2009 Ninth International Conference on Intelligent Systems Design and Applications

Texture-based segmentation of very high resolution remote-sensing images

Raffaele Gaetano, Giuseppe Scarpa and Giovanni Poggi
DIBET, University "Federico II"
Naples, Italy
Email: {firstname.lastname}@unina.it

***Abstract*—Segmentation of very high resolution remote-sensing images cannot rely only on spectral information, quite limited here for technological reasons, but must take into account also the rich textural information available. To this end, we proposed recently the Texture Fragmentation and Reconstruction (TFR) algorithm, based on a split-and-merge paradigm, which provides a sequence of nested segmentation maps, at various scales of observation.**

Early experiments on several high-resolution test images confirm the potential of TFR, but there is room for further improvements under various points of view. In this paper we describe the TFR algorithm and, starting from the analysis of some critical results propose two new version that address and solve some of its weak points.

I. Introduction

High resolution remote sensing images exhibit a considerable amount of textural information at various scales, from "micro" textures (roofs, road surfaces, basic land covers, etc.) to more complex "macro" structures (e.g. urban areas), which must be taken into account to address effectively the problem of image segmentation. While a good deal of work exists on micro-textures, e.g. [1], [2], less attention has been devoted to large-scale textures, also because of their more elusive properties which make them difficult to characterize and single out. To overcome this limitation, we have recently proposed the Hierarchical Multiple Markov Chain (H-MMC) model [3], [4], which proved especially effective in describing large-scale textures, and is the basis for a fully unsupervised segmentation algorithm, called *Texture Fragmentation and Reconstruction* (TFR), which has been successfully used, among other applications, for the segmentation of multiresolution Ikonos images [5].

The TFR follows the split-and-merge paradigm as shown in Fig. 1(a). In a first *fragmentation* phase, clusters of elementary regions that share the same spectral, geometrical, and contextual properties are detected. Such clusters are then regarded as states of a set of Markov chains, and characterized in terms of their empirical transition probabilities. Based on such probabilities, the elementary states are then pairwise merged during the *recostruction* phase, giving rise to a hierarchy of nested segmentation maps at different scales of observation, $S_2, \ldots, S_N$.

TFR has many desirable properties. To begin with, it has a moderate computational burden, since all processing steps, but for the first clustering, operate on regions rather than pixels. Then, it provides a rich *multiscale* description of the image, particularly useful in the presence of multiple heterogeneous land cover types. More peculiar, it allows for the identification of macrotextures that span large areas of the images and cannot be easily characterized by synthetic features. Indeed, at the higher scales of observations, when just a few regions survive the merging process, TFR is typically able to tell apart the major semantic areas of the image based on their structural properties, with little or no attention to the spectral signature of individual pixels. A good example is provided by Fig.2 where the 2-class top segmentation divides clearly the image in vegetation and urban area, and the latter class correctly includes a large number of vegetation spots embedded in residential blocks despite their spectral features.

Besides such evident successes, however, there have also been more puzzling results in the experiments, which shed some light on the weak points of the algorithm and motivate the ongoing research. A good "bad" example will be shown in the experimental results where both the 3-class and 4-class segmentations fail (in different ways) to tell apart the major semantic areas of the image (but for the sea). We will follow that example through the rest of the paper, analyzing the problem genesis and proposing and testing two developments of the original TFR algorithm that address it. Next Section provides some basic concepts about TFR and the underlying H-MMC image model. Section 3 focuses on the problem of background regions, which leads to the cited TFR failures. Then Section 4 describes an additional step of the algorithm, based on geometrical properties, that tackles this problem while Section 5 proposes in alternative a recursive version of the TFR. Section 6 finally discusses experimental results.

II. The TFR algorithm

In this Section we will provide a very high-level description of the texture fragmentation and reconstruction algorithm in order to convey just the main ideas and help understanding both its potential and limitations. The interest reader is referred to [4] for more detail and experimental results.

The algorithm, whose flow chart, is depicted in Fig.1(a), comprises three steps: color-based classification (CBC), spatial-based clustering (SBC), and texture merging.

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.63

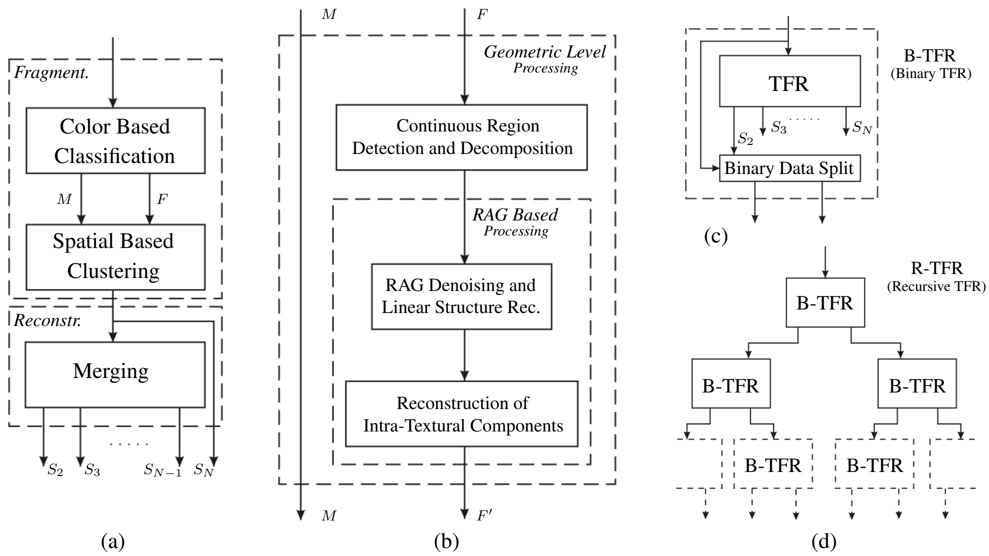

Figure 1. Flow charts: TFR (a), its binary restriction, B-TFR (c), the proposed geometric processing layer (b) and the proposed recursive TFR, R-TFR (d).

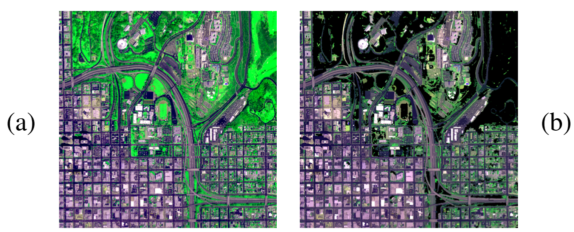

Figure 2. Binary segmentation through TFR: image of San Diego (a), and the urban segment (b). Black areas on (b) correspond to vegetation.

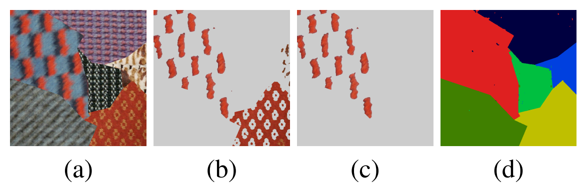

Figure 3. TFR evolution: A texture mosaic (a), a color state singled out by CBC (b), a texture state (c) provided by SBC applied to color state (b), and the final 6-class segmentation map (d) drawn from the final hierarchical segmentation.

The first step carries out a segmentation of the image pixels based on their “color”, that is, their spectral properties or, more in general, some extended feature vectors associated with them. All pixels with the same color form a class, where, for highly textured images, the class is typically spread over the whole image giving origin to a large number of color-homogeneous connected fragments. In Fig.3(a), for example, drawn from the Prague texture database [6], a “red” class is clearly present, which gives origin to quite a few isolated fragments, see Fig.3(b). Any reasonable pixel-based segmentation technique can be used to carry out this first step, from plain K-means to more sophisticated bayesian algorithms.

Looking at our example image, we would like to tell apart six spatial regions, each characterized by a different texture. Color properties cannot help much, as the same colors appear again and again in several textures, so we must definitely resort to spatial properties. Indeed, from Fig.3(b) it is clear that all red fragments of a given texture share similar features in terms of size, shape, and contextual properties, that is, relations with neighboring fragments of different colors. Therefore, if we are able to associate a suitable feature vector with each fragment we should find out that vectors related to fragments of the same texture form a well-defined cluster in the feature space. This is exactly what is done in step two of the TFR algorithm. Each fragment is characterized by a set of probabilities $p_j(\omega'|\omega)$ giving the likelihood of finding a pixel of color ω' when leaving a pixel of the fragment (of color ω) along the direction j. The red fragments of the upper-left texture of Fig.3(a), for example, will be characterized by a large probability of finding a blue pixel when moving right and a black pixel when moving left. Based on such feature vectors, the spatial based clustering step is able to collect together quite reliably fragments belonging to the same texture, as shown in Fig.3(c).

The final step has the goal of recovering complete textures by merging together groups of fragments (also called regions or “states” in TFR) that have a tight spatial relationship, like the red, blue and black patches of the upper-left texture mentioned before. To this end, each state is characterized by a "texture score", related to its size, compactness, and distribution of neighbors, which indicate whether it is a complete texture (high score), with no need of further refinements, or else just a texture component (low score), in which case it is merged with the dominant neighboring state in order to form a more complete texture. During this merging process, a hierarchy of segmentation maps, $S_N, \ldots, S_2$, is formed, starting from elementary texture states (fragments collections) to end up, in the final stages, with well recognizable textures. In this process the coarsest segmentation is the binary partition S_2. An example is shown in Fig.3(d), where the six regions are quite reliably segmented.

III. The problem of background regions

Going back to the problem of segmenting the test image of Fig.4(a), and identify its major semantic areas, a desirable output would be a 3-4 class map which separates sea, hills, and man-made area, the latter possibly further divided in residential area and harbor.

To reach this goal, all atomic fragments must be first identified in the preliminary color-based classification. Although conceptually less interesting than other steps, the initial fragmentation phase is crucial for obtaining the desired results. A good example is provided by the 4-class color-based classification of Fig.4(b) where a "background" region[1] appears. In fact, while the gray class, for example, gives rise to a large number of small connected fragments, the light green class originates some small fragments but also a very large one, which spans the whole image and goes across different semantic areas. Subsequent TFR phases treat this fragment as an atomic (literally, unsplittable) element, and are therefore incapable of telling apart distinct semantic areas that comprise parts of such a fragment, which explains the genesis of the wrong 3-class and 4-class segmentation maps shown before.

The large background region acts as a collector for smaller fragments, and therefore the residential area is either included in a single class together with the hills, in the 3-class map, or recognized as distinct, in the 4-class map, but without all the green patches (gardens, trees, etc.) associated with the hills, both clear failures of the algorithm.

To solve this problem, one can use more classes in the initial fragmentation step, but this might also lead to an excessive fragmentation of the image, unreliable estimated features, and finally an even worse segmentation. In Section 5 we will actually follow this path, but with a suitable modification of the processing flow that guarantees a stable functioning.

[1]The term "background" is used here for continuous regions of large extension. This is motivated by the fact that very often these regions actually correspond to backgrounds.

In any case, a finer fragmentation does not really solve the background region problem, but reduces only its frequency. Therefore, a further fragmentation step, based on geometrical properties, is required to solve it at its root.

IV. A Geometric Level for TFR

We improve upon the original TFR algorithm by introducing a new module in the fragmentation phase which relies mainly on the geometrical properties of the fragments provided by CBC. In particular, the CBC block provides both the color class map M and the corresponding fragment label map F. The aim is to decompose each background region $f \in F$ based on its *local* structural properties, which change when going from one textured environment to another, thus obtaining a refined fragment map F'. The main tasks performed by the new geometric level processing block are identified in Fig. 1(b) and detailed in the following.

A. Background region detection and decomposition

Candidate background regions are first selected based on their size w.r.t. to the whole image, making sure to discard only very small fragments.

For each candidate fragment $f \in F$ the following steps are then carried out:

1) a distance image D_f is built, where each pixel has a value proportional to the distance from the closest contour; Fig.4(c) shows the distance image for our example critical fragment;
2) the image D_f is segmented based on a Watershed transform [7], obtaining a number of nearly convex sub-fragments; Fig.4(d) shows such decomposition again for our example;
3) if a sufficiently large number of sub-fragments is generated f is considered critical and marked as background for further processing, otherwise, its decomposition is ignored.

Even a quick analysis of Fig.4(d) makes clear that the sub-fragments can be clustered in regions of similar characteristics, on the basis of their spatial location, size, density, connectivity, etc. However, many of such features are not easily quantified in the spatial domain, which is why we resort to a graph-based description in next step.

B. Building a region adjacency graph

In this phase a Region Adjacency Graph (RAG) [8] is associated with the decomposition of each background region, where each sub-fragments is represented by a vertex, and neighboring sub-fragments are connected by a link. Each RAG is subject to two refinement steps:

1) the denoising step aims at simplifying the RAG structure by removing single-link nodes corresponding to very small marginal sub-fragments due to small contour variations;

2) a further simplification of the RAG is obtained by recovering linear structures which are sometimes decomposed in many small sub-fragments in the Watershed transform because of noisy contours; indeed, such structures are easily recognized in the RAG since they correspond to long chains of nodes having only two neighbours but for the first and last ones.

This refinements is extremely useful both because it allows to recover linear image elements and because it drastically reduces the RAG cardinality improving the subsequent RAG characterization and navigation, with a positive impact on the final accuracy and complexity.

Fig.4(e) shows the final RAG for our running example. A few subgraphs corresponding to geometrically coherent areas have been highlighted, such as the shoreline (green), the terraced vegetation in the upper-right corner (orange) and a part of the forest in the lower part of the image (brown).

C. RAG-based featuring and clustering

The final task is to cluster the sub-fragments of a background region on the basis of their features so as to finally segment the original region in a small number of geometrically-coherent single-texture parts.

A set of features is associated with each node of the RAG, and then a conventional clustering technique, the Mean-Shift algorithm [9] in our current implementation, is used in such a feature space so as to obtain the desired RAG decomposition which corresponds to a background region partition.

Let S be a sub-fragment of the background region $f \in F$, N the corresponding RAG node, and R_N the local RAG (L-RAG) centered on N, that is, the subgraph of the whole RAG formed by all nodes which can be reached from N in no more than L steps (two, in our experiments). For each node N, we compute the following features:

- **node position**: barycentre of S, this helps obtaining a spatially compact partition;
- **node weight**: size of S;
- **node moment**: pixel-wise inertial moment of S;
- **local node density**: number of nodes within a given distance from N;
- **local link density**: number of links in R_N;
- **L-RAG moment**: discrete-mass inertial moment of R_N, with node weights as defined above;
- **L-RAG orientation**: discrete-mass average orientation of R_N, with reference to central node N.

The automatic clustering of the RAG nodes using the above features is shown in Fig.4(f) for our running example, obviously coherent with the homogeneous regions highlighted in Fig.4(e).

V. Recursive TFR

The problem of background regions can sometimes be solved simply by increasing the number of color classes in the initial fragmentation. This way, in fact, large quasi-homogeneous regions can be split from the beginning in smaller fragments based on minor variations in their local spectral characteristics. More in general, in order not to miss any relevant detail, it is necessary to carry out a very fine initial segmentation in the color-based step, that is, with quite a large number of colors. Likewise, in the spatial-based clustering, for the same reasons as above, each color class must be segmented using a large number of clusters in the feature space. Needless to say, increasing the number of clusters in each of these phases leads very easily to estimation problems, due to lack of sufficient data for statistical characterization, and inconsistencies. Such problems lead to the formation of many states that do not correspond to semantically homogeneous areas, and hence to inaccurate segmentation maps at the finer scales.

Our experiments, though, showed quite clearly that, despite such problems, the *high-level* textures recovered towards the end of the merging process do catch important semantic areas of the image. Therefore, we decided to improve upon the original TFR by developing a recursive version.

The modification is straightforward. TFR is run on the whole image, carrying out the merging process until only two textures remain. By keeping just the binary segmentation S_2 and discarding the rest of the segmentation stack $S_3, \ldots, S_N$, we have the *binary* TFR (B-TFR) as shown in Fig.1(c). Each of these two textures is then regarded as a new image (defined on an irregular domain) and possibly subject again to a binary TFR segmentation, and this process goes on, as shown in Fig.1(d), until a suitable stopping criterion is met, which in our initial implementation is again related to the texture score.

Recursive TFR offers two obvious advantages. First, because we are interested only in binary high-level segmentation, there is no need to know in advance how many colors and textures are present in the image, and the number of clusters in each phase can be significantly reduced, thus avoiding all the problems cited above, and potentially reducing computational complexity. Second, if the binary segmentation succeeds in isolating areas of very different characteristics, subsequent steps might well build upon this, by adapting the algorithm parameters to the local statistics or, more interesting, by resorting to “ad hoc” techniques, that take into account explicitly the nature of the texture under analysis. For example, if the first step isolates rural and urban areas, then quite different techniques can be used for such different areas, possibly using prior models, *e.g.* [10], tailored to each of these general classes to guide the process.

VI. Experimental Results

We have followed the processing of our test image, acquired by an airborne sensor portraying the bay of Maiori, Italy, with a spatial resolution of around 1 m and overall size of 750×500 pixels, through all intermediate processing steps.

The introduction of the geometry-based new fragmentation step has led to the partition of the critical background region in a small number of geometrically homogeneous subregions which can be easily absorbed in local, semantically relevant, textures. The final segmentation maps, shown in Fig.4(h)/(k), speak clearly in favor of the new step. In the 3-class map, sea, hills and man-made areas are very well identified, and adding a new class allows one to tell apart the harbor from the residential area.

Quite interestingly, also the recursive version of TFR provides much improved results. as shown by the final maps, shown in Fig.4(i)/(l). Dry land and sea are easily separated in the first split, then the algorithm focuses on the first region and tells apart quite accurately urban area and hills. The third split concerns again the urban area, however the two newly generated classes have a less compelling interpretation than the two urban classes of Fig.4(k), one of them being associated more with the road network (and shadows), and the other more with individual buildings. Beyond any considerations on accuracy, it is interesting to observe how the two versions of the algorithm, after the obvious initial splits, led to two readings of the scene quite different but equally meaningful.

The encouraging results obtained with both the proposed changes suggest us to merge them together in the near future in order to take benefits from both solutions. This way we expect to be able to relax further the role of the color-based classication layer, as to speed up the overall processing without loss of accuracy.

VII. Acknowledgements

Thanks to Flight Air Link Cooperation (FALCO) Project for funding.

References

[1] M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Transactions on Image Processing*, vol. 4, no. 11, pp. 1549–1560, November 1995.

[2] A. Clausi and H. Deng, "Design-based texture features fusion using gabor filters and co-occurrence probabilities," *IEEE Transactions on Image Processing*, vol. 14, no. 7, pp. 925–236, July 2005.

[3] G. Scarpa, M. Haindl, and J. Zerubia, "A hierarchical finite-state model for texture segmentation," in *In Proc. of IEEE ICASSP'07*, Honolulu, HI (USA), April 2007, vol. 1, pp. 1209–1212.

[4] G. Scarpa, R. Gaetano, M. Haindl, and J. Zerubia, "Hierarchical multiple markov chain model for unsupervised texture segmentation," *IEEE Transactions on Image Processing*, vol. 18, no. 8, pp. 1830–1843, August 2009.

[5] R. Gaetano, G. Scarpa, and G. Poggi, "Hierarchical texture based segmentation of multiresolution remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 7, pp. 2129–2141, July 2009.

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

[7] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulation," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583–599, June 1991.

[8] B. Fischer, C. J. Thies, M. O. Guld, and T. M. Lehmann, "Content-based image retrieval by matching hierarchical attributed region adjacency graphs," in *In Proceedings of SPIE*, San Diego, CA (USA), 2004, vol. 5370, p. 598606.

[9] D. Comaniciu and P. Meer, "Mean Shift: a robust approach toward feature space analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.

[10] 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*, vol. 45, no. 8, pp. 2676–2682, August 2007.

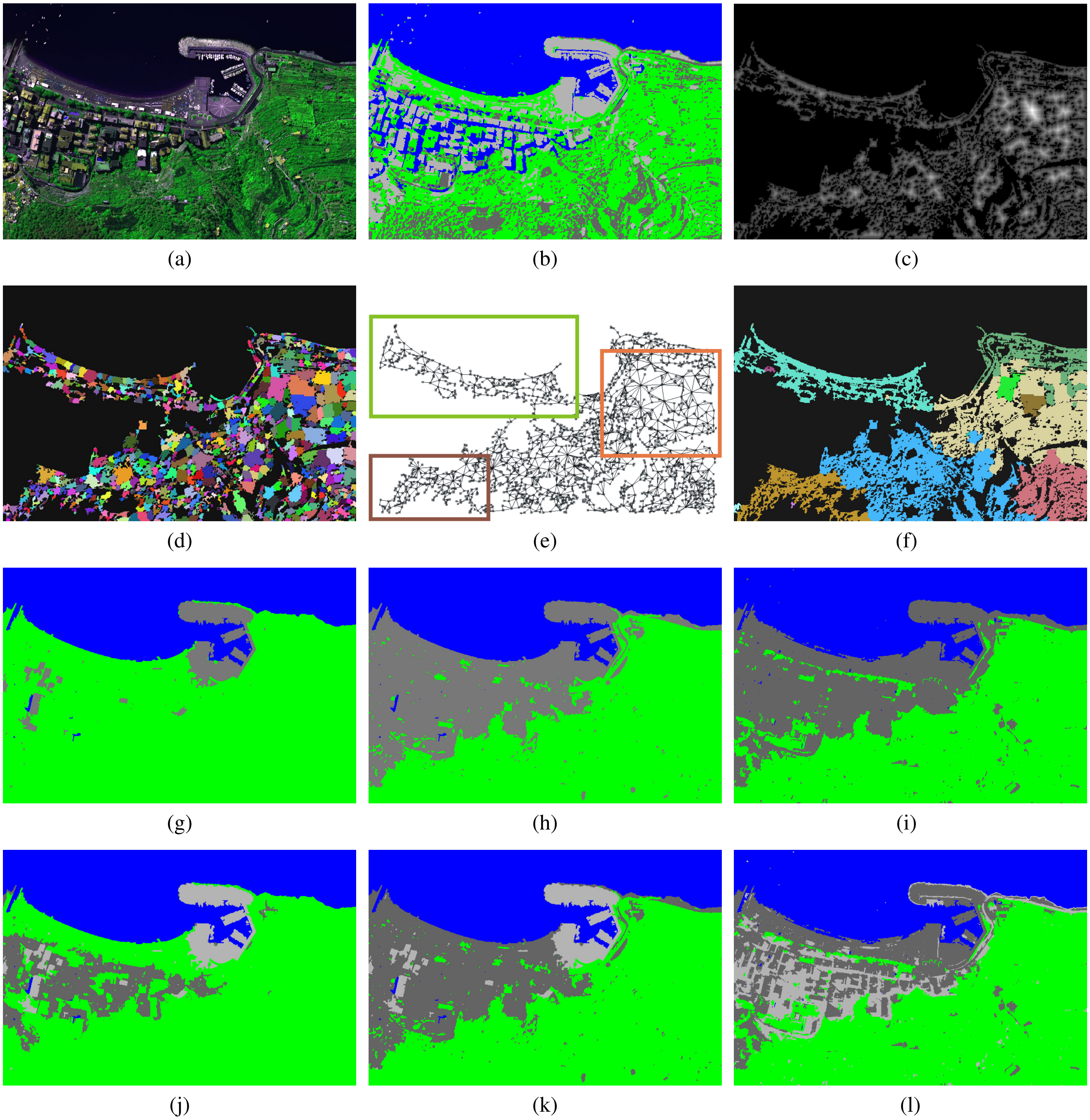

Figure 4. Segmentation results: test image (a); 4-class color-based segmentation (b); distance image D_f for a selected background region f (c); watershed decomposition of region f (d); RAG over f after simplification (e); final intra-textural components (f); 3-/4-class segmentations by TFR (g)/(j), TFR with geometric level (h)/(k), and recursive TFR (i)/(l), respectively.