# SEGMENTATION OF URBAN AREAS IN SPOT IMAGES USING MRF

*F. Richard* *
INRIA
BP 93
06902 Sophia Antipolis - FRANCE
richard@math-info.univ-paris5.fr

*F. Falzon*
ALCATEL ALSTHOM Recherche
Route de Nozay
91460 Marcoussis - FRANCE
falzon@aar.alcatel-alsthom.fr

*J. Zerubia, G. Giraudon*
INRIA
BP 93
06902 Sophia Antipolis - FRANCE
zerubia@sophia.inria.fr

## ABSTRACT

Classification of urban areas from a density measure is required in many application fields. For instance, propagation models for cellular radio networks are parameterized by coefficients whose value is linked to the nature of the soil. Geomarketing is also interested in knowing the population concentration (urban, suburban ...). This paper presents the first results about the construction of an algorithm devoted to the classification of urban areas in SPOT panchromatic images. This algorithm has been inspired by [5, 3] and we point out the differences betwen their method and ours.

## 1 INTRODUCTION

This paper deals with segmentation of urban areas [6] in SPOT panchromatic images. It starts from the MRF model proposed in [5]. Investigation of this model led us to undertake the extension of an initialization phase, and we point out the difficulties that appear when labelling differently two regions with same textural properties.

Consequently, we modify the labelling procedure proposed in [5] and, more generally, we improve the labelling coherence. These modifications are included into a new segmentation/classification algorithm presented herein that also improves the speed of successive scans by introducing information on spatial context.

Our ultimate objective is to obtain an accurate classification of urban areas based on concentration (or density) criteria. This latter point, however, will not be explicitely developped herein, but it is implicitely embedded in the new segmentation schemes.

The first section is related to the model proposed in [5]. The second section is devoted to the new classification algorithm we have proposed, and results are discussed in the last section.

*F. Richard is now with the Projet PRISME, Univ. René Descartes, UFR maths. info., 45 rue des saints pères, 75270 Paris cedex 6.

## 2 MODELLING

The observed image $I$ is defined on a regular lattice $S$. In a segmentation process, each site in $S$ has to be labellised according to the region it belongs to.

Let $E$ denote the image resulting from pixels' labelling, $\Lambda$ the set of all possible labels, $\Omega_I$ the configurations' space of the observed image and $\Omega_E$ the configurations' space of labels' image.

Segmentation is tackled as an energy minimisation problem. A mapping

$$U : \Omega_I \times \Omega_E \longrightarrow \mathbb{R}$$

is designed to express the adequation of the labels configuration with the observed image.

For a given observed image $i$, the solutions are the configurations $\hat{e}$ which minimise $U(., i)$.

In a stochastic setting, we assume that $I$ and $E$ are two random fields. As it is proposed in [3], we opt for a Gibbs representation of the distributions and define an a posteriori distribution

$$P(E \mid I = i) = \frac{1}{Z(i)} \exp\ (-U(E, i)) . \qquad (1)$$

From then on, it is equivalent to seek either for energy minimum or for Maximum A Posteriori (MAP).

To design the energy, we choose a neighborhood system $G = \{G_s, s \in S\}$ where $G_s$ is a neighbors' set associated to the pixel $s$. $G$ determines a set of cliques (a clique is a set for which two elements are neighbors, or a singleton).

We then define $U$ as a sum of potentials whose values depend exclusively on cliques' configurations. As a consequence, the Gibbs random field $E$ is defined with respect to $G$. This implies that $E$ is a Markov random field with respect to $G$.

$$U(e, i) = \sum_{s \in S_E} \sum_{\lambda \in \Lambda} \delta(e_s = \lambda)\, \Phi_s(\lambda, i) + R_s(e) \qquad (2)$$

$\delta(e_s = \lambda) = 1$ , if $e_s = \lambda$, 0 otherwise. $\Phi_s$ designates a disparity measure between a block of pixels centered at

$s$ and all the pixels labelled $\lambda$. $R_s(e)$ is a regularity term (Potts model) that we did not use in our experiments. Note that the cliques are limited to the singletons. This is an important difference with the approach in [3] where neighborhood systems are fully used.

**Segmentation process :** The segmentation process is based on textures' features. We derive from $I$ a vector image $O$: $O_s$ is a vector including local features computed from $I$ within a window centered at $s$.

In our experiments, we employed a quantised version of Haralick textures attributes [4] and $\Phi$ is as follows:

$$\Phi(e_s, \lambda) = \sum_{m=1}^{M} \Big( 2\,\delta(d(F_s^m, \tilde{F}^m(\lambda)) > c_m) - 1 \Big) \qquad (3)$$

where M is the number of features. $F_s^m$ is the distribution of the $m^{\text{th}}$ feature on a block centered at $s$. $\tilde{F}^m(\lambda)$ is the distribution of the $m^{\text{th}}$ feature on the whole sites labelled $\lambda$. $d$ is the Kolmogorov-Smirnov distance [3, 1] between distributions. $c_m$ is a threshold. This threshold can be fixed using values from the statistical test of Kolmogorov-Smirnov. It explains why we use the Kolmogorov-Smirnov distance, but we could have taken any distance between distributions.

Let us note that a "sites relation" is introduced via the definition of $\Phi_s$. This relation is global. In [3], because the relation is defined exclusively via the neighborhood system, the relation is local.

## 3 IMPLEMENTATION

We present herein two algorithms that we used to compute the MAP and we point out the differences with the algorithm proposed in [5].

Both algorithms are divided in $K$ phases. The $k^{\text{th}}$ phase consists in a sequence of steps. At the $n^{\text{th}}$ step, labels are in a set $\Lambda(n)$, the set of current labels.

A ***special*** label $\lambda(r)$ is also introduced. This label controls label creation. Indeed, at the begining of a step, the ***special*** label of the previous step becomes a new current label. Besides, steps are stopped when the special label is not used. Thanks to this label, the number of labels needed for the description of the image needs not to be fixed a priori.

**Algorithm :** Within a step, the lattice is scanned several times. During a scan, sites are labeled in accordance with the following rules : Let us call $d(F_s^m, \tilde{F}^m(\lambda))$ the response of the $m^{\text{th}}$ feature to the label $\lambda$ at site $s$. This response is negative if the distance is below the threshold, positive otherwise.

The rule for algorithm $A$ is as follows: $\lambda(r)$ is assigned to site $s$, if for all $\lambda$, less than $k$ features are giving a positive response; otherwise, site $s$ takes the current label for which the number of positive responses is the highest.

The rule for algorithm $B$ is almost the same except that a site label is modified if and only if the site is located on an edge, and that possible labels for a site are the current labels of its neighbors and its own label.

We presented the algorithm in terms of rules to give a better understanding of its behavior. Remark however that it can be seen as an ICM applied to conditional probabilities that are defined with energies similar to Eq. 2.

Both rules imply that, at the $k^{\text{th}}$ phase, pixels are grouped in accordance to $k$ features. Thus, as $k$ increases, the criterium for a site to keep or to get a current label becomes finer and finer. In the first phase, sites are clustered following a coarse criterium. At each phase, the labels configuration of the previous phase is reorganied according to additional requirements.

**Remarks :** Actually, the decomposition of the algorithm into phases extends the idea of an initialisation phase presented in [5]. As a consequence of this decomposition, the progression of the process is more steady. We can also have a complete control of the evolution.

Let us mention another remark about the role of phases. Sites are grouped in accordance with their responses to a certain number of features, whatever those features are. Thus, provided that phases are stopped for small $k$, features which are "not good descriptors" of the image do not alter the behavior of the algorithm. This shows that the algorithm is able to adapt to a set of images for which descriptions are accomplished differently. From a practical point of vue, this property is interesting when it is hard to define a unique feature which describes a set of studied images.

**Differences between the two algorithms :** There are two main differences between algorithms $A$ & $B$ and the algorithm described in [5]. First of all, at the beginning of a step in [5], labels are assigned to each region having the ***special*** label. Therefore, different labels can be assigned to sites having same textural properties. Consider the case when two regions with two different labels but with same textural attributes meet. This configuration is a local minimum of the energy. On the frontier, the label of sites are indifferently one of the two regions labels and oscillates between them. The ICM can thus be slowed or stopped in that type of configuration.

The relabelling procedure we choosed for algorithms $A$ and $B$ is a way to avoid this problem. At the end of each step, the ***special*** label is changed into **one** current label. This procedure improves both energy landscape and ICM convergence. It also allows an interpretation of the labels. Once the algorithm has converged, we are insured that the correspondence between the current labels and the textures in the image is one-to-one. Moreover, in [5], sites that can change their label are

determined by an instability heap. This heap is equivalent to sites' selection in rule $B$, in the sense that it restricts the number of modified sites and thus speeds up scans. However, it seems possible that, using such selections, a site can have the *special* label at the end of an ICM, even if a current label suits it. This can lead to the problem we just mentionned. That is why we have added a scan at the end of each step in $B$ so that all sites could be checked and properly labelled with one of all possible labels[1].

One can remark that algorithm $A$ does not take spatial information into account to make decision about pixel labels. This is essentially due to the form of the energy (Eq. 2) which is restricted to singleton cliques. This weakness is bypassed in $B$ and [5] by choosing possible labels among the current labels of neighboring sites.

Finally, we would like to point out the fact that, due to the modifications explained above, the labels correspond exactly to clusters of pixels belonging to the same class. Therefore algorithm B can be used for classification.

## 4 RESULTS

Algorithm $B$ was performed on SPOT images of Cherbourg and Paris. Those images present textures that are quite different. In all experiments, we used Haralick features. They were computed on $(6 \times 6)$ local windows; blocks' size was $(15 \times 15)$. For all images, the thresholds applied to the distance between distributions of the features were fixed to 0.34. This threshold tune the confidence on the features distribution and this value has been found experimentaly and corresponds to a minimum amount of misclassified pixels

As one can see in Fig. 1(a), 1(c), 1(e), phase 1 permits to delimit urban regions from non urban regions. Segmentation within urban regions can be observed at phase 2 in 1(b), 1(d), 1(f). In next phases, images can be over-segmented. We thus stopped the algorithm after phase 2.

## 5 CONCLUSION

We have described a new algorithm for the classification of satellite images in terms of types of urban areas. This method is based on MRF and is a modification of the one proposed in [5]. Indeed, we have changed the rules for the attribution of the labels. Moreover, our algorithm is decomposed in successive phases that allows us to control the evolution of the migration/creation of labels. Our first results are promising and now we aim at reducing both the amount of memory used and the execution time for application of this method to images of larger size.

[1] For initialisation, we have also added a scan of this kind at the beginning of the steps.

## References


[1] M. Basseville *Distance Measures for Signal Processing and Pattern Recognition* Signal Processing, vol. 18, n. 4, pp. 349-369, December 1989.

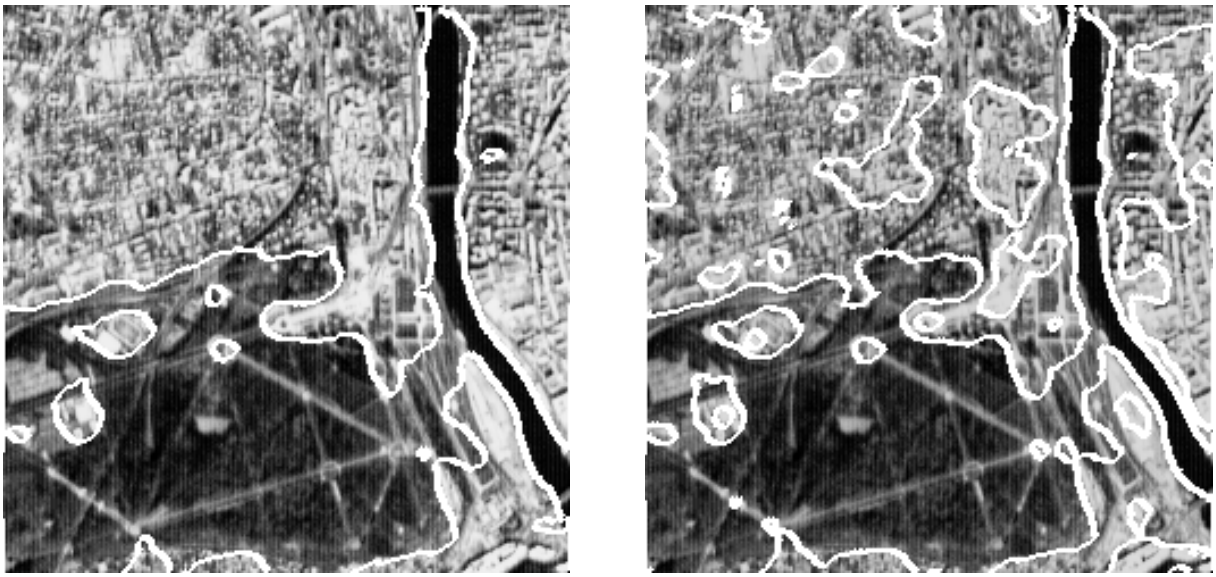
[2] D. Geman & S. Geman *Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images.* IEEE Trans. Patt. on Anal. Mach. Intel., vol. 6, n. 6, November 1984.

[3] D. Geman & S. Geman & C. Graffigne & P. Dong *Boundary Detection by Constrained Optimization*, IEEE Trans. Patt. on Anal. Mach. Intel., vol. 12, n. 7, pp. 609-628, 1990.

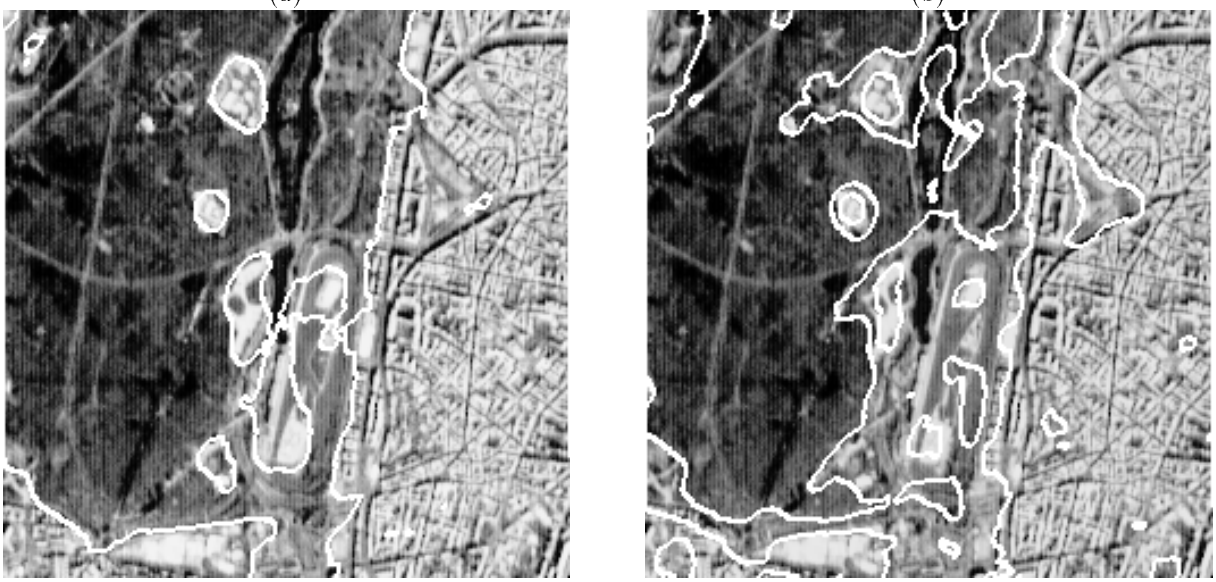
[4] R.M. Haralick *Statistical and Structural Approches to Texture.* IEEE Trans. Patt. on Anal. Mach. Intel. vol. 67, n. 5, May 1979.

[5] C. Kervrann & F. Heitz *A Markov Random Field Model-based Approach to Unsupervised Texture Segmentation Using Local and Global Spatial Statistics.*, IEEE Trans. on Image Proc., vol. 4 no. 6, June 1995.

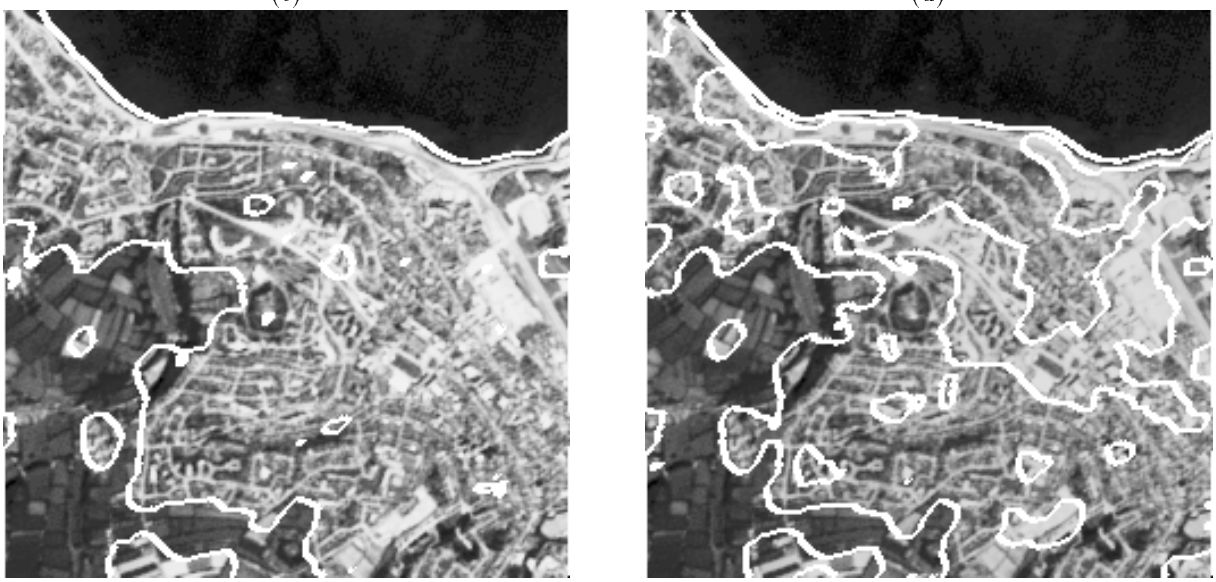
[6] C. Weber *Images Satellitaires et Milieu Urbain* Collection Géomatique, Hermes ed., 1995.



(a) (b)



(c) (d)



(e) (f)

Figure 1: Segmentation of SPOT images using algorithm $B$: $Paris_1$: (a) phase 1, (b) phase 2; $Paris_2$: (c) phase 1, (d) phase 2; $Cherbourg_1$: (e) phase 1, (f) phase 2.