16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, August 25-29, 2008, copyright by EURASIP

TEXTURE IMAGE SEGMENTATION BY HIERARCHICAL MODELING

Giuseppe Scarpa, Raffaele Gaetano, and Giovanni Poggi

Dipartimento di Ingegneria Elettronica e delle Telecomunicazioni,
Università "Federico II" di Napoli, Naples, Italy
phone/fax: + (39) 0817683151/49, email: *firstname.lastname*@unina.it

ABSTRACT

The *Texture Fragmentation and Reconstruction* (TFR) algorithm has been recently proposed for the unsupervised hierarchical segmentation of textures. It is based on a hierarchical image model, where textures are characterized in terms of their spatial interaction properties, modeled by means of a set of Markov chains, each one associated with a major spatial direction. The TFR algorithm fits the image to the hierarchical model by means of a split-and-merge procedure where the first step (*fragmentation*) aims at extracting the elementary texture states, which are progressively merged in the second step (*reconstruction*), so as to obtain a hierarchical nested segmentation.

Although TFR results are usually very good, it has been sometimes observed a bias towards the undersegmentation for complex images. Here, we analyze this phenomenon and propose the use of an improved fragmentation step, where would-be elementary states are ranked based on a suitable measure of their reliability and possibly purged. Experimental results validate the effectiveness of the new algorithm.

1. INTRODUCTION

Image segmentation has been intensely studied in recent decades because of its importance in such diverse fields as medical imaging, security, remote sensing, industrial automation, etc. Even so, in many cases it still remains an open problem, as happens with textured images where the spatial interactions may cover long ranges, asking for high-order complex modeling. Due to its complexity, the texture segmentation problem is usually split in two tasks, texture feature extraction and clustering, the first of which, especially relevant, has received considerable attention in the past. Tuceryan and Jain [1] divide feature extraction methods into four categories: statistical, model-based, signal processing, and geometrical. Typical tools used in the various categories are, respectively, co-occurrence matrices [2], Markov random field [3] and auto-regressive models [4], wavelet [5] or Gabor [6] filtering, and fractal dimension [7].

It is widely recognized that a visual texture, which humans can easily perceive, is very difficult to spot automatically. The main problem is that texture definition itself is quite debated, depending often on the application or on different perceptual motivations, and there is no general agreement on it. Even more challenging, there are many instances where the textural properties may completely change depending on the scale of observation, in which case *unsupervised* segmentation becomes an ill-posed problem unless additional indications come from the application domain. When no such information is available, the most reasonable output is a *hierarchical* segmentation, that is, a stack of nested segmentations, leaving to the user the freedom to choose the one that better fits the application requirements.

In this work, texture segmentation is approached through the recently proposed [8] Hierarchical Multiple Markov Chains (H-MMC) model. In H-MMC the image is seen as a complex collection of textures, emerging at different scales of observation. At each scale of the hierarchy, spatial interactions among textures are described through a set of Markov Chains, and then taken into account to decide which textures should be merged at the next higher level. In [8] a segmentation algorithm was also proposed, named *Texture Fragmentation and Reconstruction* (TFR), which builds upon the hierarchical image representation provided by the H-MMC model. It first singles out the homogeneous elementary components of the textures and then proceeds to merge them until all textures of interest emerge, possibly at multiple scales. The TFR presents a number of interesting properties: it is able to recognize macro-texture at various scales of observation, is completely unsupervised, and hence applicable in many different domains, and its computational complexity is quite limited. Moreover, its accuracy appears to be superior to most texture segmentation techniques [9].

Among its drawbacks, one of the most disturbing is certainly a bias towards the under-segmentation of images, especially in the presence of very complex textures. In this work we address this problem, which appears to be mainly related to the presence of elementary texture components that are not really homogeneous, and propose an improved version of the algorithm in which such components are analyzed, ranked on the basis of their homogeneity, and possibly further fragmented, before the merging process begins.

The main ideas of the H-MMC model and of the TFR segmentation algorithm are presented in Section 2, Section 3 analyzes in more detail the undersegmentation problem and the proposed solution, and finally, Section 4 presents experimental results and draws conclusions.

2. HIERARCHICAL TEXTURE MODEL AND SEGMENTATION STRATEGY

We will try to convey the main concepts of interest through a running example, the segmentation of the simple image shown in Fig.1(a), a urban residential area, with a rectangular road network and regular blocks comprising both buildings and green spots. Our first processing step is a "conventional" segmentation [10] based only on spectral information. Once applied to our image, the segmenter outputs the three-class map shown in Fig.1(b), a good description of the image in terms of its local properties but certainly unable to catch its dominant structures, the blocks and the road networks. Such structures emerge, however, if the green and light-gray classes are merged, as shown in Fig.1(c). These two alterna-

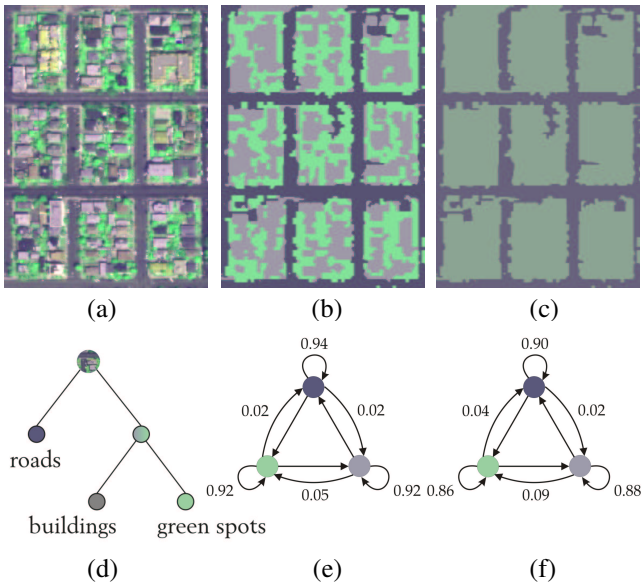


Figure 1: HMMC model: *urban area* sample (a); *3-state* (b) and *2-state* (c) maps; states hierarchy (d); *3-state* Markov chains for north (e) and south-east (f) directions.

tive segmentations of the image can be associated with different subtrees of a tree of classes as shown in Fig.1(d): by using all the leaves of the tree, namely, all the original classes, we obtain the original map of Fig.1(b), while pruning the two rightmost branches produces the map of Fig.1(c). The whole tree is therefore associated with a hierarchy of segmentation maps, or image descriptions, among which, one can select the most appropriate for the application needs.

Such a hierarchical description of the image allows one to extract quite different, and valuable, pieces of information, by just looking at different scales of segmentation. Of course, an automatic tool is necessary to decide which classes should be merged each time or, in other words, which classes can be regarded as different components of the same textured region. Such decisions depend only on spatial properties, and hence the key for texture description in this framework will be the analysis of the spatial interactions among the different classes.

To this end, let us scan the map of Fig.1(b) pixel-wise along a given direction, and take note of the classes encountered along the path. In a probabilistic setting, the resulting sequence can be interpreted as a realization of a Markov chain, where the classes take on the role of states, and the transition probabilities are computed as relative frequencies of transition among classes. By repeating this modeling step for a set of significant directions, (for example, north, north-east, etc.), we obtain an accurate description of inter-class spatial dependencies.

More formally, let Ω be the label set for the segmentation map, then, to each spatial direction, $j = 1, \ldots, 8$, we associate a Markov chain described by its transition probability (TP) matrix $\mathbf{P}_j = \{p_j(\omega'|\omega)\}$ where:

$$p_j(\omega'|\omega) \triangleq \Pr(x_{(s+1)_j} = \omega' \mid x_s = \omega) \qquad \forall \omega, \omega' \in \Omega \quad (1)$$

where $x_s \in \Omega$ is the state of pixel s of the map, $(s+1)_j$ indicates the pixel following s along the direction j, and the probabilities are estimated as frequency ratios on the map.

In the example of Fig.1, we readily compute by (1) the 8 three-state TP matrices for the map of Fig.1(b), thus obtaining a set of Markov chains, two of which, corresponding to the north and south-east directions respectively, are shown in Fig.1(e)-(f). Such a representation provides valuable information about the spatial distribution of elements in this textured image. Intrastate TPs, for example, give indications about the average shape and orientation of the objects of a given class, as for the "roads" state, where the TPs suggest that its elements are characterized by a dominant vertical direction. Interstate TPs, instead, describe the frequency of transitions between different states along the selected directions, and hence give information on their spatial contiguity. For example, the relatively large TPs between the "building" and "green spots" states (0.05 and 0.09, Fig.1(e)-(f)) show that a tight interaction exists between these states, which could be possibly regarded as different elements of a single texture.

The analysis of all the TP matrices allows one to build a tree like that of Fig.1(d) (we use only binary trees for the sake of simplicity), by recursively merging couples of terminal states[1] until all nodes collapse in the tree root. At each step, we should single out and merge the classes that exhibit the strongest mutual interaction, since they are the most likely to belong to the same textured area. To this end, for each terminal class ω we define a synthetic parameter called Texture Score.

$$\mathrm{TS}^{\omega} = \frac{p(\omega)}{\max_{\omega' \neq \omega} p(\omega'|\omega)}, \quad (2)$$

which 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

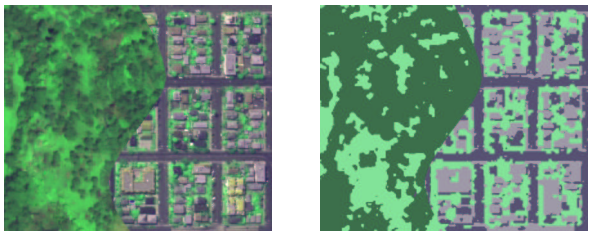
$$\mathrm{TS}^{\omega} = p(\omega) \cdot \frac{1}{p(\bar{\omega}|\omega)} \cdot \frac{p(\bar{\omega}|\omega)}{\max_{\omega' \neq \omega} p(\omega'|\omega)}, \quad (3)$$

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 ($\max_{\omega' \neq \omega} p(\omega'|\omega) \simeq p(\bar{\omega}|\omega)$). Hence, they are 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 (although it can always attract other classes).

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

$$\omega^* = \arg\max_{\omega \neq \widehat{\omega}} p(\omega|\widehat{\omega}) \quad (4)$$

[1] The same entity can be regarded as an image region, a segmentation class, a state of a Markov chain, or a tree node, depending on the context, and then, from now on, we will use such terms interchangeably.



(a) (b)

Figure 2: A complex image (a) composed of two real textured patches (forest and urban area), and its four-class segmentation (b).

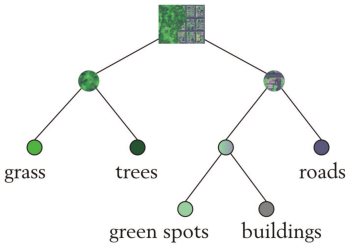


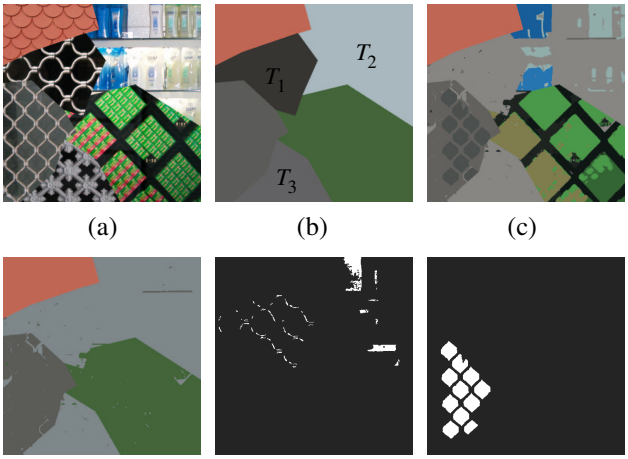
Figure 3: A tree-structured description for image 2(a).

TP 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.

3. SELECTION OF RELIABLE INITIAL STATES

In last Section, we described the TFR algorithm as if the "colors" found in the initial segmentation were the elementary classes from which the merging process could start. However, this is usually not the case as we will show by considering the image of Fig.2(a) which is the synthetic composition of two different textured environments, "forest" and "urban". A human interpreter would easily provide a description of the image by the hierarchical structure of Fig.3, where the two main environments correspond to the left and right subtrees departing from the root, and each subtree comprises some component classes of its own. However, if we apply a color quantization algorithm to this image, we end up with the 4-class map shown in Fig.2(b), which highlights an important problem: the light-green class of the map corresponds to two different semantic classes, belonging to two different textures, namely, the "grass" class for the forest and the "green spots" class for the urban area; something similar happens for the "dark grey" class too, which comprises the road network along with some darker buildings within the blocks. The segmentation algorithm, working only on spectral features, has no way to tell apart these two classes: nonetheless they should be distinct from the beginning in order to recover two different high-level textures after recursive merging.

The use of all connected fragments of the image as initial classes might then look as a reasonable alternative, but this must be ruled out, not so much for the significant increase in complexity, as for the fact that such isolated fragments would lack any spatial structure, and could not become reliable seeds of larger textures.



(a) (b) (c)

(d) (e) (f)

Figure 4: Under-segmentation error: data (a); ground-truth (b); 12-class (c) and 4-class (d) segmentations; an unreliable (e) and a reliable (f) initial state.

In [8] it was therefore proposed to perform a further split of the initial color states by first collecting all the isolated fragments having the same color, and then grouping them again in a small number of clusters based on their spatial characteristics, including their shape, orientation, and especially spatial context, in terms of directional adjacency with other color states. In fact, as it is clear from Fig.2 itself, the *grass* and *green spots* semantic classes can actually be distinguished from one another if we take into account their different spatial properties, and in particular the fact that "grass" regions border mainly regions of the "trees" class, while "green spots" are typically adjacent to regions of the "roads" and "buildings" classes.

To take into account such properties, for each connected region, a set of transition probabilities is computed, $p_j(\omega'|\omega)$ similar to those defined in equation 1, but for the fact that the initial state ω is now given, being the color of the region of interest. Such probabilities form a vector that, after a principal component analysis meant to keep only a few relevant components, becomes a feature vector $\mathbf{F}_k^{\omega}$ that characterizes the k-th fragment of color ω in terms of its size, shape, and spatial interaction with other fragments, and that will be used for the subsequent K-means clustering.

The well-known problem with clustering is that the number of clusters in data is not known in advance and difficult to estimate. In simple images, most colors are well described by just one or two clusters, but it can also happen that a single color corresponds to a very large number of clusters, maybe also for the presence of data outliers. Whenever the number of clusters is underestimated, errors may occur, mostly consisting in *under*-segmentation. Indeed, if a cluster with fragments belonging to two different textures is identified as an elementary state, and is a large-scale state, with high score, it will likely absorb other states from both textures during the merging process, producing a single texture instead of two.

An example of this phenomenon is shown in Fig.4. The image to be segmented (a) is a synthetic mosaic of different textures, with the corresponding ideal segmentation shown in (b). When the merging process is stopped at 12 classes

the corresponding segmentation map shown in Fig.4(c) is already unsatisfactory, with a very large segment (in light grey) spanning three different textures T_1, T_2 and T_3. Of course, such a problem cannot be solved by subsequent merging steps and eventually the 4-class segmentation in (d) is the best we can have starting from (c) but misses two classes out of six. The origin of this problem is the presence, from the beginning, of several elementary states shared by two textures, some corresponding to white regions (linking T_1 and T_2) and others to black ones (linking T_1 and T_3). One of such states which contributes to the fusion of textures T_1 and T_2 is shown for example in part (e). On the other hand, the presence of reliable high-score initial states, like the one shown in part (f), is fundamental for the recovery of the correct textures, since they strongly attract smaller regions, and this is the reason why some form of clustering is needed anyway.

Once identified the problem, we experimented with several solutions, first trying to find a reasonable rule to select adaptively for each color the number of clusters to use in the K-means, then testing different clustering techniques, such as Fuzzy C-means or mean-shift procedures, always with mixed and inconclusive results.

Eventually, we devised a solution strictly based on the observations made above. We keep using the K-means clustering algorithm, with a fixed number of clusters decided in advance and equal for all colors, since more complex techniques do not seem to provide any benefit. Then, for each cluster ω we compute a reliability figure R_ω: when such a figure falls under a given threshold the cluster is considered unreliable and totally disaggregated, namely, each component fragment becomes a different initial state. This is not really a problem since such isolated fragments will typically have a small texture score and will be soon absorbed by neighboring reliable states. Instead, it is important to guarantee that enough reliable states, like the one of Fig.4(f), survive to become seeds of well structured textures. To this end, the reliability threshold is set so that the reliable states cover an aggregate area of at least 50% of the image.

The reliability figure R_ω is defined by

$$R_\omega^{-1} = \frac{1}{W^\omega} \sum_{k=1}^{N_\omega} W_k^\omega \|\mathbf{F}_k^\omega - \mathbf{F}^\omega\|^2, \qquad (5)$$

where N_ω is the number of fragments in cluster ω, $\mathbf{F}_k^\omega$ is the feature vector of the k-th fragment, $\mathbf{F}^\omega$ the average feature vector of the cluster, and the W_k^ω are suitable weights, with W^ω their sum. In practice, the inverse of R_ω measures the dispersion of the feature vectors around their means. The weights were originally set equal to the fragment areas A_k^ω, to account for the fact that the k-th fragment is representative of A_k^ω pixels, but then, supported by the experimental results, we switched to the log of the areas to avoid that large dominant fragments weigh too much in the overall figure, hiding the risks coming from smaller fragments.

It is important to underline that all clusters, irrespective of the original color, are put in the same list, and sorted according to their reliability figures. Therefore, it can legitimately happen that all clusters of the same color are declared unreliable and disaggregated, something that could not have been achieved by simply changing the clustering algorithm.

4. EXPERIMENTAL RESULTS AND COMMENTS

The basic version of the TFR has been already assessed and compared with several reference techniques by means of experiments on the Prague benchmark [11]. Results, available on the same benchmark website, and extensively discussed in [9], are quite favorable to the TFR algorithm. More recently, the algorithm has been used successfully [12, 13] for the segmentation of real remotely sensed images.

In this work we will therefore focus our attention on the effectiveness of the improved solution by comparing experimental results on the 512×512-pixel color texture mosaics of the Prague benchmark with those of the original TFR. Before turning to segmentation results, it is worth underlining that the new version of the algorithm has about the same computational complexity of the original TFR (about 20 seconds of CPU time on a notebook with a 1.66 GHz processor for the test images) which is almost entirely due to the pixel-based processing for the initial color class formation.

In Fig.5 we show some insightful results over mosaics that the TFR was not able to segment correctly, and that the proposed solution, instead, deals with successfully[2]. For the first image, both algorithms identify correctly the six textures, but the original TFR has a much higher misclassification rate, especially over the two most complex textures, T_1 and T_2. In the three subsequent images a more serious problem of under-segmentation has occurred with the original algorithm. For the second image (already used in the example of Fig.4) TFR detects only four out of six classes, while the new algorithm successfully detects all of them, even if with some residual artifacts. Likewise, for the third image, the old version is able to single out only five of the eight textures while the proposed solution captures them all.

For the last image, instead, the interpretation of results is a bit more complex, since the TFR segmentation map exhibits both under- and over-segmentation. In fact, textures T_3 and T_7 are merged, and the same happens partially with texture T_1 and T_6; at the same time, both T_4 and T_6 are split over two classes. Just as before, the modified version of TFR solves satisfactorily the under-segmentation phenomena. As for the oversegmentation, texture T_6 is now correctly represented, while the problem persists with T_4, and there is probably little hope to do a better job since the texture itself is rather adverse and even a human observer may choose to identify two separate textures in it. In addition, also texture T_3 is now oversegmented, probably as an unwanted effect of the state ranking procedure which has disaggregated too many states belonging to that texture. Nonetheless, the 9-class partition obtained with the new algorithm is totally compatible with the 7-class ground-truth, which could be obtained by means of just two suitable merging steps. Unfortunately, in this case the merging process, driven by the texture score, gives higher priority to the merging of other classes (T_6 absorbed by T_1) before recomposing T_3 and T_4.

In conclusion, the figure R_ω proposed here seems to be a good indicator of the reliability of the initial states. By removing all unreliable states singled out by such a measure, while at the same time keeping enough reliable states to work as texture seeds, most of the undersegmentation phenomena disappear.

[2] It is worth underlining that the new solution provides the same good results as the TFR for the other mosaics not shown here.

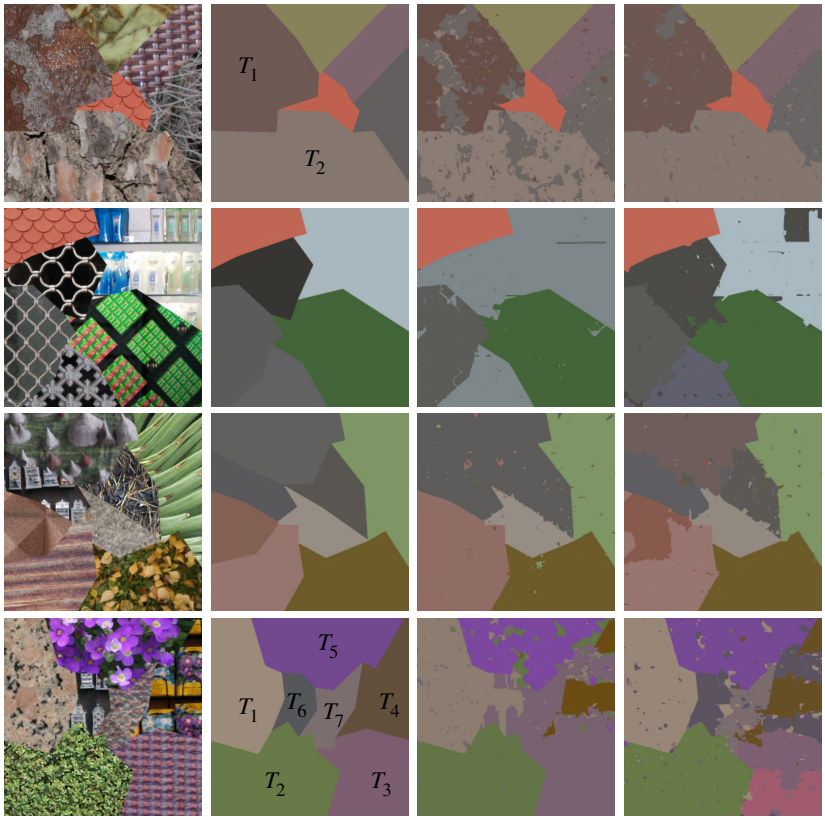


Figure 5: From left to right: texture mosaic, ground-truth, segmentations by original and (resp.) proposed algorithm.

As the last experiment indicates, future work should focus on the merging process, since the image structure is not always correctly identified. In particular, the case of texture T_4 shows an intrinsic weakness of the texture score measure which allows inter-texture fusions to take place before intra-texture ones, only because of the region scale. In fact, since the subtextures of T_4 happened to occur at a larger scale than the whole texture T_6, the latter was subject to merging before texture T_4 was recomposed. We are currently investigating the use of an adaptive texture score formula, where the relative weights of spatial scale (first two factors in (3)) and context (last factor) change as a function of the scale of the textures currently detected.

REFERENCES

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

[2] R.M.Haralick, "Statistical and structural approaches to texture," *Proc. of the IEEE*, pp.786–804, May 1979.

[3] S.Krishnamachari and R.Chellappa, "Multiresolution Gauss-Markov random field models for texture segmentation," *IEEE Trans. Im. Proc.*, pp.251–267, Feb. 1997.

[4] M.Haindl and S.Mikeš, "Colour texture segmentation using modelling approach," In *Proc.3th ICARP*, LNCS 3687, pp.484–491, Bath, UK, 2005.

[5] M.Unser, "Texture classification and segmentation using wavelet frames," *IEEE Trans. Im. Proc.*, pp.1549–1560, Nov. 1995.

[6] O.Pichler, A.Teuner and B.J.Hosticka, "An unsupervised texture segmentation algorithm with feature space reduction and knowledge feedback," *IEEE Trans. Im. Proc.*, pp.53–61, Jan. 1998.

[7] B.B.Chaudhuri and N.Sarkar, "Texture segmentation using fractal dimension," *IEEE Trans. PAMI*, pp.72–77, Jan. 1995.

[8] G.Scarpa and M.Haindl, "Unsupervised texture segmentation by spectral-spatial-independent clustering," *Proc. ICPR 2006*, pp.151–154, Hong Kong, China, 2006.

[9] G.Scarpa, M.Haindl and J.Zerubia "A hierarchical finite-state model for texture segmentation," *Proc. ICASSP 2007*, pp.I-1209-1212, Honolulu, HI (USA), April 2007.

[10] C.D'Elia, G.Poggi and G.Scarpa, "A tree-structured Markov random field model for Bayesian image segmentation," *IEEE Trans. Im. Proc.*, pp.1259–73, Oct. 2003.

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

[12] G.Scarpa, M.Haindl and J.Zerubia, "A hierarchical texture model for unsupervised segmentation of remotely sensed images," *Proc. SCIA 2007*, pp.303-312, 2007.

[13] R.Gaetano, G.Scarpa and G.Poggi, "Hierarchical texture-based segmentation of multiresolution remote sensing images," to appear in *IEEE Trans. Geosci. Rem. Sens.*