A HIERARCHICAL FINITE-STATE MODEL FOR TEXTURE SEGMENTATION

Giuseppe Scarpa[†‡*], *Michal Haindl*[‡], *Josiane Zerubia*[†]

[†] ARIANA Research Group, INRIA/I3S, Sophia Antipolis, France.
[‡] Pattern Recognition Dep., ÚTIA, Academy of Sciences, Prague, Czech Republic.
[*] DIET, University Federico II, Naples, Italy.

ABSTRACT

A novel model for unsupervised segmentation of texture images is presented. The image to be segmented is first discretized and then a hierarchical finite-state region-based model is automatically coupled with the data by means of a sequential optimization scheme, namely the Texture Fragmentation and Reconstruction (TFR) algorithm. Both intra- and inter-texture interactions are modeled, by means of an underlying hierarchical finite-state model, and eventually the segmentation task is addressed in a completely unsupervised manner. The output is then a nested segmentation, so that the user may decide the scale at which the segmentation has to be provided. TFR is composed of two steps: the former focuses on the estimation of the states at the finest level of the hierarchy, and is associated with an image fragmentation, or over-segmentation; the latter deals with the reconstruction of the hierarchy representing the textural interaction at different scales.

Index Terms— Segmentation, classification, co-occurrence matrix, structural models, Markov chain, texture synthesis.

1. INTRODUCTION

Image segmentation [1, 2, 3, 4] is a low-level processing which is of critical importance for many applications in several domains, like medical imaging, remote sensing, source coding, and so on. Although it has been widely studied in the last decades in many cases it remains still open, as for textured images, where the spatial interactions may cover long ranges asking for high order complex modeling. Indeed, in the unsupervised case the situation is much more critical as testified by experimental results provided below.

There are a large number of approaches to segmentation, but due to space limitations, here we confine ourselves to reviewing only those that have been tested using the same benchmarking system [5] as we use, and which therefore serve as points of comparison. In [6] image blocks are modeled by means of local GMRF and the segmentation is performed in the parameter space by assuming an underlying Gaussian Mixture. Similar to the previous, but with an auto-regressive 3-D model (AR3D) in place of the GMRF, is the method presented in [3]. In [7] an approach, namely the JSEG, is presented where segmentation is achieved in two steps: a color quantization followed by a processing of the label map which accounts for spatial interaction. Another method taken in consideration is the segmentation algorithm underlying the CBIR system *Blobworld* [1]. Here a Gaussian Mixture model is assumed in a feature space, where contrast, anisotropy and polarity are the salient texture descriptors, and the EM algorithm carries out the clustering. Finally, the algorithm presented in [8] (EDISON) combines a region-based approach with a contour-based one, hence balancing the global evidence which characterizes a region-based model with the local information typically dominant in the contour modeling.

In this work we present a method based on a hierarchical finite-state probabilistic texture modeling. The model is coupled with an optimization scheme, namely the Texture Fragmentation and Reconstruction (TFR) algorithm, which first estimates the states of the finest level (fragmentation), and then relates them hierarchically (reconstruction) as to provide the desired hierarchical segmentation. All the methods cited above, as well as two versions of the proposed method, were compared through the Prague benchmark [5].

2. TEXTURE MODEL

In this work we present a *hierarchical*, *discrete* and *region-based* probabilistic model for texture representation, which is particularly suited for *unsupervised* image segmentation. In order to apply the model, an early processing is then needed to provide a discrete image that roughly represents the original data. In general this processing may be any known pixel-wise texture feature extraction followed by a clustering, but in practice we reduce it to a simple color-based segmentation, since the textural information will be handled in the discrete space. This operation is associated with an information loss which reduces the description capability of the model. However, while this could be a serious limit in a synthesis framework, it is not so critical in analysis problems, like segmentation, and especially in an unsupervised case where robustness,

This work was carried out during the tenure of an ERCIM fellowship (Scarpa's postdoc), and supported by EU MUSCLE project (e-team: shape modelling), FP6-507752, and partially by the project 1ET400750407.

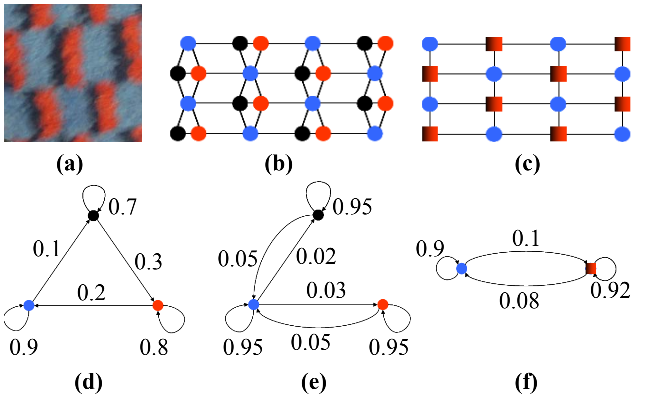

Fig. 1. Hierarchical texture model. Textile pattern (a); H-RAG: 3- and 2-state RAG, (b) and (c) respectively; 3-state chain models for east and south directions, (d) and (e) respectively; and 2-state chain for east direction (f).

rather then precision, is often the most relevant issue.

To introduce the model, let us consider the example in Fig.1, where a textile pattern (a) is associated with some graphical representations. Imagine first a simple 3-level discrete approximation of the data (say, the color-states *blue*, *black* and *red*), and consider its partition in uniform connected regions. A Region Adjacency Graph (RAG) representation of this partition is shown in (b). Likewise, in case of a 2-state partition (for example, let *black* and *red* collapse in a single state) we would get a RAG like that depicted in (c). Notice that, by merging state *black* with *red* without involving the *blue* one, we established a clear relationship between the two graphs, which form together a Hierarchical RAG (H-RAG). In this toy example the H-RAG has only two layers because we have considered only two nested partitions, but in practice it has usually more layers as we start from much finer segmentations.

Now, let us observe how the textural properties are reflected in the adjacency graphs (b) and (c) as cyclic occurrence (strictly periodic in the specific example) of subgraphs of three and two nodes, respectively for (b) and (c). Such phenomenon can be synthetically represented for any given spatial direction by means of state diagrams, as in (d) and (e) for directions east and south respectively, when three color states are considered (b), and in (f) for east direction if we have only two states (c). As well as the RAGs, and for the same reasons, these diagrams are hierarchically related for any given direction, (see for example (d) and (f)). The example also clearly shows that, for a fixed periodical texture component, the coarser the scale of the RAG representation, the lesser the order at which it is revealed on the graph. In other words, the multiscale representation allow us to represent simultaneously both micro- and macro-textural features with the same (low) order but in different layers of the hierarchical model.

As can be seen, the compact representation (d)-(f) not only accounts for the adjacency among states but also for their directionality (mutual positioning) and relevance, through the specification of transition probabilities, TPs, on a pixel-by-pixel step basis. Approximated TPs are indicated on the graphs just to give an idea of their relationship with the visual appearance of the texture. In particular, observe that intra-region TPs account for the shape of the texture components. As an example, consider the blue patches that regularly occur in the sample. Due to their rectangular shape, the associated intra-region TP in the vertical direction (e) is larger then the horizontal one (d). The remaining, inter-region, TPs accounts instead for the spatial context, that is, the relative occurrence and positioning of the neighboring regions.

More precisely our texture model refers to the graphical representations introduced above and is basically a simultaneous hierarchical finite-state Markov model that for a given texture is completely defined by the triple $(\Omega, \mathcal{T}, \mathcal{P})$, where Ω is the set of states of the finest, but discrete, version of the texture, $\mathcal{T}$ is a tree structure representing the hierarchical relationships among the states[1] and, finally, $\mathcal{P} = \{\mathbf{P}_\omega\}_{\omega\in\Omega}$ is the set of TP matrices (TPMs) for the terminal states. TPMs are given by

$$\mathbf{P}_\omega(\omega', j) = \frac{|\mathcal{S}_{\omega \xrightarrow{j} \omega'}|}{|\mathcal{S}_\omega|} \qquad \forall \omega' \in \Omega, \;\; 1 \le j \le 8,$$

where $\mathcal{S}_\omega$ is the set of pixels with state ω and $\mathcal{S}_{\omega \xrightarrow{j} \omega'}$ is the restriction of $\mathcal{S}_\omega$ to the sites whose neighbour in position (direction) j belongs to state ω'. While the TPMs defined above describe globally a texture, a single connected region element n of a given state ω has itself an own TPM, $\mathbf{P}^n_\omega$, computed through the same formula but restricted to the region $\mathcal{S}^n_\omega \subseteq \mathcal{S}_\omega$.

Observe that at coarser level representations the states are completely defined by combination of related offspring states according to the given structure $\mathcal{T}$, which means that their TPMs are derived by simple weighted averages. Moreover, notice that in general a color may occur in a texture according to different configurations, hence increasing the number of states which do not necessarily represent different colors.

3. TFR ALGORITHM

Let us consider now the application of the above modeling in the particular case of unsupervised segmentation. The image to be segmented is then a composition of an unknown number of different textures whose corresponding models are unknown as well and need to be estimated during the process of texture identification. The model fitting consists in estimating the states (with related TPMs) at the finest scale and the hierarchical tree which univocally defines each intermediate state.

[1] Hence, the states of Ω are associated with the terminal nodes, while the root represents the whole image.

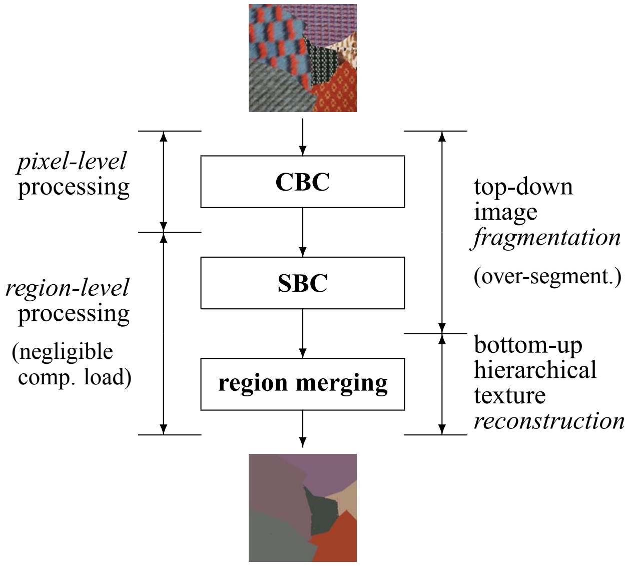

Fig. 2. TFR algorithm flow chart.

The determination of the number of textures of a given image, classically referred to as *cluster validation problem*, is strictly related to the spatial scale (hence to the hierarchical structure) at which we are interpreting the image. When the scale is not fixed somehow, the cluster validation becomes an ill-posed problem. To give an example, the same texture of Fig.1 may be interpreted as a composition of three different textures if we refer to a finer scale.

As a consequence we aim at solving this problem simultaneously with the estimation of the internal structures, according to the model defined above. In practice, this means that we fit the image with only one hierarchical model which (when correctly derived) includes as non-overlapped substructures the marginal models associated with the single textures. Then, by specifying a spatial scale, we automatically get the proper pruning of the structure which provides us with the marginal models and the associated image partition.

In order to estimate this overall model we realized the optimization scheme shown in Fig.2, namely the Texture Fragmentation and Reconstruction (TFR) algorithm, which first extracts a proper number of terminal states through the top-down fragmentation step, composed of blocks CBC (Color-Based Clustering) and SBC (Spatial-Based Clustering), and then relates them by means of a recursive bottom-up merging step, as to reconstruct the whole hierarchical structure.

The estimation of the states is performed in two steps, the former (CBC) dealing with color information, hence working at pixel level, the latter (SBC) focused on the spatial information at region level in the TPM space. In principle, CBC may be any color quantization process, but in our implementation we preferred the use of the TS-MRF (tree-structured Markov random field) segmentation algorithm [2], since it avoids the generation of punctual regions (which are not reliably characterized in terms of TPM) due to regularization of the MRF. Furthermore, the tree-structured formulation ensures a quick processing and allows to balance the energy among the discrete color states.

Once the color segmentation has been obtained, we switch to a region-based representation, by taking connected regions with uniform color as basic elements characterized by TPMs. Since the color of a region only partially defines its state[2], the SBC applies to each set of elements with common color, as to split it in subgroups which are homogeneous also in terms of TPM, that is providing the desired states. The split is realized by means of a k-means algorithm applied in the feature space resulting from a PCA (Principal Component Analysis) on the TPM space. The PCA was necessary because of the large dimensionality of the full feature space w.r.t. the number of elements which does not allow a reliable characterization.

Region merging, or state merging, is nothing but a sequential binary combination of the states driven by a specific parameter, namely the *region gain* which accounts for the mutual spatial relationships among the corresponding regions. Indeed the merging selection process is not symmetric, as the gain is a measure of the scale of the region weighted by an additional term which quantifies the attraction operated by the other regions (candidates for the merging). The scale factor allows to privilege always the merging of small regions so that the final hierarchy is such that micro-textural features are represented at the bottom, while the macro will appear at upper levels, and finally inter-texture mergings will be placed at the top of the structure, in order to keep separate the marginal sub-models corresponding to the different textures.

In this work we compare two different region gains. The former is defined as

$$\mathcal{G}^i \triangleq \frac{p(s\in\mathcal{R}_i)}{\max_{j\neq i} p(r\in\mathcal{R}_j|s\in\mathcal{R}_i)} =$$
$$p(s\in\mathcal{R}_i)\cdot\frac{1}{p(r\notin\mathcal{R}_i|s\in\mathcal{R}_i)}\cdot\frac{p(r\notin\mathcal{R}_i|s\in\mathcal{R}_i)}{\max_{j\neq i} p(r\in\mathcal{R}_j|s\in\mathcal{R}_i)}$$

where $\mathcal{R}_i$ is the region of interest, s is an image site and r is any of the eight neighbours of s. The first two factors represent the scale, since one is proportional to the area of the region and the other quantifies its compactness. The third term, instead, accounts for the relative occurrence of the nearest neighbour region (context).

The latter is a modification of the former where the contextual term has been reinforced by means of the Kullback-Leibler Divergence (KLD), $D(q_i\|q_j)$, between the region spatial distributions, that is

$$\log\mathcal{G}^i_{\mathrm{KL}} \triangleq \min_{j\neq i}\left\{\log\frac{p(s\in\mathcal{R}_i)}{p(r\in\mathcal{R}_j|s\in\mathcal{R}_i)} + D(q_i\|q_j)\right\},$$

where q_i and q_j are normals (see details about KLD in [9]).

[2] More states may correspond to the same color, because either it appears in different configurations in a texture or it occurs in two different textures.

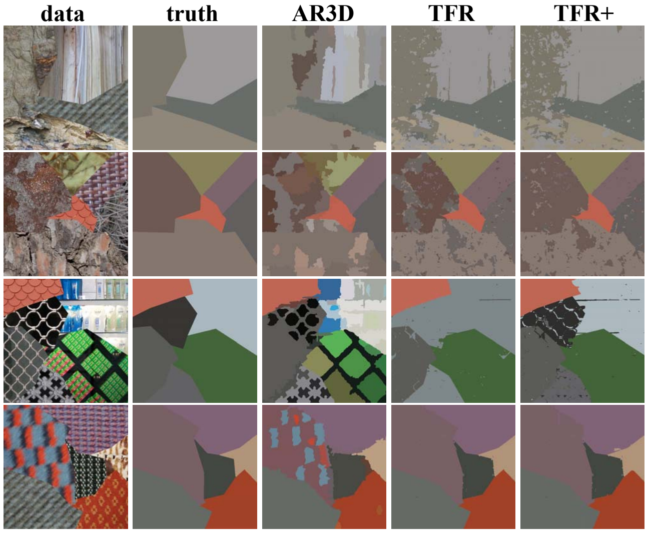

Fig. 3. Segmentation of data sets nr. 2, 4, 14 and 19, respectively

4. EXPERIMENTAL RESULTS

The proposed algorithm, that is the TFR or the TFR+ (when the gain includes the KLD term), is compared with other algorithms (briefly recalled in the introduction) which were tested on the same benchmark system [5], that provides mosaics of real natural textures. The system provides a comparison w.r.t. a large number of indicators, some of which are region-based, some others are pixel-wise accuracy indicators, and a few of them give a measure of consistency. A complete description of all the parameters, as well as all the results presented here, can be found on the system website [5].

The interpretation of the numerical results (Tab.1) would require a description of all the indicators involved, which cannot be satisfied due to space limitations. Hence we invite the interested readers to refer to [5] for this purpose and draw their own conclusions. Nonetheless it can be easily recognized that the two versions of TFR seem to outperform the others w.r.t. many indicators, with TFR+ being generally better than TFR. Indeed, the visual inspection of the different segmentations allows an easier and convincing interpretation. In Fig.3 we reported the segmentations just for a few (4) representative texture mosaics over a set of twenty. Also we only consider the best among all the comparative algorithms, that is the AR3D. The main drawback for the AR3D, as well as for those not shown, is the tendency to over-segment, contrarily to TFR that has a tendency to under-segment and is outperformed by TFR+ that eventually reaches the best tradeoff.

5. REFERENCES

[1] C.Carson, M.Thomas, S.Belongie, J.M.Hellerstein, and J.Malik, “Blobworld: A system for region-based image indexing and retrieval,” in *3th ICVIS*, 1999.

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

Table 1. Up arrows indicate that larger values of the parameters are better; down arrows, the opposite. Benchmark criteria: CS, correct segmentation; OS, over-segmentation; US, under-segmentation; ME, missed error; NE, noise error; O, omission error; C, commission error; CA, class accuracy; CO, recall - correct assignment; CC, precision - object accuracy; 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; GCE (LCE), Global (Local) Consistency Error.

[2] C. D’Elia, G. Poggi, and G. Scarpa, “A tree-structured Markov random field model for Bayesian image segmentation,” *IEEE Trans. on IP*, vol. 12, no. 10, pp. 1259–1273, Oct. 2003.

[3] M.Haindl and S.Mikeš, “Colour texture segmentation using modelling approach,” in *Proc. ICAPR (2)*, 2005, pp. 484–491.

[4] G. Poggi, G. Scarpa, and J. Zerubia, “Supervised segmentation of remote-sensing images based on a tree-structured MRF model,” *IEEE Trans. on GRS*, vol. 43, no. 8, pp. 1901–1911, August 2005.

[5] S. Mikeš and M. Haindl, “Prague texture segmentation data generator and benchmark,” *ERCIM News*, , no. 64, pp. 67–68, 2006. http://mosaic.utia.cas.cz.

[6] M. Haindl and S. Mikeš, “Model-based texture segmentation,” in *Image Analysis and Recognition*, A. Campilho and M. Kamel, Eds., Berlin, 2004, LNCS 3212, pp. 306–313, Springer-Verlag.

[7] Y. Deng and B. S. Manjunath, “Unsupervised segmentation of color-texture regions in images and video,” *IEEE Trans. on PAMI*, vol. 23, no. 8, pp. 800–810, 2001.

[8] C.M. Christoudias, B. Georgescu, and P. Meer, “Synergism in low level vision,” in *16th ICPR*, 2002.

[9] W.D.Penny, “Kullback-Liebler divergences of normal, gamma, dirichlet and wishart densities,” Tech. rep., Wellcome Dep. Imag. Neurosc., Univ. College Longon, 2001.