpening/TS-MRF approach, and then carrying out the H-MMC based merging process. Even with the best combination of parameters, numerical results are significantly worse than those of the proposed algorithm, with accuracies of 54.0%, 68.9%, and 95.1% for the best 7-, 5- and 2-class pruning, respectively. By looking at the pansharpened version of the same detail considered before, Fig. 5.18, the reason for such impairment is easily understood: some contours are clearly smoothed, leading to the creation of spurious mixed classes that disturb the merging process, and also to some clear errors on boundaries, like those highlighted in the 2-class map of Fig. 5.18(b), which are not present in our segmentation Fig. 5.18(c).

Conclusions

The work of this thesis has concerned the study and development of new hierarchical models and algorithms for image segmentation. In particular, methods for unsupervised color-based and texture-based segmentation have been taken into account, with main application to the domain of remotely sensed images.

The reference models concerning color-based segmentation belong to the family of Tree Structured Markov Random Fields, defined as a hierarchical combination of several reference MRFs, each representing the probabilistic constraints among classes associated with a given node in the region-scale coarse-to-fine hierarchy of the whole TS-MRF. Classes are first associated with the leaves of a tree which must fit the hidden data structure, if any, then each internal node of such a tree is associated with an *ad hoc* MRF model, "local" to that node and involving only its offsprings, which may be real classes or merging of real classes when offsprings are not terminal nodes. The definition of the model is then recursive and, as such, it allows for top-down recursive inference algorithms, where each local MRF is solved once the ancestors fields are solved.

Unsupervised image segmentation based on the TS-MRF model relies heavily on the detection of a tree structure that correctly describes the data structure and on the accurate optimization of MRFs at each node. In the basic segmentation algorithm, a split-by-split growth of the tree is performed, from the root representing the whole image until all the leaves are reached, which is controlled by a test parameter, the split gain, defined locally at each node. In this way the tree structure, and then the number of classes, is automatically detected, while the inference algorithms operate to single out the segmentation by splitting regions recursively.

This segmentation algorithm often proves unsatisfactory both in detecting a suitable tree structure and in performing an accurate MRF optimization, mainly because of some important limitation that are removed in this work. In particular, we allow for the use of generic rather than binary trees, and improve

the MRF initialization at each node, resorting in both cases to the Mean-Shift procedure. In the first case, Mean-Shift allows us to estimate the number of pdf modes at each node, and hence the number of children nodes, while in the latter it is used, together with a Maximum-Likelihood classifier, to replace the much less reliable GLA clustering.

To this end, a fast new Mean-Shift clustering algorithm is proposed, characterized by two main innovative features. First, the selection of the kernel size, that determines the resolution at which modes are detected, is here made adaptive via a k-*Nearest Neighbors* approach, that accounts for wide variations of density in the data space usually happening in the cases of interest. Moreover, a speed-up strategy which reduces the computational burden with little harm for the clustering accuracy has been used to devise the clustering procedure, based on the assumption that points traversed by a kernel function during a single step of the mode detection procedure are likely to belong to the basin of attraction of the final mode detected.

Experiments that prove the effectiveness of the proposed solutions have been carried out both on synthetic images and on remotely sensed ones: land classification experiments in particular gave very promising results, both for unsupervised segmentation and as a tool for the automatic definition of a suitable tree structure in the context of supervised segmentation.

As a second main topic for this work, we treated the problem of texture modeling and texture-based image segmentation, resorting to 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 the unsupervised TS-MRF discussed above, which provides a rough discrete approximation of the original data to be fitted with the texture model at the region level. This 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, that scores segmentation algorithms by means of several accuracy indicators. Moreover, the algorithm was also tested on the natural images of the Berkeley dataset. 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 tex-

ture 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 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.

In the last part of the work, an application of the TFR agorithm to the domain of high-resolution remote sensing images has also been proposed, focusing on multiresolution Ikonos imagery. Given the high resolution of such images, and the consequent presence of complex structures and textured areas, the use of a slightly modified version of the TFR algorithm has been considered, where the initial color map is obtained by means of a sequence of operations using data at different resolution: first, the panchromatic image is segmented by means of the TS-MRF unsupervised algorithm, and then spectral feature are injected at region level from the lower resolution multispectral, to finally perform a color clustering of the image fragments. The choice to use only panchromatic data for the initial segmentation step allows us to better preserve fine details and structures and, together with the use of a tree-structured segmenter, guarantees a reasonable processing time.

Experimental results on a test Ikonos image are encouraging: at a visual inspection, all major regions of interest are clearly recognized, especially at the larger scales, and such a good subjective performance is confirmed by the objective classification accuracy computed w.r.t. an *ad hoc* ground truth.

Bibliography

[1] 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.

[2] L. Bruzzone and L. Carlin. A multilevel context-based system for classification of very high spatial resolution images. *IEEE Transactions on Geoscience and Remote Sensing*, 44(9):2587–2600, September 2006.

[3] A. Tsai, Jr. Yezzi, A., W. Wells, C. Tempany, D. Tucker, A. Fan, W. E. Grimson, and A. Willsky. A shape-based approach to the segmentation of medical imagery using level sets. *IEEE Transactions on Medical Imaging*, 22(2):137–154, Feb. 2003.

[4] Y. Zhang, M. Brady, and S. Smith. Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE Transactions on Medical Imaging*, 20(1):45–57, Jan 2001.

[5] L. Salgado, N. Garcia, J. M. Menendez, and E. Rendon. Efficient image segmentation for region-based motion estimation and compensation. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(7):1029–1039, Oct. 2000.

[6] A. Shamim and J. A. Robinson. Object-based video coding by global-to-local motion segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(12):1106–1116, Dec. 2002.

[7] P. Dillinger, J. F. Vogelbruch, J. Leinen, S. Suslov, R. Patzak, H. Winkler, and K. Schwan. Fpga-based real-time image segmentation for

medical systems and data processing. *IEEE Transactions on Nuclear Science*, 53(4):2097–2101, Aug. 2006.

[8] C. Lindner and F. Puente Leon. Model-based segmentation of surfaces using illumination series. *IEEE Transactions on Instrumentation and Measurement*, 56(4):1340–1346, Aug. 2007.

[9] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Mechine Intelligence*, 22(12):1349–1380, Dec. 2000.

[10] G. Stockman and L.G. Shapiro. *Computer Vision*. Pearson Education, 2001.

[11] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–685, 1989.

[12] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *International Journal of Computer Vision*, 1:321–331, 1988.

[13] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton–jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[14] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, series B*, 48:259–302, 1986.

[15] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1:721–741, Nov. 1984.

[16] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Mechine Intelligence*, 22(8):888–905, Aug. 2000.

[17] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society, series B*, B-36:192–236, February 1974.

[18] 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.

[19] D. Comaniciu and P. Meer. Mean Shift: a robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002.

[20] A. Gersho and R. Gray. *Vector quantization and signal compression*. Kluwer, Boston, MA, 1992.

[21] M. Tuceryan and A. K. Jain. *The Handbook of Pattern Recognition and Computer Vision, 2nd Edition*. River Edge, NJ: World Scientific, 1998.

[22] 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*, 50(1):106–120, January 2003.

[23] A. Clausi and H. Deng. Design-based texture features fusion using gabor filters and co-occurrence probabilities. *IEEE Transactions on Image Processing*, 14(7):925–236, July 2005.

[24] R. M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, May 1979.

[25] B.B. Chaudhuri and N. Sarkar. Texture segmentation using fractal dimension. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 17(1):72–77, January 1995.

[26] Y. Xia, D. Feng, and R. Zhao. Morphology-based multifractal estimation for texture segmentation. *IEEE Transactions on Image Processing*, 15(3):614–623, March 2006.

[27] D. Charalampidis and T. Kasparis. Wavelet-based rotational invariant roughness features for texture classification and segmentation. *IEEE Transactions on Image Processing*, 11(8):825–837, August 2002.

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

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

[30] O. Pichler, A. Teuner, and B. J. Hosticka. An unsupervised texture segmentation algorithm with feature space reduction and knowledge feedback. *IEEE Transactions on Image Processing*, 7(1):53–61, January 1998.

[31] H. C. Hsin. Texture segmentation using modulated wavelet transform. *IEEE Transactions on Image Processing*, 9(7):1299–1302, July 2000.

[32] M. Unser. Texture classification and segmentation using wavelet frames. *IEEE Transactions on Image Processing*, 4(11):1549–1560, November 1995.

[33] 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*, 20(3):252–262, March 1998.

[34] M. Haindl and S. Mikeš. Model-based texture segmentation. *Image Analysis and Recognition, Lecture Notes in Computer Science*, 3212:306–313, 2004. Porto, Portugal.

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

[36] S. Krishnamachari and R. Chellappa. Multiresolution gauss-markov random field models for texture segmentation. *IEEE Transactions on Image Processing*, 6(2):251–267, February 1997.

[37] J. Portillo Garcia, I. Trueba Santander, G. De Miguel Vela, and C. Alberola Lopez. Efficient multispectral texture segmentation using multivariate statistics. *IEE Proceedings on Vision, Image and Signal Processing*, 154(5):357–364, October 1998.

[38] J. Woo and A. C. S. Chung. A segmentation model using compound Markov random field based on a boundary model. *IEEE Trans. on Image Processing*, 16(1):241–252, January 2007.

[39] C. A. Bouman and M. Shapiro. A multiscale random field model for bayesian image segmentation. *IEEE Transactions on Image Processing*, 3(2):162–177, March 1994.

[40] N. Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *International Journal of Computer Vision*, 46(3):223–247, 2002.

[41] M. Rousson, T. Brox, and R. Deriche. Active unsupervised texture segmentation on a diffusion based feature space. In *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages II – 699–704, 2003.

[42] T. Brox and J. Weickert. A tv flow based local scale measure for texture discrimination. *Proc. 8th Eur. Conf. Computer Vision*, 2(2):578–590, May 2004.

[43] T. Brox and J. Weickert. Level set segmentation with multiple regions. *IEEE Transactions on Image Processing*, 15(10):3213 – 3218, 2006.

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

[45] T. Cour, F. Bénézit, and J. Shi. Spectral segmentation with multiscale graph decomposition. *Proc. of IEEE Conference on Computer Vision and Pattern Recognition CVPR 2005*, 2:1124–1131, June 2005.

[46] S. C. Zhu, C. E. Guo, Y. Z. Wang, and Z. J. Xu. What are textons? *International Journal of Computer Vision*, 62(1/2):121–143, 2005.

[47] A. Barbu and S. C. Zhu. Multigrid and multi-level swendsen-wang cuts for hierarchic graph partitions. In *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–731 – II–738, 2004.

[48] 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.

[49] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley, 2000.

[50] H.L. Van Trees. *Detection, Estimation and Modulation Theory*. John Wiley and Sons, 1968.

[51] S. Z. Li. *Markov random field modeling in image analysis*. Springer-Verlag, 1st edition edition, 1995.

[52] G. Winkler. *Image analysis, random fields and dynamic Monte Carlo methods*. Springer-Verlag, 1st edition edition, 1995.

[53] S. Lakshmanan and H. Derin. Simultaneous parameter estimation and segmentation of Gibbs random field using simulated annealing. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 11(8):799–813, August 1989.

[54] M.C. Zhang, R.M. Haralick, and J.B. Campbell. Multispectral image context classification using stochastic relaxation. *IEEE Transaction on Systems, Man and Cybernetics*, 20(1):128–140, Gen.-Feb. 1990.

[55] R. Kinderman and J.L. Snell. *Markov Random Fields and Their Applications*. RI: Amer. Math. Soc., 1980.

[56] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 648–655, 23–25 June 1998.

[57] E. Ising. Beitray sur theorie des ferromagnetismus. *Zeitschrift Physik*, 31:253–258, 1925.

[58] F. Salzenstein and W. Pieczynski. Parameter estimation in hidden fuzzy Markov random fields and image segmentation. *Graphical Models and Image Processing*, 59(4):205–220, 1997.

[59] A. Mohammad-Djafari. Joint estimation of parameters and hyperparameters in a bayesian approach of solving inverse problems. In *IEEE International Conference on Image Processing*, volume 2, pages 473–476, 1996.

[60] G. Poggi and A. R. P. Ragozini. Image segmentation by tree-structured markov random fields. *IEEE Signal Processing Letters*, 6(7):155–157, July 1999.

[61] J.K. Fwu and P.M. Djuric. Unsupervised vector image segmentation by a tree structure ICM algorithm. *IEEE Transaction on Medical Imaging*, 15(6):871–880, December 1996.

[62] J. Zhang, J.W. Modestino, and D.A. Langan. Maximum-likelihood parameter estimation for unsupervised stochastic model-based image segmentation. *IEEE Transaction on Image Processing*, 3(4):404–420, 1994.

[63] Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14:465–478, 1978.

[64] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, January 1975.

[65] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, Aug. 1995.

[66] A. H. Kam and W. J. Fitzgerald. General unsupervised multiscale segmentation of images. In *Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, and Computers*, volume 1, pages 63–67, 24–27 Oct. 1999.

[67] Qiming Luo and T. M. Khoshgoftaar. Unsupervised multiscale color image segmentation based on mdl principle. *IEEE Transactions on Image Processing*, 15(9):2755–2761, Sept. 2006.

[68] M. P. Wand and M. Jones. *Kernel Smoothing*. Chapman and Hall, 1995.

[69] R. Gaetano, G. Poggi, and G. Scarpa. Identification of image structure by the Mean Shift procedure for hierarchical MRF-based image segmentation. In *Proc. EUSIPCO 2006*, Florence, Italy, Sept. 2006.

[70] V. Epanechnikov. Nonparametric estimates of a multivariate probability density. *Theory of Probability and its Applications*, 14:153–158, 1969.

[71] D. W. Scott. *Multivariate Density Estimation*. Wiley, 1992.

[72] R.G. Congalton. A review of assessing the accuracy of classifications of remotely sensed data. *Remote Sensing of Environment*, 37(1):95–96, 1991.

[73] 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 Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[74] G. Scarpa, G. Poggi, and J. Zerubia. A binary tree-structured mrf model for multispectral satellite image segmentation. Theme 3 5062, INRIA Sophia Antipolis, project ARIANA, December 2003.

[75] G. Scarpa and M. Haindl. Unsupervised texture segmentation by spectral-spatial-independent clustering. In *18th International Conference on Pattern Recognition, 2006*, volume 2, pages 151–154, August 2006.

[76] W. D. Penny. Kullback leibler divergences for normal, gamma, dirichelet and wishart densities, technical report. Technical report, Wellcome Dept. of Imaging Neuroscience, University College Longon, 2001.

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

[78] 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.

[79] J. Munkres. Algorithms for assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, March 1957.

[80] 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.

[81] 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 Con-*

ference on Computer Vision, pages 675–682, Bombay, India, January 1998.

[82] 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.

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

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

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

[86] A.K. Shackelford and C.H. Davis. A combined fuzzy pixel-based and object-based approach for classification of high-resolution multispectral data over urban areas. *IEEE Transactions on Geoscience and Remote Sensing*, 41(10):2354–2363, October 2003.

[87] P.C. Smits and A. Annoni. Updating land-cover maps by using texture information from very high-resolution space-borne imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 37(3):1244–1254, May 1999.

[88] Wanxiao Sun, V. Heidt, Peng Gong, and Gang Xu. Information fusion for rural land-use classification with high-resolution satellite imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 41(4):883–890, April 2003.

[89] I. Epifanio and P. Soille. Morphological texture features for unsupervised and supervised segmentations of natural landscapes. *IEEE Transactions on Geoscience and Remote Sensing*, 45(4):1074–1083, April 2007.

[90] G. Perrin, X. Descombes, and J. Zerubia. A marked point process model for tree crown extraction in plantations. In *ICIP 2005. IEEE International Conference on Image Processing, 2005.*, volume 1, pages I–661–4, 11-14 September 2005.

[91] S. Aksoy and E. Dogrusoz. Modeling urban structures using graph-based spatial patterns. In *International Geoscience and Remote Sensing Symposium, IGARSS 2007*, July 2007.

[92] A.A. Farag, R.M. Mohamed, and A. El-Baz. A unified framework for map estimation in remote sensing image segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 43(7):1617–1634, July 2005.

[93] A. Baraldi, V. Puzzolo, P. Blonda, L. Bruzzone, and C. Tarantino. Automatic spectral rule-based preliminary mapping of calibrated Landsat TM and ETM+ images. *IEEE Transactions on Geoscience and Remote Sensing*, 44(9):2563–2586, September 2006.

[94] Z. Kato, J. Zerubia, and M. Berthod. Unsupervised parallel image classification using a hierarchical markovian model. In *5th International Conference on Computer Vision, ICCV'05*, pages 169–174, June 1995.

[95] J. M. Laferté, P. Pérez, and F. Heitz. Discrete markov image modeling and inference on the quadtree. *IEEE Transaction on Image Processing*, 9(8):390–404, March 2000.

[96] A. Katartzis, I. Vanhamel, and H. Sahli. 'a hierarchical markovian model for multiscale region-based classification of vector-valued images. *IEEE Transaction on Geoscience and Remote Sensing*, 43(3):548–558, March 2005.

[97] Zhijun Wang, D. Ziou, C. Armenakis, D. Li, and Qingquan Li. A comparative analysis of image fusion methods. *IEEE Transactions on Geoscience and Remote Sensing*, 43(6):1391–1402, June 2005.

[98] B. Aiazzi, R. Baronti, and M. Selva. Improving component substitution pansharpening through multivariate regression of MS+Pan data. *IEEE Transactions on Geoscience and Remote Sensing*, 45(10):3230–3239, October 2007.

[99] M. Baatz and A. Schäpe. Multiresolution segmentation: an optimization approach for high quality multiscale image segmentation. *Angewandte Geogr. Informationsverarbeitung*, 12:12–23, 2000.

[100] P. Gamba, F. Dell'Acqua, G. Lisini, and G. Trianni. Improved VHR urban area mapping exploiting object boundaries. *IEEE Transaction on Geoscience and Remote Sensing*, 45(8):2676–2682, August 2007.

[101] J. Bosworth, T. Koshimizu, and S.T. Acton. Multi-resolution segmentation of soil moisture imagery by watershed pyramids with region merging. *International Journal of Remote Sensing*, 24(4):741–760, 2003.

[102] F. Bovolo and L. Bruzzone. A detail-preserving scale-driven approach to change detection in multitemporal sar images. *IEEE Transaction on Geoscience and Remote Sensing*, 43(12):2963–2972, December 2005.

[103] C. Unsalan and K.L. Boyer. Classifying land development in high - resolution satellite imagery using hybrid structural - multispectral features. *IEEE Transactions on Geoscience and Remote Sensing*, 42(12):2840–2850, December 2004.

[104] X. Huang, L. Zhang, and P. Li. Classification and extraction of spatial features in urban areas using high-resolution multispectral imagery. *IEEE Geoscience and Remote Sensing Letters*, 4(2):260–264, April 2007.

[105] S.L. Horowitz and T. Pavlidis. Picture segmentation by a tree traversal algorithm. *Journal of the Association for Computing Machinery*, 23(2):368–388, April 1976.

[106] J.-M. Beaulieu and M. Goldberg. Hierarchy in picture segmentation: A stepwise optimization approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(2):150–163, February 1989.

[107] J. Le Moigne and J.C. Tilton. Refining image segmentation by integration of edge and region data. *IEEE Transaction on Geoscience and Remote Sensing*, 33(3):605–615, May 1995.

[108] H.G. Akçay and S. Aksoy. Morphological segmentation of urban structures. In *4th IEEE GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas*, Paris (France), April 11-13 2007.

[109] N. Keshava and J. Mustard. A survey of spectral unmixing algorithms. *Lincoln Laboratory Journal*, 14(1):55–78, 2003.

[110] J.C. Tilton. Analysis of hierarchically related image segmentations. In *2003 IEEE Workshop on Advances in Techniques for Analysis of Remotely Sensed Data*, pages 60–69, October 2003.

[111] C. A. Laben and B. V. Brower. Process for enhancing the spatial resolution of multispectral imagery using pan-sharpening. U.S. Patent Tech. Rep. 6 011 875, Eastman Kodak Company, January 2000.