Hierarchical segmentation of multiresolution remote sensing images

Camille Kurtz, Nicolas Passat, Anne Puissant, Pierre Gançarski

Hierarchical Segmentation of Multiresolution Remote Sensing Images

Camille Kurtz[1], Nicolas Passat[1], Anne Puissant[2], and Pierre Gançarski[1]

[1] Université de Strasbourg, LSIIT, UMR CNRS 7005, Strasbourg, France
[2] Université de Strasbourg, LIVE, ERL CNRS 7230, Strasbourg, France
{ckurtz,passat,gancarski}@unistra.fr, anne.puissant@live-cnrs.unistra.fr

Abstract. The extraction of urban patterns from very high spatial resolution optical images presents challenges related to the size, the accuracy and the complexity of the data. In order to efficiently carry out this task, a multiresolution hierarchical approach is proposed. It enables to progressively segment several images (of increasing resolutions) of a same scene, based on low level criteria. The process, based on binary partition trees, is partially performed in an interactive fashion, and then automatically completed. Experiments on urban images datasets provide encouraging results which may be further used for detection and classification purpose.

Keywords: Hierarchical segmentation, multisource images, multiresolution, interactive/automated segmentation, partition-trees, remote sensing, urban analysis.

1 Introduction

A new generation of sensors of submetric resolution has led to the production of Very High Spatial Resolution (VHSR) images, and to an improved ability to analyse urban scenes [**?**]. In particular, in such images, basic urban patterns (*e.g.*, individual houses, gardens, roads) are formed by different materials, while complex ones (*e.g.*, urban districts, urban blocks) generally contain different kinds of basic patterns. Thus, all of them are not necessarily composed of homogeneous pixels (but are often hierarchically organised). These specific properties of VHSR images lead to new challenges for human experts (since the size and the complexity of the images make visual analysis a time consuming and error prone task), and for image analysis tools (since those developed for lower resolutions are generally designed to extract segments based on radiometric homogeneous hypotheses).

In this context, and due to the importance to analyse VHSR images, it is then useful to develop tools adapted to the extraction of complex patterns from such data, and in particular (low-level) segmentation ones. Moreover, the availability of data with a large range of spatial resolutions can enable the extraction of potentially hierarchical patterns, especially when such data are provided by different acquisition devices, providing complementary information at distinct radiometric bands. Such segmentation tools should allow the end-user to obtain satisfactory results, at different levels (*i.e.*, scales) of pattern extraction, with minimal time (by automating the tasks which do not require human expertise), minimal efforts (by reducing the parameters), and ergonomic interaction.

We propose to take advantage of the data available at several resolutions, and to involve them in a hierarchical strategy which enables, at any given resolution, the exploration of the whole structure of an urban scene. This approach, based on the skills of the end-user, aims at using hierarchical segmentation (and especially partition trees) to make the segmentation process as automated as possible.

The article, which is an improved version of the preliminary work described in [?], is organised as follows. Section 2 provides a (non-exhaustive) state of the art on hierarchical and multiresolution segmentation. Section 3 describes the proposed segmentation method. Section 4 gathers experiments enabling to assess the relevance of the approach. Conclusions and perspectives will be found in Section 5.

2 Related works

Efforts have been conducted to automatically extract features from satellite images, in order to involve them into learning systems. This extraction, often performed thanks to low-level processing, generally relies on radiometric homogeneousness hypotheses. This can lead to valid results for basic objects extraction from High Spatial Resolution (HSR) images [?], but not for images (*e.g.*, VHSR ones) and/or objects of higher complexity [?]. A first way to extract complex objects is by grouping several basic ones, using, for instance, a graph-based approach [?]. Such techniques, devoted to the first semantic level of complex objects (*e.g.*, complex buildings) can be improved by considering hierarchical strategies.

Hierarchical segmentation methods provide, as output, a series of partitions of an image with an increasing (or decreasing) level of details. In the field of remote sensing (and especially for HSR images), several techniques have been proposed. In [?], compositions of opening and closing operations with structuring elements of increasing sizes generate morphological profiles for any pixel, enabling their characterisation. In [?], morphological profiles are enriched with neighbourhood and spectral information. Another method, relying on region connection calculus, can also be found in [?]. These approaches emphasise the potential of hierarchical segmentation. However, these "pixel-based" methods hardly take into account the intrinsic and semantic information of the images. By opposition, "object-based" segmentation hierarchies provide several segmentations of the same image at different levels of detail. Such hierarchies can be built by following two opposite paths. In the top-down approaches, the process starts from a coarse segmentation and successively refines the regions, as in [?], where segmentation is treated as a graph partitioning problem. In the, more frequent, bottom-up approaches, the finest segmentations are produced first, and their regions are then merged, based on similarity criteria [?].

In mathematical morphology, connected operators [?] may be used in a hierarchical segmentation fashion by using, for instance, tree data structures. In such structures, the nodes are associated with regions in the image whereas the edges represent the inclusion relation. Notions such as component-tree [?] and level-lines tree [?], potentially enable to perform hierarchical segmentation, by enabling the fusion of flat zones. However, such structures strongly relying on the image intensity and in particular on extremal values, the obtained segmented components may be non relevant in the case of satellite images. By opposition, the Binary Partition Tree (BPT) [?] reflects a (chosen) similarity measure between neighbouring regions, and models the hierarchy between these regions *via* the tree structure. It has been used to extract complex objects from various kinds of images [?,?]. A last approach, based on the constrained connectivity paradigm, has been recently introduced in [?] and applied to process (V)HSR images in [?]. The connectivity relation generates a partition of the image definition domain. Fine to coarse partition hierarchies are then produced by varying a threshold value associated with each connectivity constraint. In the case of remote sensing, these methods are limited by the spatial and spectral properties of the images. Indeed, complex objects appear in (V)HSR images too much heterogeneous to be extracted in an ascendant way.

This justifies the use of multiresolution data to enhance the behaviour of hierarchical ones for the extraction of such complex objects. Multiresolution methods take advantage of the data available at several resolutions (from Medium Spatial Resolution (MSR) to VHSR ones) [?], and involve them in a hierarchical strategy. By analysing first the image content at a coarse resolution and then gradually increasing this resolution, it is indeed possible to detect complex patterns while avoiding the semantic noise induced by the details [?].

From these considerations, we propose a hierarchical segmentation method, extending the BPT to deal with multiresolution images. It combines the advantages of multiresolution strategies and the efficiency of the connected operators approaches, in the context of the mapping of urban areas. It is based on interactive tree-cut segmentation (based on the skills of the end-user), automatically reproduced on the remainder of the data. The method operates first on the low-resolution data, extracting the global structure of the scene, and subsequently enriches this description thanks to the high-resolution data. It aims, in particular, at understanding the scene in the same way as the human vision system.

3 Methodology

The proposed multiresolution approach (Section 3.2, Figure 2) combines two methods: (1) a new (interactive) hierarchical segmentation process (Section 3.1, Figure 1), and (2) a multiresolution clustering method [?] (Section 3.2, Figure 3), into an original iterative process. It is devoted to hierarchically segment n images of a same scene, at various resolutions, from the lowest to the highest one.

3.1 Interactive segmentation

The first contribution is an interactive segmentation approach, enabling to segment k images of same resolution and semantics, provided by the same sensor.

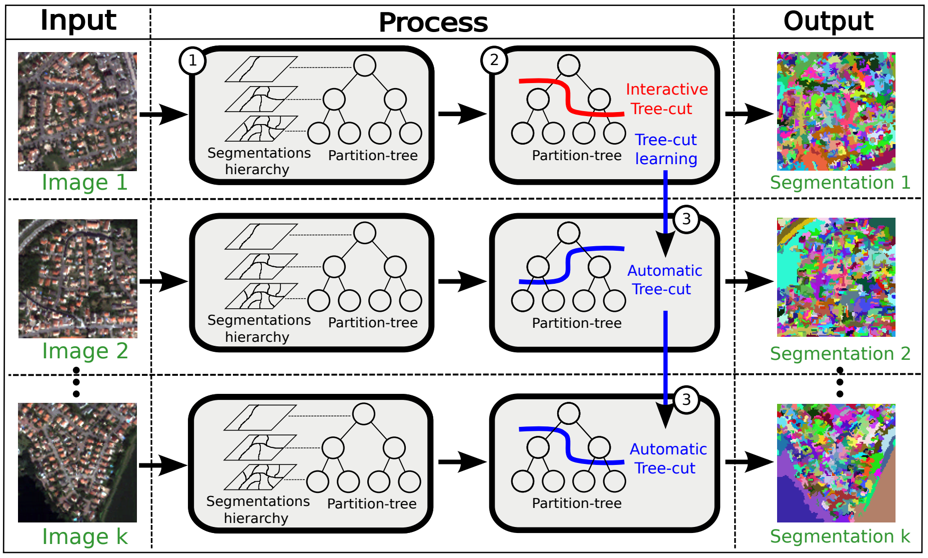


Fig. 1. Interactive segmentation approach (see Section 3.1). In green: input/output. In red: user interactions. In blue: automatic processing.

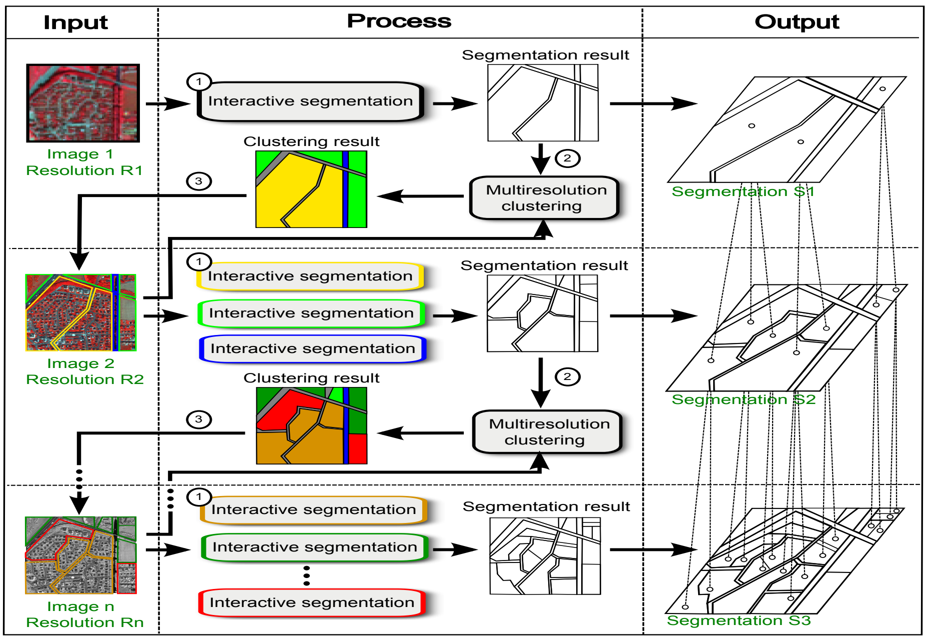


Fig. 2. Work-flow overview (see Section 3.2). In green: input/output.

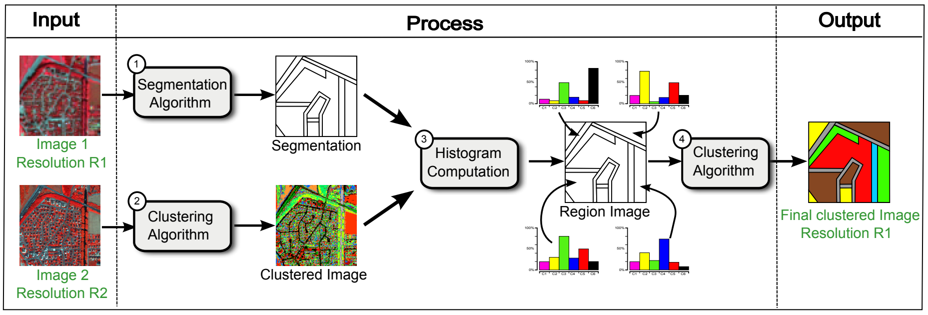


Fig. 3. Multiresolution clustering approach (see Section 3.2). In green: input/output.

From the BPT of one image (Figure 1-①) and an interactive tree-cut of this BPT, inducing a segmentation (Figure 1-②), the $k-1$ other images are automatically segmented, by reproducing a similar tree-cut in their own BPT (Figure 1-③).

Notations Let $E = [\![0, d_x - 1]\!] \times [\![0, d_y - 1]\!] \subset \mathbb{N}^2$. Let $V_b = [\![0, v_b - 1]\!] \subset \mathbb{N}$. A (monovalue) image is a function $\mathcal{I}_b : E \to V_b$ which to each point $\mathbf{x} = (x, y) \in E$ of the scene, associates a spectral intensity $\mathcal{I}_b(\mathbf{x}) = v$.

Let $V = \prod_{b=1}^{s} V_b \subset \mathbb{N}^s$ $(s \geq 2)$. A (multivalue) image is a function $\mathcal{I} : E \to V$ which to each point $\mathbf{x} = (x, y) \in E$ associates $\mathcal{I}(\mathbf{x}) = \mathbf{v} = \prod_{b=1}^{s} \mathcal{I}_b(\mathbf{x})$.

A segmentation of an image $\mathcal{I} : E \to V$ is a partition $\mathfrak{S} = \{R_i\}_{i=1}^{n}$ $(n \geq 2)$ of E.

Binary partition tree The Binary Partition Tree (BPT) [**?**] is a hierarchy of regions created by a merging algorithm that can make use of any similarity measure. Starting from a given partition, this algorithm proceeds by (1) computing a similarity measure for all pairs of neighbour regions, (2) merging the most similar pair of regions, and (3) updating the similarity measures (iterating (2,3) until all regions are merged into a single one). The BPT generation then relies on two main notions: the *region model* (which specifies how regions are characterised), and the *merging criterion* (which defines the similarity of neighbouring regions and, thus, the merging order).

Region model A region R_i is modelled here by (1) $M_{R_i}^{Rad} = \langle (v_{b_{min}}^{R_i}, v_{b_{max}}^{R_i}) \rangle_{b=1}^{s}$, where $v_{b_\star}^{R_i}$ are the extremal radiometric values of the b-th radiometric band of $\mathcal{I}$ (*i.e.*, $\mathcal{I}_b$), and (2) $M_{R_i}^{Geo} = (e(R_i), a(R_i))$, where $e(R_i)$ and $a(R_i)$ are the elongation and the area of (R_i), respectively. During the merging process, the region model of two merged regions R_i, R_j is then straightforwardly provided by $M_{R_i \cup R_j}^{Rad} = \langle (\min\{v_{b_{min}}^{R_i}, v_{b_{min}}^{R_j}\}, \max\{v_{b_{max}}^{R_i}, v_{b_{max}}^{R_j}\}) \rangle_{b=1}^{s}$ and $M_{R_i \cup R_j}^{Geo} = (e(R_i \cup R_j), a(R_i) + a(R_j))$.

Merging criterion The basic merging criterion used in most of image segmentation approaches is radiometric homogeneity. Here, we propose to rely on both the increase of the ranges of the intensity values (for each radiometric band) and on area and elongation of the regions in order to merge in priority objects which do not structure the scene:

$$O^{Rad}(R_i, R_j) = \frac{1}{s} \sum_{b=1}^{s} |\max(v_{b_{max}}^{R_i}, v_{b_{max}}^{R_j}) - \min(v_{b_{min}}^{R_i}, v_{b_{min}}^{R_j})| \qquad (1)$$

$$O^{Geo}(R_i, R_j) = \frac{1}{2}(e(R_i \cup R_j) + a(R_i \cup R_j)) \qquad (2)$$

The similarity measure between two neighbouring regions R_i and R_j is then

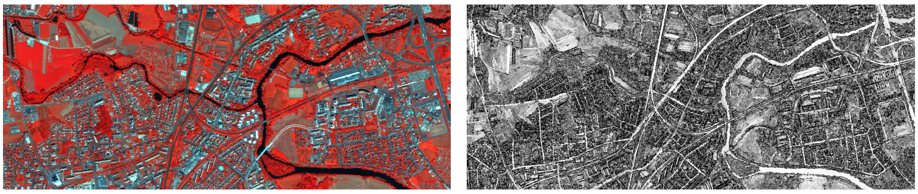
$$O(R_i, R_j) = \alpha.O^{Rad}(R_i, R_j) + (1-\alpha).O^{Geo}(R_i, R_j) \text{ with } \alpha \in [0,1] \qquad (3)$$

In practice, the closest the nodes are to the root, the less relevant O^{Rad} is. Consequently, the weight α can be defined as a function depending directly on the value of O^{Rad} (and decreasing when O^{Rad} increases). In particular, we have experimentally set $\alpha = \exp(-(O^{Rad})^2)$.

Elongation map The proposed elongation map characterises the linear structures (roads, rivers, railways, etc.) which generally divide an urban scene into (large) regions. Our purpose, here, is not to get the best elongation results, but to be able to compute correct elongations with a low computational cost. This map is computed as follows:

(1) for each pixel (considered as a seed), a series of region-growing segmentations (based on radiometric intensity) is performed with an increasing tolerance;
(2) for each segmentation, a score is computed using the ratio width/length of the best bounding box of the region (computed in several orientations);
(3) for each pixel, the best (*i.e.*, the lowest) elongation value is assigned.

This approach presents an algorithmic cost bounded, for each pixel, by the area of the neighbourhood where Step (1) is carried out (which, in practice, needs not to be high). The computation of the elongation map is then globally linear with respect to the size of E. Figure 4 provides an example of an elongation map computed on a HSR image with a spatial resolution of 2.4 m and obtained thanks to this heuristic strategy.



(a) HSR image (b) Elongation map

Fig. 4. Elongation map computation. (a) HSR image with a spatial resolution of 2.4 m (QuickBird, © DigitalGlobe Inc.). (b) Corresponding elongation map (elongated structures in light grey, non-elongated ones in dark grey).

Automatic tree-cut learning Once the BPT built, a cut has to be chosen through the hierarchy. This can be done by performing a thresholding on the similarity measure (called *energy* in the sequel) related to the O function in the BPT (Equation (3)). Broadly speaking, to each node of the tree an energy is attached (by saving for a node R_i, the value of O required to create this node). A cut can then be easily extracted by determining an adequate energy in a threshold-like fashion, thus preserving the subtree formed by the nodes of higher value (with a possible refinement of one or several branches). This first (manual) step is performed on *one input image* among the k available ones (in red in Figure 1). It is then possible to automatically reproduce this tree-cut operation to segment the $k-1$ other images. This is done by performing a similar tree-cut (*i.e.*, with a same energy) in each one of the $k-1$ corresponding BPTs (in blue in Figure 1).

Without loss of generality, this approach can be applied on k images defined on connected subsets of $\mathbb{N}^2$ (and not only rectangular ones). As a consequence, it enables to segment several sub-parts of a same image. In particular, this is the way it is used in the multiresolution approach described hereafter.

3.2 Multiresolution strategy

The proposed multiresolution strategy is dedicated to hierarchically segment n images of a same scene at various resolutions, from the lowest to the highest one. In the classical case, three images are considered, namely a MSR (30–5m), a HSR (3–1m) and a VHSR (less than 1m) image. The strategy (Section 3.2) combines iteratively the previously described interactive segmentation approach (Section 3.1) and a multiresolution clustering method [**?**] (Section 3.2). At each resolution/step, the output of this process (a segmentation map) is embedded in the next resolution image (using a correspondence map function) to be treated as input of the next step. This process is iterated n times (one per image, from low to high resolution). It provides as final output n segmentations (one per considered image/resolution), hierarchically linked, enabling different scales of interpretation.

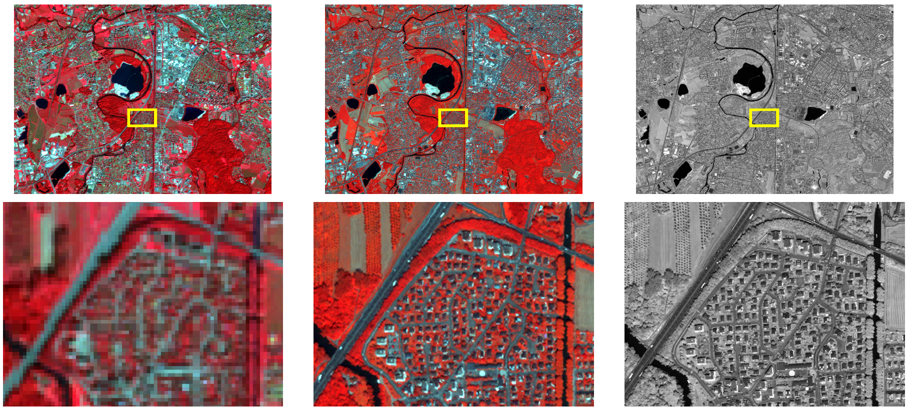
Partitioning strategy at a given resolution At each resolution, the following partitioning strategy is applied. It takes as input one (or more) family(ies) of subimages (with the same semantics). (For instance, in Figure 2, the input used to process the image 2, is composed of three semantic families: the yellow family (composed of two regions of urban areas), the green family (composed of four regions of urban vegetation), and the blue one (composed of one region of water).) First, each family is processed, individually, by the interactive segmentation method (Figure 2-①) generating a segmentation result. (For each family, the user has only to provide one example of tree-cut for a selected region (Section 3.1)). These segmentation results are then combined to produce a whole segmentation map. Then, regions provided by this segmentation are gathered into different clusters using a multiresolution clustering method (Figure 2-② and Section 3.2) taking as input these regions and the next resolution image. Finally, these classified segments are embedded in the next resolution (Figure 2-③), forming, for each resulting class, a new family of subimages which can be processed by following the same strategy.

Multiresolution clustering [?] This approach takes as input two multivalue images ($\mathcal{I}^1 : E^1 \to V^1$ and $\mathcal{I}^2 : E^2 \to V^2$) of the same scene. The main idea is to fuse the information provided by (1) the analysis of the lowest resolution regions of $\mathcal{I}^1$ (obtained by a segmentation, Figure 3-①) and (2) the highest resolution semantic clustering of $\mathcal{I}^2$ (provided by a classical clustering method directly applied on the radiometric values of the pixels, Figure 3-②), to obtain a final clustering result corresponding to an intermediate level. For each region R_i of the lowest resolution segmented image, an histogram is computed (Figure 3-③) taking into account the distribution of the pixels of R_i in terms of clusters in the highest resolution clustered image. The final clustering result is computed by classifying (in an unsupervised way, Figure 3-④) the regions of the lowest resolution segmented image using their composition histograms. The method finally provides as output a clustering image at an intermediate semantic level (*i.e.*, a level corresponding to a resolution between the ones of $\mathcal{I}^1$ and $\mathcal{I}^2$). This clustering is modelled by a label image $\mathcal{R} : E^1 \to [\![1, k]\!]$ which, to each point $\mathbf{x}$ of the scene (at the lowest resolution), associates a class value $\mathcal{R}(\mathbf{x})$ among the k possible ones. For a more detailed and formalized description of this approach, the reader may refer to [?].

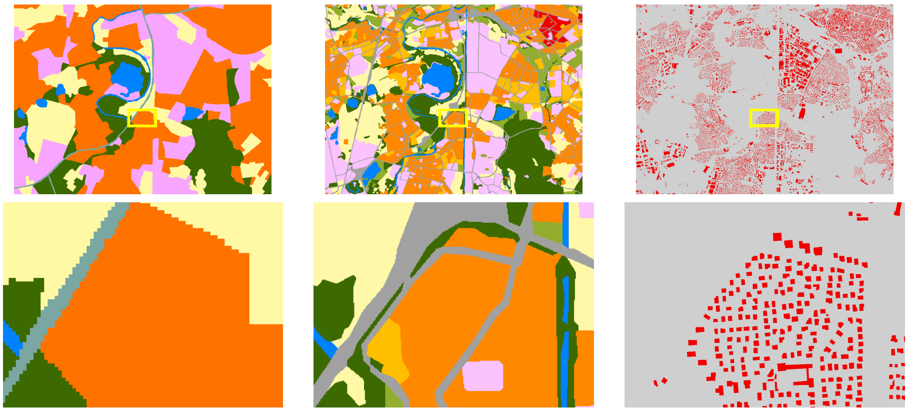
4 Experiments and Results

4.1 Experiments

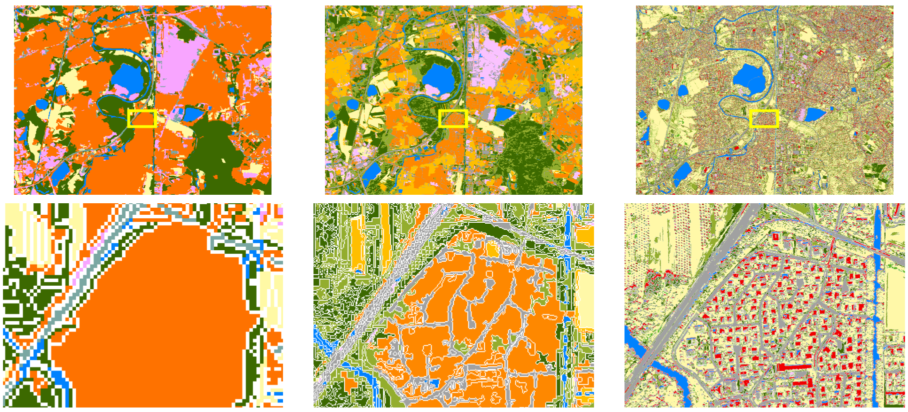
Experiments have been performed on a multiresolution set of satellite images, presenting a part of the city of Strasbourg, France. These images present a typical suburban environment with water surfaces, forest areas, industrial areas, individual/collective housing blocks and agricultural zones. This set is composed by (1) a single Spot-5 MSR (9.6m) multispectral image (Figure 5(a), 685×583



(a) 1 pixel = 9.6m × 9.6m (b) 1 pixel = 2.4m × 2.4m (c) 1 pixel = 60cm × 60cm



(d) Districts level. (e) Block level. (f) Houses level.



(g) MSR segmentation. (h) HSR segmentation. (i) VHSR segmentation.

Fig. 5. (a–c) Satellite images of the same area (6 500m × 5 400m) with different spatial resolutions: (a) MSR (© CNES–ISIS program), (b) HSR, (c) VHSR (© DigitalGlobe Inc.). (d–f) Corresponding ground-truth maps. (g–i) Obtained segmentations: (g) MSR segmentation, (h) HSR segmentation, (i) VHSR segmentation. A zoom (× 15) on a same given area (the yellow one) is proposed for each image.

pixels) and by (2) a couple of QuickBird images composed by a HSR (2.4m) multispectral image (Figure 5(b), 2 740 $\times$ 2 332 pixels) and a VHSR (60cm) panchromatic one (Figure 5(c), 10 960 $\times$ 9 328 pixels).

To assess the efficiency of the proposed approach, several tests have been performed to help end-users to extract a hierarchy of complex urban patterns (urban districts, urban blocks, urban objects) from these data. At each step/resolution, the obtained regions have been classified (Figure 5(g–i)) in order to compare them to certified ground-truth maps (Figure 5(d–f)) using the Kappa index.

4.2 Results

Step 1 has been applied on the MSR image to separate the largest structures of the scene (*e.g.*, urban districts, forest areas, water surfaces, etc.). After classification, the comparison between the classified resulting regions (Figure 5(g)) and the ground-truth map has shown a Kappa value of 0.77.

Step 2 has been applied on the HSR image to split these urban districts into different large regions corresponding to: mixed urban districts, commercial or industrial sub-districts, housing blocks, etc. After classification of the resulting regions (Figure 5(h)), the Kappa value for the resulting partition was 0.81.

Step 3 has been performed on the VHSR image to extract "basic" urban objects (*e.g.*, individual/collective houses, vegetations, streets/car parks, shadows, etc.) from these urban blocks. Due to the unavailability of all the "class" information for the ground-truth map corresponding to the VHSR image (only available for the building class, Figure 5(f)), the Kappa index has not been computed. However, after classification of the resulting regions (Figure 5(i)), the percentage of pixels (from the red cluster) matching with the building class was 84% and the percentages of false negatives and false positives were 7% and 16%.

4.3 Runtime

Table 1 provides the runtime and the memory usages for the segmentation of the images. Experiments have been run on an Intel® Core$^{\text{TM}}$2 Quad running at 2.4 GHz with 8 GB of RAM.

As shown by the second column of Table 1, the extraction process is linear with the size of the images. For instance, a HSR image which contains 16 times more pixels than a MSR one, requires 16 times more operations and time to be processed than a MSR one. Since the multiresolution clustering approach (which is mainly based on a partitioning clustering) is linear through the data, we can assume that the interactive segmentation approach is also linear through the data. However, one can see that the memory consumption remains significant when processing the VHSR images (third column of Table 1).

5 Discussion and Perspectives

This article has presented an interactive hierarchical segmentation approach based on binary partition trees and the first results obtained with this method-

Table 1. Runtime and memory usage for the segmentation of the multiresolution images.

Image (Size - pixels)	Extract Runtime	Memory (RAM)
MSR (685 $\times$ 583)	23.8 s	54 MB
HSR (2 740 $\times$ 2 332)	5 min 49 s	418 MB
VHSR (10 960 $\times$ 9 328)	1 h 13 min	2.47 GB

ology on a multiresolution dataset. Experiments have shown that the quality of the extracted urban patterns seems sufficient to further accurately perform both classification or object detection. This seems to validate (1) the relevance of the proposed method and (2) the soundness of the semi-automation of the photo-interpretation approach. However, one can observe that some of the partition results are composed of several regions matching with urban patterns and numerous tiny regions forming linear structures and covering vegetation areas (in particular for the MSR image, Figure 5(g)). These over-segmentation problems are probably due to the spatial criteria used by the algorithm (the elongation one) which does not consider the vegetation areas.

A main advantage of this method is to be parameter-free. Indeed, the only significant parameter is actually the level of tree-cut, the effect of which can be visually assessed by the user. Due to the pre-processing of the data structures, the short computation times enable, in particular, to carry out several segmentations to select the best one.

However, a weakness of this method is that the automation of the interactive tree-cut approach is currently only based on the global energy of the nodes. The next step of this work will then consist of finding a more robust way to reproduce automatically the tree-cut approach in other images. In order to do so, it could be possible to perform nodes matching based on nodes properties and structures.

In a next issue, the results obtained with this method will be fully assessed by quantitative comparisons (using datasets provided by different sensors) and will be compared to the results produced by other hierarchical region-based methods.