Stable Mean-Shift Algorithm and Its Application to the Segmentation of Arbitrarily Large Remote Sensing Images

Julien Michel, *Member, IEEE*, David Youssefi, and Manuel Grizonnet

***Abstract*—Segmentation of real-world remote sensing images is challenging because of the large size of those data, particularly for very high resolution imagery. However, a lot of high-level remote sensing methods rely on segmentation at some point and are therefore difficult to assess at full image scale, for real remote sensing applications. In this paper, we define a new property called stability of segmentation algorithms and demonstrate that piece- or tile-wise computation of a stable segmentation algorithm can be achieved with identical results with respect to processing the whole image at once. We also derive a technique to empirically estimate the stability of a given segmentation algorithm and apply it to four different algorithms. Among those algorithms, the mean-shift algorithm is found to be quite unstable. We propose a modified version of this algorithm enforcing its stability and thus allowing for tile-wise computation with identical results. Finally, we present results of this method and discuss the various trends and applications.**

***Index Terms*—Image processing, image segmentation, mean-shift, object-based image analysis, remote sensing, stability of segmentation.**

I. Introduction

FOR the past decade, the number of very high resolution (VHR) optical sensors orbiting around the Earth and imaging it has been constantly increasing. USA satellite programs such as IKONOS, QuickBird, GeoEye, or WorldView, as well as the new French constellation of Pleiades satellites, provide images with a ground sampling distance between 0.41 and 1 m and a field of view between 15 and 20 km. When including ground processing such as pan-sharpening or mosaics, the end users receive tremendously large images whose size can exceed several tenths of thousands of pixels in both dimensions, challenging both their manual and automatic processing capabilities. Meanwhile, the fine spatial resolution of these images drastically increases not only the richness but also the complexity of the information they contain and therefore enforces the need for more complex processing methods and algorithms.

Manuscript received October 22, 2013; revised March 17, 2014; accepted June 3, 2014. Date of publication July 8, 2014; date of current version August 12, 2014.

J. Michel and M. Grizonnet are with the Centre National d'Études Spatiales (French Space Agency), 31400 Toulouse, France.

D. Youssefi was with the Centre National d'Études Spatiales (French Space Agency), 31400 Toulouse, France. He is now with Communications et Systèmes, 31506 Toulouse, France.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2014.2330857

Indeed, it is well accepted that for such spatial resolutions, many objects of interest are composed of a large number pixels with high heterogeneity, and that pixel-based and even neighborhood-based image analysis techniques fail to capture this richness due to their limited perception of the spatial context of each pixel. Therefore, techniques such as object-based image analysis [1], object-based image classification, spatial reasoning [2], [3], or geospatial analysis have been extensively studied during the last years and proven of great interest for the analysis of VHR optical images, for instance. Most of these methods perform some kind of segmentation, either as a preprocessing step or within the processing itself.

However, for VHR optical images delivered by sensors cited earlier, as well as for VHR SAR delivered by TerraSAR or CosmoSkymed sensors, and even for images from the upcoming Sentinel2 sensor, with lower resolution but larger swath, the traditional data processing paradigm hardly applies. Indeed, loading the whole data into memory, processing it, and retrieving the results from memory can turn impossible because the amount of memory needed to consume the input data volume exceeds the available resources. The processing of such large data or large images can be achieved by dividing the input data into pieces such as tiles (a tile being a rectangular image subset), processing each tile sequentially [4], and gathering the final output. For some kind of image operations, such as pixel- or neighborhood-wise filtering, this methodology can be applied with guaranteed identical results with respect to processing the whole image at once. However, most segmentation algorithms do not cope well with piece- or tile-wise computation because they consider long-range interactions between pixels. As a result, the aforementioned advanced image analysis techniques have mostly been investigated at the scale of small extracts and can hardly apply to real-life remote sensing applications with real data. A few techniques have been proposed to tackle this issue and are reviewed in Section II. These techniques do not provide any guarantee regarding the exactitude of the result. In particular, they might spawn artifacts in the premises of tiles border because of the convergence of the segmentation algorithm toward incompatible solution on each side of this border.

The work presented in this paper consists in two main contributions. First, we define a new property of segmentation algorithms called stability, as well as an empirical method to measure the adequacy of a segmentation algorithm with this property. We also show that stable segmentation algorithms can be used for tile-wise segmentation with guaranteed exact results, following our proposed methodology. Second, we propose a stable segmentation algorithm derived from the mean-shift algorithm [5], which allows combining the accuracy of the mean-shift algorithm with the benefit brought by the stability property, particularly regarding large images tile-wise processing. It is noteworthy that the execution of this methodology in a parallel or distributed environment is straightforward since each step involves independent operations on image tiles that are clearly isolated. Ultimately, this methodology provides an artifact-free segmentation result that can be used to assess object-based methods at the scale of full VHR data.

The remaining of this paper is organized as follows. In Section II, we present an overview of existing work. Section III will introduce the concept of segmentation stability. It will also describe a methodology to measure this stability and present the measured stability of four different segmentation algorithms. Section IV will investigate the instability sources of the mean-shift segmentation algorithm and propose a stabilized version. In Section V, we then describe how the guaranteed stability of a segmentation algorithm leads to a trivial solution for tile-wise segmentation, yielding identical results to those obtained by processing the whole image at once. Finally, Section VI will present some segmentation results on Pleiades images.

II. Previous Work

Despite its real interest for operational applications, the use of a tiling scheme for the segmentation of very large images has been merely investigated in the literature.

In [6], the authors propose to combine the full lambda schedule algorithm (FLSA) with such a tiling scheme. They apply the algorithm on each tile and consider segments entirely included in an overlapping zone between tiles as not reliable. Pixels belonging to these segments are reinitialized to one-pixel segments and considered in the segmentation of the next tile. Although applied to the FLSA, the authors advertise that this method can apply to any segmentation algorithm based on segment merging. If this method is able to process correctly small segments within the tile overlap zone, larger segments not entirely included in this zone are not considered and may therefore exhibit artificial borders induced by the tiling scheme.

In [7], the authors propose a method for the detection of road networks on satellite images, based on segmentation. The normalized cuts algorithm of Shi and Malik is first applied to obtain an oversegmentation of the image, and segments are then merged into bigger segments, from which roads are detected. In their experiment, the image is divided into 200×200 pixel tiles, and the segmentation is applied to each tile independently. The merging step is applied afterward to all segments from all tiles, with no specific rule for segments on the tile border, which leads to some artificial borders in the final result.

In his work on a topological and hierarchical model for the segmentation of large images [8], Goffe separates the parts of segment borders that are artificially created by tiling or segmentation errors from the parts of segment borders that correspond to real image content. The criterion to detect the artificial borders shall denote the similitude of the neighboring segments they separate. The author proposes a criterion based on colorimetric analysis. However, it cannot be ensured that all tiles border are properly identified as artificial borders.

Some methods have been proposed to specifically process the artifacts created by tiled segmentation. In [9], a framework for the segmentation of large images is described. Segmentation is performed independently on non-overlapping tiles, and the FLSA algorithm is used to merge segments having borders in common with tiles, processing vertical tile borders first, and then horizontal ones. Merging stops when a minimal cost is reached. In [10], Happ *et al.* proposes a method for parallel segmentation of large images, with a segmentation method also relying on iterative segment merging. One iteration of the algorithm is performed independently on each tile, possibly in parallel, without processing segments that touch the tile border. Once this iteration is done for all tiles, a single segmentation task processes segments on borders. This scheme iterates until there is no possible merging left. Neither methods can guarantee exact results with respect to the same processing applied without tiling.

In [11], the authors propose a parallel implementation of a segmentation method based on minimum spanning trees by dividing the image into non-overlapping tiles. The authors advertise that the results are consistent whatever the number of parallel threads is, but no details are given on how this consistency is ensured.

In [12], instead of using rectangular tiles, the tiles are adapted to the image contour, with the hope that the tile border will coincide with actual segments and will therefore not create any artificial segment shape. However, it cannot be guaranteed that the tile borders adapted with respect to the strength of the local gradient will match the decision of a given segmentation algorithm.

In [13], the author proposes a modification of the recursive hierarchical segmentation algorithm, in which pixels or segments that have been inappropriately merged during the segmentation process are identified, split, and re-merged with a better candidate segment. According to the author, this modified algorithm provides hierarchical segmentations of images that are free of processing window artifacts.

In a previous work, Michel *et al.* proposed a generic tool for tiled segmentation, vectorization, and stitching of segments on the tile border based on a simple topological criterion [14]. While theoretically compatible with any segmentation algorithms, the stitching criterion is very coarse, and many artifacts remain in the final result.

III. Segmentation Algorithms Stability

A. Terms and Notations

In this paper, I denotes an image, and $S : I \to S(I)$ represents a segmentation algorithm. $S(I)$ forms a partition of image I, i.e.,

$$S(I) = \{R_i, i \in [1, n]\}, \begin{cases} I = \bigcup_{i=1}^{n} R_i \\ i \neq j \Rightarrow R_i \cap R_j = \emptyset \end{cases} \quad (1)$$

where R denotes an element of $S(I)$ and will be referred as a segment throughout this paper. Whenever needed, a collection of segments will be indexed by a subscript R_i.

$I' \subset I$ denotes an image subset and will be called likewise in this paper. Note that when dealing with implementation, the term image tile will also be used. It denotes a rectangular and axis-aligned image subset of I. A segment from an image subset I' is represented by $R' \in S(I')$.

Finally, for a segment $R \in S(I)$ and an image subset $I' \subset I$, we define $S_R(I')$ as follows:

$$S_R(I') = \{R' \in S(I') \setminus R' \subseteq R\}. \tag{2}$$

$S_R(I')$ is the set of segments $R' \in S(I')$ that are fully included in segment $R \in S(I)$.

B. Definition of Stability

With those notations, we can give a formal definition of the stability property.

Definition 1: Algorithm S is said to be stable if $\forall R \in S(I)$ and $\forall\, I' \subset I$, the following properties hold:

$$R \subset I' \Rightarrow \exists R' \in S(I') \setminus R' = R \tag{3}$$

$$R \cap I' \neq \emptyset \Rightarrow R \cap I' = \bigcup_{R' \in S_R(I')} R'. \tag{4}$$

We call property (3) the *inner property*. It states that any segment of $S(I)$ inner to I' is also a segment of $S(I')$. Property (4) is called the *cover property*. It states that the restriction to image subset I' of any segment of $S(I)$ overlapping I' must decompose into a set of segments from $S(I')$.

More intuitively, the stability of a segmentation algorithm implies the following.

- Given an image, given an object of interest within this image, a segmentation algorithm is considered stable if any image subset containing this object spawns the exact same segmentation result of this object, or alternatively,
- Given an image, given two overlapping subsets of this image, a segmentation algorithm is considered stable if the segmentation results in the intersection of these two image subsets match exactly.

C. Benefits of Stability

Stability is a much desirable property for remote sensing image segmentation. It is obvious that using unstable segmentation with tiling will lead to different results depending on the tiling scheme, and Section V will show how guaranteed stability leads to an exact solution for tiled segmentation. However, even without considering tiling, identical objects with identical backgrounds located at different places in an image might be segmented very differently, apart from any consideration of lightning, sampling, noise, or small changes. For instance, using an unstable algorithm may be sufficient to fail segmenting coherently a whole set of identical buildings in an image (cf. Fig. 1).

While there have been several studies comparing segmentation algorithms and software with various criteria such as accuracy with respect to ground truth, it is surprising that only little attention has been given to this stability property. In [15], a comparison of segmentation software is presented, and among a lot of other properties, the reproducibility when image size is modified is evaluated, which is somehow similar to the notion of stability. However, among the seven software inquired, only eCognition is considered reproducible starting version 3.0, and there is no detail on how this reproducibility has been evaluated, which is not the focus of this paper.

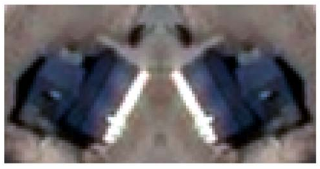
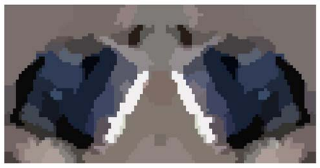
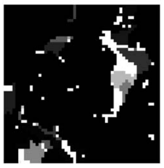
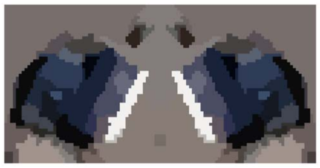
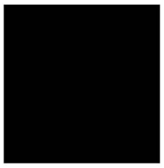
	Clustering	Difference (left-right)
Initial image		
Edison Mean-Shift algorithm		
Stabilized Mean-Shift algorithm		

Fig. 1. Segmentation of two identical houses in an image: The initial image is created by merging a 54 × 55 pixels image of a house and its horizontally flipped version. After segmenting this image with the Edison mean-shift algorithm and the stabilized mean-shift algorithm, the left part of the segmentation is flipped back and subtracted to the segmentation of the right part. Unlike the stabilized mean-shift algorithm, differences appear with the Edison mean-shift algorithm.

D. Measuring Stability

1) Comparing Segmentation Results: Evaluating quantitatively the stability of a segmentation algorithm requires to be able to compare different segmentation results of the same image. For this purpose, there is a number of quantitative metrics available in the literature. A review of unsupervised metrics is available in [16], whereas supervised metrics relying on a ground truth or reference segmentation, which are more suitable for our work, have been compared in [17] and [18]. Among the different proposed methods, we chose to use the Hoover *et al.* method [19] and its quantitative extension proposed by Ortiz *et al.* [20]. In addition to being based on the same set theory concepts than our definition of stability, this method provides both quantitative measures and segment-based classification of the type of error. Moreover, it uses thresholds that allow tuning the tolerance of the metric and error classification with respect to the aim of the experiment. Finally, even if sometimes criticized, it is cited and used as a reference metric in most papers proposing new segmentation evaluation metrics. Note that the sensitivity to the distortion issue raised in [18] is not relevant in our work since we analyze the quantitative scores of each segment and not only the count of correctly detected segments.

The Hoover method matches one or several segments from the ground truth segmentation with one or several segments from the tested segmentation, and each match is called a Hoover instance. Given two partitions of the same image $S^1(I)$ and $S^2(I)$, Hoover instances are based on the confusion matrix O, which is defined as

$$O_{ij} = \left|R_i^1 \cap R_j^2\right| \text{ for } \left(R_i^1, R_j^2\right) \in S^1(I) \times S^2(I). \tag{5}$$

Let t denote an overlap threshold $0.5 < t \leq 1$. The Hoover method classifies segments from $S^1(I)$ and $S^2(I)$ into four categories. A pair of $(R_i^1, R_j^2) \in S^1(I) \times S^2(I)$ is considered as a match if

$$\begin{cases} O_{ij} \geq t \times \left|R_i^1\right| \\ O_{ij} \geq t \times \left|R_j^2\right|. \end{cases} \tag{6}$$

In this case, the RC metric from Ortiz measures the match performance as

$$RC\left(R_i^1, R_j^2, t\right) = \min\left\{\frac{O_{ij}}{|R_i^1|}, \frac{O_{ij}}{|R_j^2|}\right\}. \tag{7}$$

A segment $R_i^1 \in S^1(I)$ is considered as fragmented by a set $R_\star^2$ of $m \geq 2$ segments $R_\star^2 = \{R_{j_k}^2 \in S^2(I), k \in [1, m]\}$ if

$$\begin{cases} \forall k \in [1, m], \quad O_{ij_k} \geq t \times \left|R_{j_k}^2\right| \\ \qquad\qquad \sum_{k=1}^{m} O_{ij_k} \geq t \times \left|R_i^1\right|. \end{cases} \tag{8}$$

In this case, the RF metric from Ortiz measures the amount of fragmentation as

$$RF\left(R_i^1, R_\star^2, t\right) = 1 - \frac{\sum_{k=1}^{m} \left(O_{ij_k}\left(O_{ij_k} - 1\right)\right)}{|R_i^1| \times \left(|R_i^1| - 1\right)}. \tag{9}$$

In a similar way, a segment $R_j^2 \in S^2(I)$ is considered as fragmented by a set $R_\star^1$ of $m' \geq 2$ segments $R_\star^1 = \{R_{i_k} \in S^1(I), k \in [1, m']\}$ if

$$\begin{cases} \forall k \in [1, m'], \quad O_{i_k j} \geq t \times \left|R_{i_k}^1\right| \\ \qquad\qquad \sum_{k=1}^{m'} O_{i_k j} \geq t \times \left|R_j^2\right|. \end{cases} \tag{10}$$

In this case, the RA metric from Ortiz measures the amount of grouping as

$$RA\left(R_\star^1, R_j^2, t\right) = 1 - \frac{\sum_{k=1}^{m'} O_{i_k j}\left(O_{i_k j} - 1\right)}{\left|\cup_{k=1}^{m'} R_{i_k}^1\right| \times \left(\left|\cup_{k=1}^{m'} R_{i_k}^1\right| - 1\right)}. \tag{11}$$

Segments from $S^1(I)$ or $S^2(I)$ that do not belong to any of the three previous categories are considered as not detected. In this case, Ortiz defines the missed measure $RM(t) = 1$.

Global metrics can then be defined for the whole image as the mean of metrics values weighted by segment sizes.

In the remaining of this paper, we adopt the following color conventions to display Hoover instances images. Given a score threshold t_s

- if $RC \geq t_s$, the segment is green (correct detection);
- if $RC < t_s$ and $RF > 0$, the segment is yellow (oversegmentation);
- if $RC < t_s$ and $RA > 0$, the segment is purple (undersegmentation);
- if $0 < RC < t_s$, $RF = 0$, $RA = 0$ or if $RM > 0$, the segment is red (missed detection);
- background is black (if applicable).

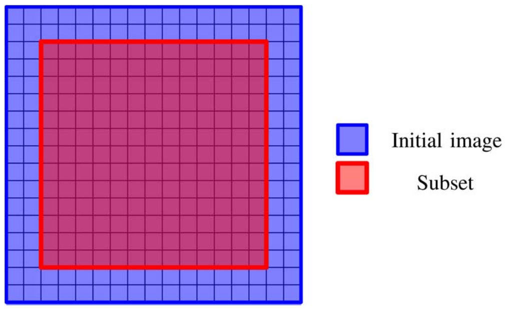


Fig. 2. Initial image and subset.

In all experiments, the threshold t to build Hoover instances is set to 0.75, whereas the threshold t_s to display the instances is adapted to each experiment.

2) Protocol for Stability Evaluation: In order to verify the stability of the segmentation algorithms, the influence of the tiling is simulated by reducing or changing the image subset observed by these algorithms. The general protocol is the following one.

First, an image is segmented, and the resulting segmentation is considered as the reference segmentation.

Second, a subset of the image is selected to create the compared segmentation (cf. Fig. 2). Finally, the two segmentations are restricted by set intersection to the image subset and compared with the Hoover method (cf. Section III-D1).

3) Results: The protocol for stability evaluation has been applied to the mean-shift segmentation algorithm, the watershed algorithm applied to the image gradient magnitude, and the connected-component algorithm [21] using a threshold on the spectral distance between neighboring pixels as the connection criterion. The watershed algorithm implementation is provided by the ITK software [22], which performs a top-down gradient descent toward local minima, as opposed to the bottom-up watershed strategy starting from seeds at local minima of the height function. Two implementations of the mean-shift algorithm have been considered: the original implementation from Comaniciu *et al.* [5] distributed as the Edison software, and the multithreaded implementation from the Orfeo ToolBox [23].

Please note that in this experiment, the intrinsic performances are not compared: the aim is only the stability evaluation. Hence, the parameters used for these experiments were arbitrarily chosen and kept identical for both the entire image and the subset segmentations. Although we cannot guarantee that results will not change with a different parameter setup, we did not observe such changes during our experiments.

Fig. 3 shows the comparison between the reference segmentation created with a 300 × 300 pixels extract from a Pleiades scene over Paris and the compared segmentations created with the same extract where the 1-pixel-wide external crown of pixels is removed. The threshold for the display of the

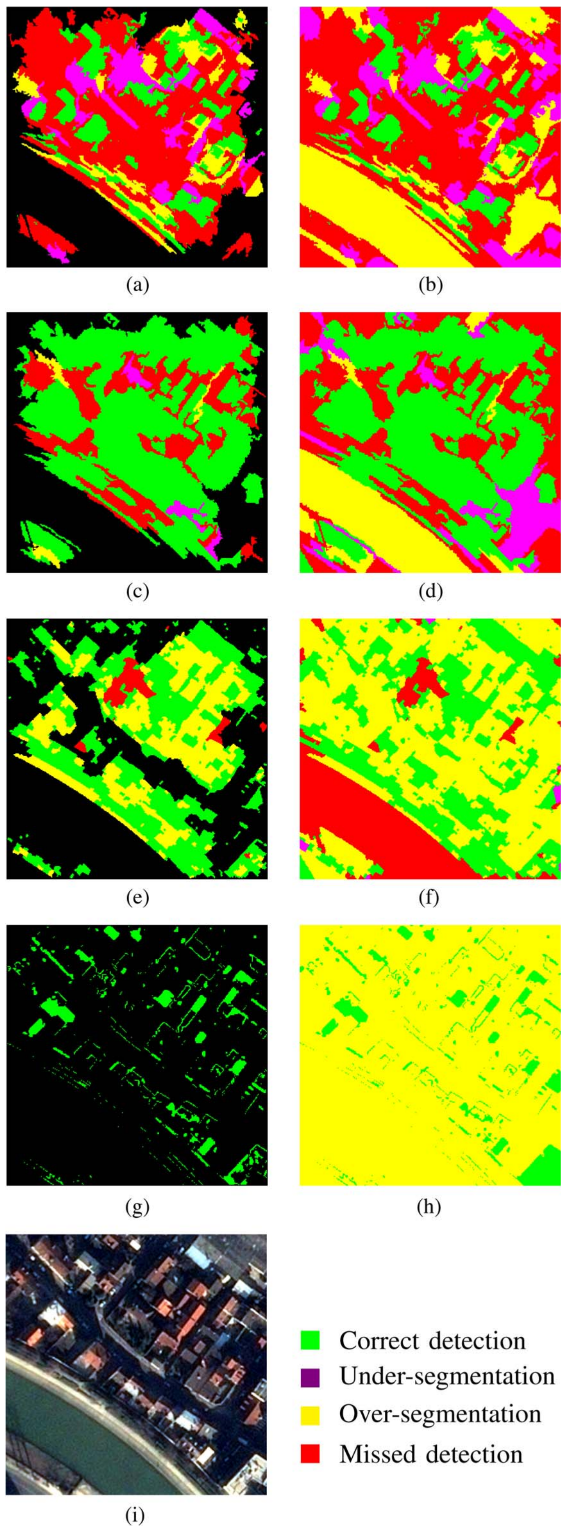


Fig. 3. Comparison between the reference segmentation and the segmentation of a Pleiades scene extract over Paris, France, in the case where the 1-pixel-wide border is deleted. Because of the *cover property*, algorithms with good stability will exhibit both green (RC) and yellow (RF) segments. (a) Mean shift (Edison, case $R \subset I'$). (b) Mean shift (Edison, case $R \cap I' \neq \emptyset$). (c) Mean shift (OTB, case $R \subset I'$). (d) Mean shift (OTB, case $R \cap I' \neq \emptyset$). (e) Watershed (case $R \subset I'$). (f) Watershed (case $R \cap I' \neq \emptyset$). (g) Connected components (case $R \subset I'$). (h) Connected components (case $R \cap I' \neq \emptyset$). (i) Initial image.

Hoover instance has been set to $t_s = 1$ to ensure that the green segment will only correspond to exact matches. The figure also distinguishes between the $R \subset I'$ case, where only segments

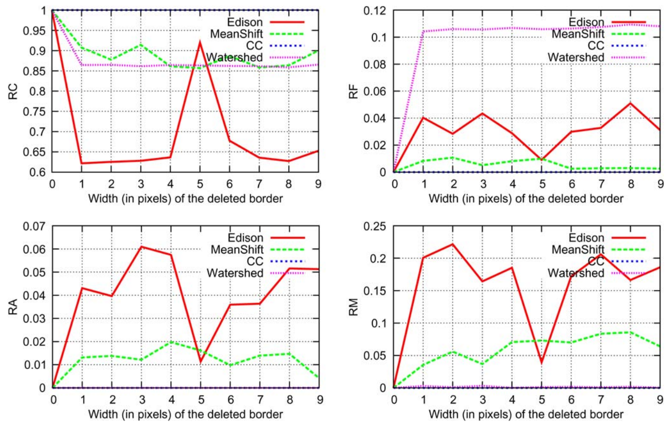


Fig. 4. Ortiz scores according to image subset shrinking (case $R \subset I'$).

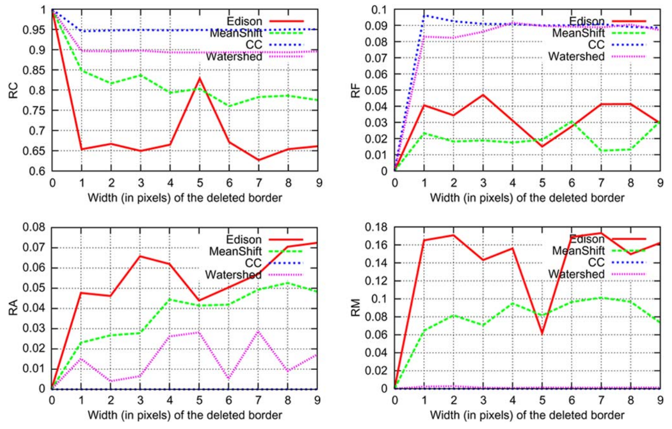


Fig. 5. Ortiz scores according to image subset shrinking (case $R \cap I' \neq \emptyset$).

that do not touch the image borders are considered, and the $R \cap I' \neq \emptyset$ case, where all segments are considered.

We can see that only the connected-component algorithm seems to comply with the stability criteria: in the $R \subset I'$ case, all segments are correctly detected (in green), whereas in the $R \cap I' \neq \emptyset$ case, all segments are either correctly detected or oversegmented (in yellow). In contrast, in the case of the mean-shift algorithms and the watershed algorithm, some segments are undersegmented (in purple) or even not detected (in red), even in the $R \subset I'$ case.

Figs. 4 and 5 show the Ortiz scores, computed between 0 (if no segment belongs to this kind of instance) and 1, with respect to the size of the segmented image subset. In Fig. 4, segments touching the edges are excluded, whereas in Fig. 5, all segments are considered.

We can see that the instability of the mean-shift algorithms and the watershed algorithm is confirmed: both mean-shift implementations show an erratic behavior, and if the watershed algorithm curves are smoother, they exhibit instability nonetheless. Regardless of the edges, the processed subset shrinking induces variations of the correctly detected metric (RC).

Moreover, the stability of the connected-component algorithm is confirmed: when segments touching the edges are excluded, all segments are correctly detected $(RC = 1)$, which shows that the algorithm complies with the *inner property* (3). When edge segments are included, the correct detection decreases in favor of the oversegmentation, which shows that the *cover property* (4) is respected. Thus, this algorithm respects the stability definition.

IV. Stabilizing the Mean-Shift Algorithm

A. Why the Mean-Shift Algorithm?

A wide range of segmentation algorithms has been applied to remote sensing images in the literature. A complete review of those algorithms is beyond the scope of this paper. Some elements can be found in [24].

The mean-shift algorithm has received a lot of attention from the remote sensing community [25]–[27]. Variants such as variable-bandwidth mean shift [28], medoid shift, and quick shift [29] have also been investigated in this context. It has been found to perform well with various remote sensing data ranging from medium to very high resolution and in various applications. Its multivariate nature, as the simplicity of the filtering step, and the availability of various implementations, among which one is from the authors themselves [30], are some of the keys to this popularity. We have already been using this algorithm for various works [31]–[33], and the need to resolve its instabilities and to allow for scalability to real remote sensing data has been driven by our applications.

B. Overview of the Algorithm

The mean-shift algorithm is not really a segmentation algorithm by itself. It is a nonparametric method first introduced by Fukunaga and Hostetler [34] in 1975 for the estimation of modes in a multivariate density of probability function. In 2002, Commaniciu [5] proposed a spatial extension of the algorithm by applying the mode estimation to the joint spatial and spectral domain.

Here, is the outline of the algorithm. Let $\mathbf{x}_i$ and $\mathbf{z}_i$, $i = 1, \ldots, n$, denote pixels from the input and output joint-domain image. For each pixel, the following steps are computed:

1) initialize $j = 1$ and $\mathbf{y}_{i,1} = \mathbf{x}_i$;
2) while $j < j_{\max}$ and $\|\mathbf{y}_{i,j} - \mathbf{y}_{i,j+1}\| > t$ do: $\mathbf{y}_{i,j+1} = (\sum_{\mathbf{x}_k \in N(\mathbf{y}_{i,j})} K(\mathbf{x}_k - \mathbf{y}_{i,j})\mathbf{x}_k)/(\sum_{\mathbf{x}_k \in N(\mathbf{y}_{i,j})} K(\mathbf{x}_k - \mathbf{y}_{i,j}))$;
3) set $\mathbf{z}_i = (\mathbf{x}^s_{i,j}, \mathbf{y}^r_{i,j})$

where $\mathbf{y}_{i,j}$ is the current mode estimation for pixel $\mathbf{x}_i$ at step j, $K(\mathbf{x})$ is a kernel function, $N(\mathbf{x})$ is the set of pixels within the spatial range h_s and spectral range h_r of $\mathbf{x}$, superscripts s and r denote the spatial and spectral components of the joint-domain image pixels, $j_{\max}$ denotes the maximum number of iterations, and t denotes the convergence threshold. Although the Edison implementation allows using both Gaussian and uniform kernels, the latter is usually chosen in most applications.

At the end of the process, each pixel is assigned the estimated spectral signature and spatial location of the local mode of the probability density function it belongs to. This can be useful for image denoising, for instance, but is not yet a segmentation of the image. In the remaining of this paper, we will refer to this step as the filtering step and refer to it as $F_{h_r,h_s,j_{\max}}(I)$.

In Commaniciu's paper, segmentation is obtained by grouping together neighboring pixels that converged toward the same spectral and spatial mode. However, because the spatial kernel bandwidth is limited by h_s, and because of the stopping criterion, each pixel converges toward its own estimate of the mode, which may slightly differ even for neighboring pixels belonging to the same object. Therefore, according to this paper, clusters are formed by grouping together all pixels that converged to spatial modes that are closer than h_s and spectral modes that are closer than h_r, which is known as biconnected components in graph theory. In this paper, this step will be referred as the grouping step.

This initial clustering may lead to an important number of very small segments that do not correspond to any meaningful object of the scene. In Comaniciu's paper, segments smaller than a given number of pixels are simply removed from the resulting segmentation, but in the Edison implementation [30], these small segments are merged with the neighboring segment with the closest spectral mode. This merging process is iterated until there are no small segments left or until a maximum number of iterations have been reached. In this paper, these two operations will be referred to as the small segments filtering step.

C. Sources of Instability

In order to propose a stable version of the mean-shift algorithm, we first need to determine the sources of the instability, which can be classified into two categories: algorithmic issues and implementation issues.

1) Algorithmic Issues: We found three main causes preventing the algorithm from being stable: the first occurs during the filtering step, second comes from the grouping step, and the last occurs during small segments processing.

As stated in Section IV-B, during the filtering step, the filtered pixel iteratively moves toward the spatial position of its mode. When filtering an extract of a given image, it may therefore happen that pixels from an image subset converges to a spatial position outside of this subset. In this case, at some point, the algorithm will be lacking pixels within the spatial radius because they do not belong to the extract. As a result, the mode associated with the current pixel will be different than the one that would have been affected with the algorithm seeing the whole image.

The grouping step is also unstable by nature: as stated by graph theory, a pixel might belong to several biconnected components of pixels satisfying the close mode predicate, whereas in image segmentation, a pixel is only associated with a single segment. For this reason, finding clusters of pixels verifying the close mode predicate is an ill-posed problem for which several solutions exist. Fig. 6(a) gives a simple illustration of this issue. The grouping step algorithm used during mean-shift segmentation therefore performs a greedy search that depends on initial conditions and on the order in which pixels are processed. As shown in Fig. 6(b), it is very easy to exhibit a toy example of a group of pixels clustered together into a biconnected component that will be split into several segments if adding more pixel neighbors, which proves that this part of the algorithm does not satisfy the *cover property* (4).

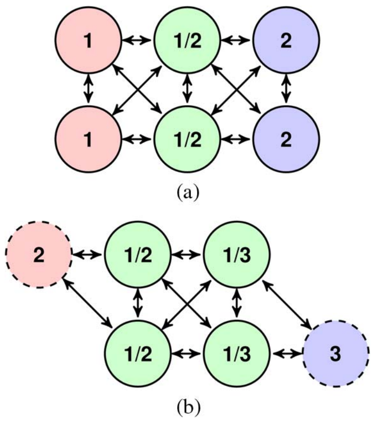


Fig. 6. Illustration of the instabilities yielded by the *close mode* predicate during the clustering step. (a) Simple example of graph with two possible clustering. Green nodes can group either in component 1 with red nodes or in component 2 with blue nodes. (b) Simple example of clustering split by additional nodes. Green nodes are grouped together in component 1, but when adding red and blue nodes, component 1 is split and left green nodes are merged in component 2 while right green nodes go to component 3.

Finally, one more stability issue comes with the processing of small segments, for two reasons. First, small segments lying on the border of the segmentation of an extract of the image cannot be considered as small segments, because they might actually belong to larger segments that have been split by the image subset. Any merging decision for these segments is therefore potentially wrong, but the worst comes when iterating the merging process: those errors on border segments will propagate toward the image center and may cause large errors to happen very far from the image subset border, where we would assume the result to be free from side effects.

2) Implementation Issues: Some implementations of the algorithm come with some optimizations that have a great impact on stability. A common optimization is to stop iterating the algorithm for a given pixel if its spatial trajectory reaches the vicinity of another pixel whose mode has already been estimated, in which case the current pixel is assigned the same mode. This allows running the algorithm faster and also performs some kind of grouping that can be reused for the grouping step, but it is very unstable by nature: the result depends on the order in which pixels are processed, and the modes associated with pixels for which the optimization operates might slightly differ from the modes that would have been estimated for the same pixels without the optimization.

The second instability caused by the implementation is numerical instabilities. It can be seen from Section IV-B that the algorithm implies iterative computation of weighted means. Stability implies that any pixel from any image subset will converge to the exact same mode than with the full image during the filtering step. It is therefore very important for the computation of these weighted means to be numerically invariant to changes of a given pixel spatial position, which will be different in all extracts. It is also very important to avoid accumulating numerical errors during the weighted sums and to avoid multiplying a weighted sum with numerical imprecision by the spatial radius, for instance, since this may lead to amplifying a very small error to a significant one.

D. Proposed Stable Version

Now that we carefully reviewed all sources of instability of the mean-shift algorithm, we propose here to resolve all of them by slightly modifying the algorithm.

To resolve instabilities related to the algorithm itself, which are presented in Section IV-C1, we apply the following improvements. For a given pixel, the spatial position of the estimated mode will not be found farther from this pixel than the product of the number of iterations by the spatial radius. We therefore define a margin of $m = j_{\max} * h_s + 1$ from any image subset border, outside of which we can guarantee that the filtering step will give the exact same results than with the entire image. Equivalently, to guarantee the stability of the filtering step on a given subset, it is sufficient to add a margin of m pixels to this extract. Please note that the pixels from the margin do not need to be processed by the algorithm to get their estimated modes: they only need to be reachable by the algorithm. Second, we relax the biconnected component search of the grouping step into a simple connected-component search, where only neighboring pixels have to satisfy the predicate on spatial and spectral mode distance. Because this relaxed constraint will lead to larger segments by nature, we propose to set the thresholds on spatial and spectral mode distances to values lower than the filtering step bandwidths. Finally, regarding the issue of the small segments filtering step, we simply decide not to do the merging or pruning of small segments, leaving it to further processing according to the user needs.

Regarding the implementation issues, we deactivated any optimization proposed by the implementation to get the closest implementation possible to the mathematical expression of the algorithm in Section IV-B.

With these modifications, we are able to define a segmentation algorithm with guaranteed stability that combines the stabilized filtering mean-shift algorithm and the proven stable connected-component algorithm:

1) Filter image I by the modified mean-shift filtering step with parameter $h_r, h_s, j_{\max}, t$ with margin $m = j_{\max} * h_s + 1$ (except for image borders), to get filtered image $F_{h_r,h_s,j_{\max}}(I)$
2) Segment filtered image $F_{h_r,h_s,j_{\max}}(I)$ into segmentation $S_{h'_s,h'_r}(F_{h_r,h_s,j_{\max}}(I))$ with the connected-component algorithm using a predicate ensuring both spatial modes closer than threshold $h'_s \leq h_s$ and spectral modes closer than threshold $h'_r \leq h_r$.

Fig. 1 shows the comparison between the original mean-shift implementation and the stabilized version. We can observe that none of the segmentation errors from the original algorithm are found with the stabilized version.

V. Stability and Segmentation of Large Images

In this section, we first prove the theoretical basis that allows us to run a stable segmentation algorithm piecewise while ensuring identical results with respect to the same run on the full data at once. We then describe and illustrate all steps of the segmentation methodology.

A. Theory

The stability property given in definition (1) leads to the following theorems.

Theorem 1: Let $T(I) = \{I'_i \subset I, i \in [1, N]\}$ denote a set of subset of image I such that $\bigcup_{i\in[1,N]} I'_i = I$. Let S be a stable segmentation algorithm, for which property (1) holds, i.e.,

$$\forall R \in S(I),\ R = \bigcup_{i\in[1,N]} \left(\bigcup_{R'\in S_R(I'_i)} R' \right). \tag{12}$$

Proof: Provided that $\bigcup_{i\in[1,N]} I'_i = I$, it is trivial that

$$R = \bigcup_{i\in[1,N]} R \cap I'_i.$$

In addition, from the *cover property* (4), we know that $\forall I' \in T(I)$

$$R \cap I' = \bigcup_{R'\in S_R(I')} R'.$$

Thus

$$R = \bigcup_{i\in[1,N]} \left(\bigcup_{R'\in S_R(I'_i)} R' \right). \qquad \square$$

Theorem 1 states that a segment R exactly decomposes into a set of segments from each segmentation of image subsets $I' \in T(I)$, which can be a tiling scheme, for instance. Note that $T(I)$ does not need to form a partition of I, only $\bigcup_{i\in[1,N]} I'_i = I$ is needed. As a result, the different subsets I'_i may overlap, which will be useful in the following.

Theorem 2: Let $I' \subset I$ denote a subset of image I and S be a stable segmentation algorithm, for which the *cover property* (4) holds. $\forall R \in S(I), \forall R' \in S(I')$

$$R' \cap R = \emptyset \Leftrightarrow R' \nsubseteq R. \tag{13}$$

Proof: First, let $R \in S(I)$ and $R' \in S(I')$ such that $R' \cap R = \emptyset$. From the *cover property* (4), we know that

$$R \cap I' = \bigcup_{R''\in S_R(I')} R''.$$

However, since $R' \subseteq I'$

$$R \cap R' = R \cap I' \cap R' = \emptyset.$$

Thus

$$\bigcup_{R''\in S_R(I')} R'' \cap R' = \emptyset$$

which means that $R' \notin S_R(I')$, and therefore, $R' \nsubseteq R$.

Second, let $R \in S(I)$ and $R' \in S(I')$ such that $R' \nsubseteq R$. According to (2), if $R' \nsubseteq R$, then $R' \notin S_R(I')$. From the *cover property* (4), we know that

$$R \cap I' = \bigcup_{R''\in S_R(I')} R''.$$

Since $R' \subseteq I'$

$$R \cap I' \cap R' = R \cap R' = \bigcup_{R'\in S_R(I')} R'' \cap R'.$$

We can also note that $S_R(I') \subseteq S(I')$. According to (1), $S(I')$ is a partition of I', and

$$R' \notin S_R(I') \Rightarrow R'' \cap R' = \emptyset \qquad \forall R'' \in S_R(I').$$

Thus

$$R \cap R' = \emptyset.$$

$\square$

Theorem 3: Let $I'_1 \subset I$ and $I'_2 \subset I$ such that $I'_1 \cap I'_2 \neq \emptyset$. Let S be a stable segmentation algorithm, for which property (1) holds. Let $R \in S(I)$, $R'_1 \in S(I'_1)$, and $R'_2 \in S(I'_2)$, i.e.,

$$R'_1 \cap R'_2 \neq \emptyset \Rightarrow \exists R \in S(I) \setminus (R'_1 \subseteq R \text{ and } R'_2 \subseteq R). \tag{14}$$

Proof: Let $R' = R'_1 \cap R'_2$. It is trivial that

$$\exists R \in S(I) \setminus R \cap R' \neq \emptyset.$$

Therefore

$$R \cap R'_1 \neq \emptyset \text{ and } R \cap R'_2 \neq \emptyset.$$

Thus, from Theorem 2, we have

$$R'_1 \subseteq R \text{ and } R'_2 \subseteq R. \qquad \square$$

From Theorem 3, we learn how to set together segments from the segmentations $S(I'_1)$ and $S(I'_2)$ of overlapping subsets I'_1 and I'_2: if those segments overlap, they belong to the same final segment R in segmentation $S(I)$.

Those three theorems will be used in the following section to build an algorithm for the piecewise segmentation of an arbitrarily large image on top of the stabilized mean-shift algorithm proposed in Section IV-D.

It is very interesting to note that both proofs rely on the *cover property* (4) and not on the *inner property* (3). This means that an algorithm that only respects the *cover property* can be computed piecewise at large scale with our method. No artifacts will be found on boundaries, but in this case, there will be no guarantee that the result will exactly match the segmentation of the whole image at once.

B. Algorithm for Large-Scale Segmentation

We propose the following methodology based on the theoretical proofs given in Section V-A. For a given image I, spatial radius h_s, range radius h_r, maximum number of iteration $j_{\max}$, and tile size (t_x, t_y) (note that if the proposed methodology is valid for any shape of image subset, for the sake of implementation, rectangular tiles are used):

1) extract and filter each tile;
2) segment each filtered tile with connected components;
3) relabel segmented tiles;
4) process small segments.

Fig. 7 illustrates each of the steps above. In particular, we can see how the relabelization step merges exactly segments across multiple tiles.

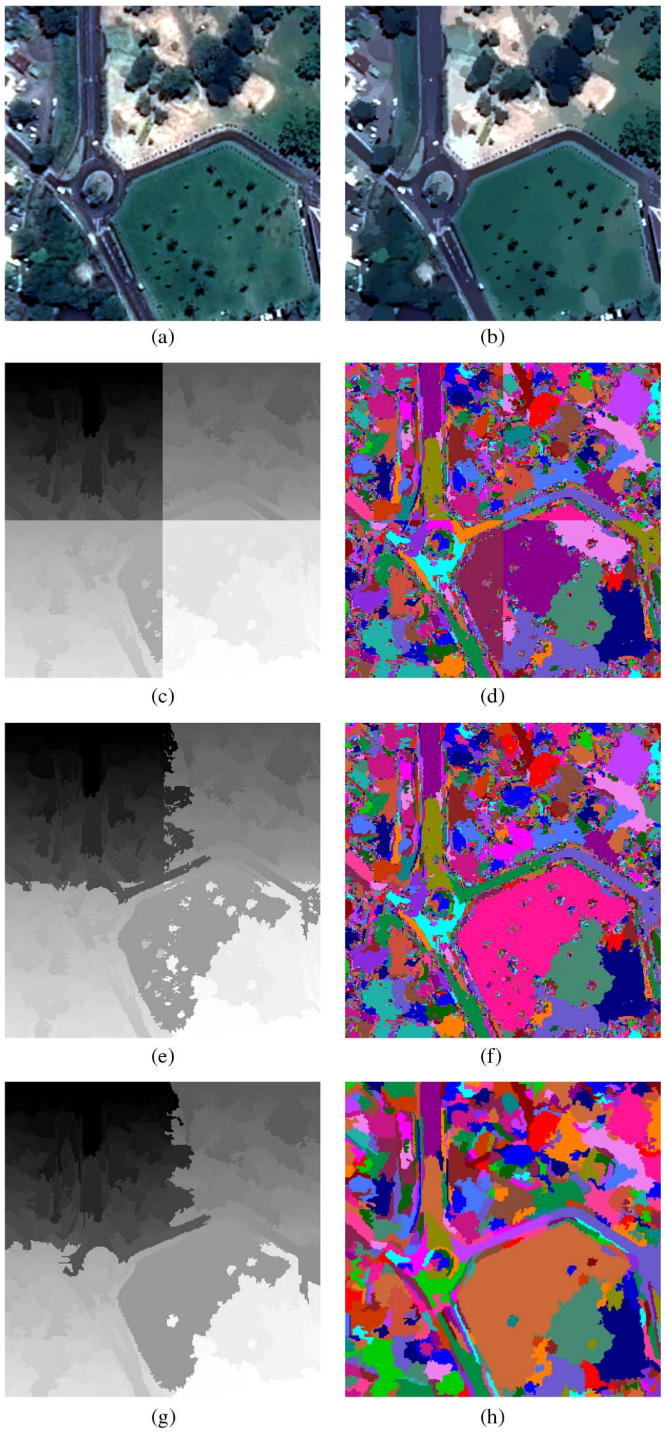


Fig. 7. Illustration of the main steps of the proposed large-scale segmentation method on a pan-sharpened Pleiades extract over Saint Denis, France. (c), (e), and (g) show the labeled images, whereas (d), (f), and (h) show a colorized version of the labeled images maximizing color differences between adjacent segments, for better visual interpretation. (a) Input image. (b) Filtered image. (c) Tile segmentation. (d) Tile segmentation (colorized). (e) Tile relabeling. (f) Tile relabeling (colorized). (g) Small segments processing. (h) Small segments processing (colorized).

1) Filtering Step: First, the stabilized mean-shift filtering algorithm is applied to each tile of size $(t_x + m, t_y + m)$, with margin $m = j_{\max} * h_s + 1$, which leads to overlapping tiles, as shown in Fig. 8. As mentioned in Section IV-D, pixels belonging to the margin do not need to be filtered, but they must

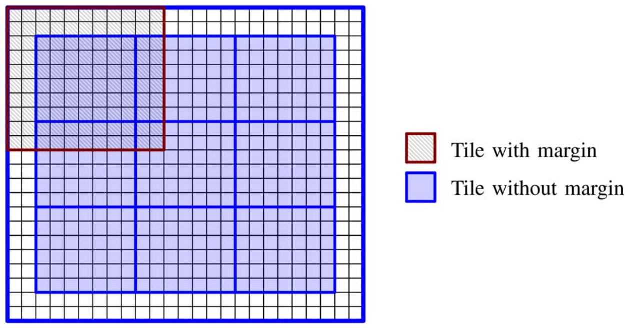


Fig. 8. Overlapping tiling used for the filtering step.

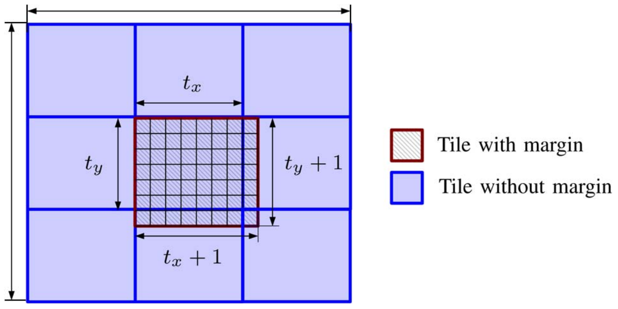


Fig. 9. Overlapping tiling used for the segmentation step.

be reachable by the filtering algorithm. The processing of each tile is completely independent and is therefore compatible with parallel or distributed execution. Once all tiles with margin have been processed, they can be stitched together to form exactly the result of the filtering of the whole image.

2) Segmentation Step: Once the filtering step is achieved, we extract tiles of size $(t_x + 1, t_y + 1)$ from the filtered image, as shown in Fig. 9. Note that one can also use a tile size completely different from the filtering step here, as long as the 1-pixel overlap is respected. This overlap will be used during the relabeling step. Each tile is independently processed by the connected-component algorithm, as explained in Section IV-D, which is again compatible with parallel or distributed execution.

Since we proved in Section IV-D that this combination of the mean-shift filtering and the connected-component algorithm makes a stable segmentation algorithm, the *inner property* defined in (3) already ensures us that any segment from the segmentation of the whole image I that is fully contained in a given tile will be exactly recovered by the segmentation of this tile.

3) Relabeling Step: After the segmentation step, provided that we ensure unique labels by shifting their values from one tile to another, we can already recover a full segmentation of the input image I. However, in this segmentation, we still have segments split across multiple tiles, and even split into several pieces within each tile, in the case of concave segments.

However, since our segmentation algorithm is stable, we know that Theorem 1 applies. Therefore, we know that if we identify which segments of the segmentations of each tile belong to the same segment of the final segmentation, then we can reconstruct this segment of the final segmentation by a simple union of the segments from each tile segmentations.

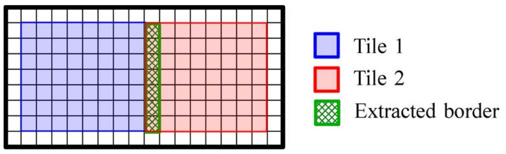


Fig. 10. Overlap of 1 pixel used to build the equivalence table.

The solution to identify which segments from the segmentations of each tile belong to the same final segment comes from Theorem 3. It states that for a pair of overlapping tiles, any pair of overlapping segments belongs to the same final segment. We only need a small overlap to determine which pairs of segments should be merged; this is why we chose the minimal overlap of 1 pixel presented in Figs. 9 and 10.

To recover the final segmentation, we explore all overlaps between all neighboring tiles to fill a label equivalence table and then relabel and merge all tiles at once.

4) Small Segments Pruning: In Section IV-D, we explicitly removed the small segments filtering step from our algorithm as a major source of instability. However, in most cases, those small segments are merely segmentation noise and can decrease the performances of downstream processing methods that assume a certain level of segmentation quality. In our large-scale segmentation method, we included a last step to filter those segments, with two possibilities: removing them or merging them. This step cannot be performed earlier in the process because we cannot ensure that small segments on the border of a tile are not actually belonging to a larger segment.

The removing of small segments is straightforward: one last pass on the final segmentation image to evaluate the number of pixels in each segment and build a table to relabel all small segments as background. The merging strategy is a bit more complex. In the original mean-shift implementation, segments with size below a user-defined threshold are merged to the closest neighboring segment in terms of radiometry. This process is iterated until no small segments are left or a maximum number of iterations have been reached. We choose a slightly different strategy: we merge segments by increasing sizes, starting with segments of size 1, until segments of the minimal acceptable size. This strategy is more conservative because it ensures that the noisier (i.e., smaller) segments will be merged first. Fig. 11 shows the results of these different strategies.

VI. Results and Performance Assessment

A. Experimental Verification of Stability

This first experiment aims at demonstrating the stability of our method with respect to the tiling scheme. To do so, a 1000×1000 pixels extract of a pan-sharpened Pleiades image is segmented with our method and with increasing number of tiles. The segmentation results are compared using Ortiz metrics to a reference segmentation obtained by processing the image with a single tile. We use a range bandwidth of 50, a spatial bandwidth of 10, a maximum number of iterations of 10, and a convergence threshold of 0.1. The connected-component predicate uses a threshold of half the spatial and spectral bandwidths, and segments smaller than 50 pixels are processed by merging. With number of tiles ranging from 2×2 (i.e., 500×500 pixel tiles) to 10×10 (i.e., 100×100 pixel tiles), the same exact result is obtained: $RC = 1$ and other metrics are null. This supports our theoretical demonstration that our method performs an exact reconstruction of the segmentation result from the segmentation of individual tiles.

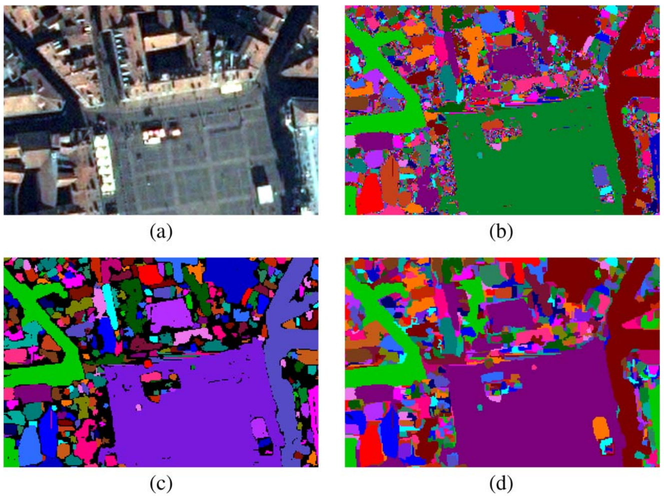


Fig. 11. Examples of the different strategies to process small segments: no processing in (b), pruning in (c), and merging in (d). Only the colorized version of the labeled images maximizing color differences between adjacent segments is shown, for better visual interpretation. (a) Input image. (b) Segmented image. (c) Removing small segments. (d) Merging small segments.

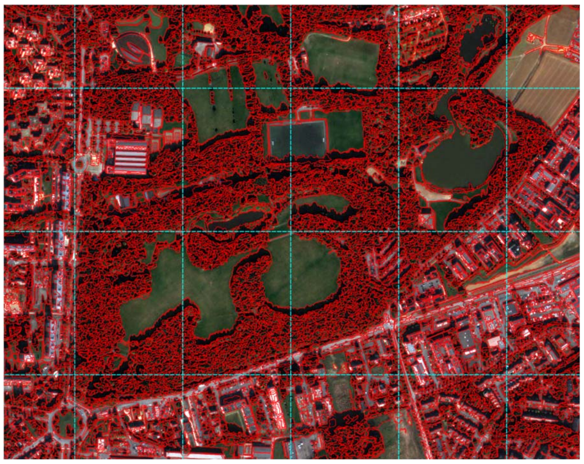


Fig. 12. Extract of segmentation results of full Rennes image ($9091 \times 12\,707$ pixels). The image has been divided into 20×20 tiles of 455×604 pixels for processing. In the extract, tile borders are represented by the dashed blue lines (8 of them are fully visible in the figure), while segments are outlined in red.

B. Large-Scale Performances

In this section, we apply our segmentation method to a Pleiades pan-sharpened scene of Rennes city in France. This extract contains $9000 \times 12\,000$ pixels with four spectral bands. For the filtering step, we used a spatial radius of 5, a range radius of 50, a maximum number of iterations of 4, and a convergence threshold of 0.1. During the segmentation step, we used a distance threshold of 30 for the connected components and a minimum size of 0 pixels. A shadow mask is also used to avoid considering shadowed pixels during the segmentation step. Finally, we performed the merging of small segments up

TABLE I
EVOLUTION OF PROCESSING TIME AND FILE SIZES WITH RESPECT TO THE INPUT IMAGE SIZE IN PIXELS

Image size	Proc. time	Nb. segments	In. raster size	Out. vector size
9000x12000	5h15	171k	1.7Go	500Mo
4500x6000	1h00	96k	420Mo	150Mo
2250x3000	0h10	53k	110Mo	40Mo

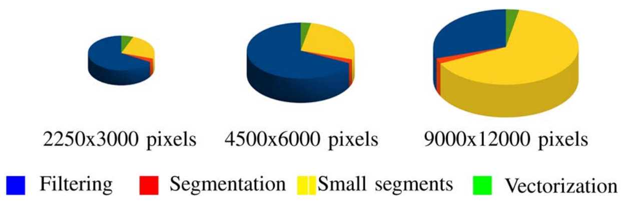


Fig. 13. Distribution of processing time among the different steps of the method.

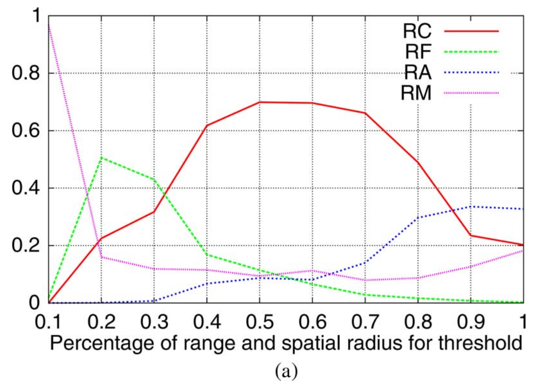
to a segment size of 200 pixels. In each step, we used a tiling scheme of 20 $\times$ 20 tiles. While it is not possible to display such large results in this paper, Fig. 12 shows a detailed view of an extract of those results. This experiment confirms that no artifacts related to tiling can be observed in the segmentation result, as stated by theory.

All experiments were conducted on an Intel Xeon(R) quadricore CPU at 2.67 GHz with 8 Gb of RAM. For the sake of compactness of the information and also compatibility with GIS software, we convert our raster segmentation to a vector layer of polygons. Table I shows some figures denoting the performances of the method for different input image sizes. We can see that the full scene is processed in 5 h and 15 min and contains around 170 000 segments. While the input image file size was 1.7 Go, the vector containing one polygon per segment, as well as statistics on image radiometry and additional shape descriptors, only weighs 500 Mo. Fig. 13 shows the distribution of the processing time among the different steps. We can see that the filtering step and the processing of small segments account for the majority of it. Note that as our work focuses on overcoming memory limitations by performing controlled tile-wise segmentation, we did nothing particular regarding processing time. Much speed up can be excepted from an optimized implementation or port to the graphics processing unit (GPU) of the filtering or processing of small segments steps, for instance, as long as the stability constraint is met. Moreover, Steps 1–3 of our large-scale segmentation method can be applied independently to each tile, and this method is therefore compatible with cluster or grid parallelism.

C. Comparison With the Original Mean-Shift Algorithm

As described in Section IV, our method uses a modified version of the mean-shift algorithm. In this section, we therefore try to assess how close our segmentation results are to the original algorithm. As a reminder, the two main changes that may affect the results are as follows: 1) the relaxed connected components with range and spatial thresholds on mode, with respect to the original grouping step strategy; and 2) the processing of small segments steps.

The first main change adds two new parameters to the algorithm: the range and spatial thresholds used during the connected-component step. To evaluate the influence of those



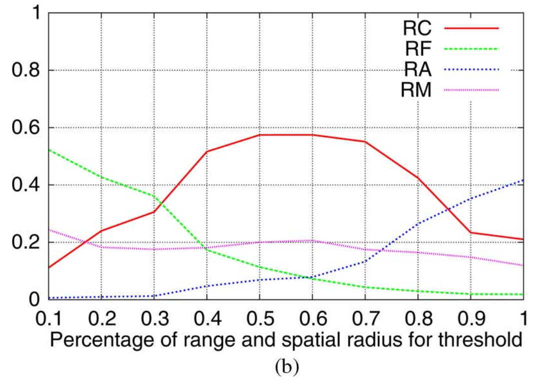


Fig. 14. Evolution of Ortiz scores for both the small segments processing strategy with respect to the percentage of range and spatial radius used as the connected-component threshold. (a) Removal strategy. (b) Merging strategy.

parameters on the segmentation quality with respect to the original mean shift, the following experiment has been conducted. A pan-sharpened Pleiades image has been segmented with both the original and stable algorithms. For both algorithms, we used a spatial radius of 10, a range radius of 50, a maximum number of iteration of 10, a convergence threshold of 0.1, and a minimum segment size of 50. We compute the Ortiz scores with respect to the original mean-shift result while varying the additional range threshold and spatial threshold parameters of the stable mean shift from 10% of range and spatial radii to 100% of range and spatial radii. We also evaluate both small segments processing strategies. Fig. 14 shows the variation of the Ortiz scores with respect to the percentage of radii used as thresholds. We can see that for both small segments processing strategies, there is an optimal around 50%, where we reach an RC score of 0.7 for the removal strategy and 0.6 for the merging strategy. We can also observe that these threshold parameters allow setting the tradeoff between oversegmentation and undersegmentation with respect to the original algorithm results. Other combination with other properties can be obtained by using different ratios for spatial and range thresholds. For instance, when trying to segment elongated homogeneous objects, setting higher spatial thresholds may lead to a better delineation.

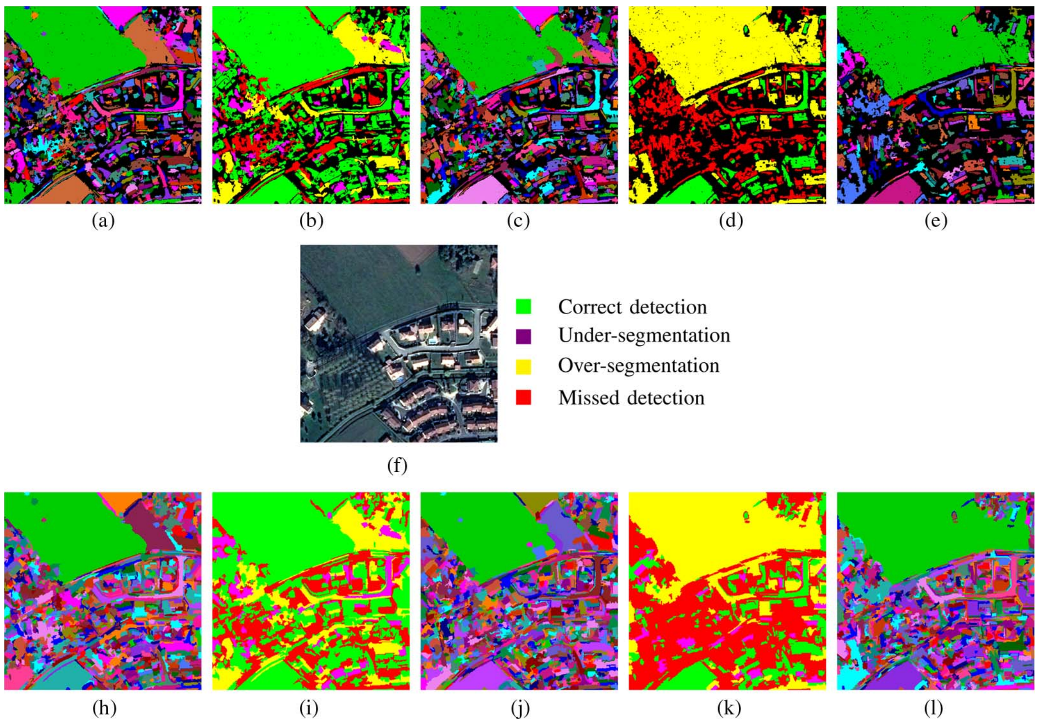


Fig. 15. Hoover scores between results from the original mean shift, connected components, and stable mean shift for both merging strategies, for range and spatial threshold set to 50% of range and spatial radius (in the case of connected components, only range threshold is used and set to range radius). Note that the threshold to display Hoover instances has been decreased to $t_s = 0.8$ (typical value for segmentation evaluation). Only the colorized version of the labeled images maximizing color differences between adjacent segments is shown, for better visual interpretation. (a) Original mean shift, removal. (b) Hoover scores (a)–(c). (c) Stable mean shift, removal. (d) Hoover scores (c)–(e). (e) Connected components, removal. (f) Input image. (h) Original mean shift, merging. (i) Hoover scores (h)–(j). (j) Stable mean shift, merging. (k) Hoover scores (j)–(l). (l) Connected components, merging.

Fig. 15 shows the Ortiz scores between original and stable mean-shift results when using a threshold of 50% of radii for both strategies, which seems optimal according to Fig. 14. It shows that both segmentation results are very similar, even if some oversegmentation occurs mainly because of the relaxed constraint induced by the connected-component algorithm. In the same figure, we also compare the results from our method to those from the connected-component algorithm alone, with a distance threshold of 50% of the range radius. We can observe that the results of our method are far more close to the original mean shift than to the connected components. Using a distance threshold of 100% yields even more different results and poorer segmentation. This highlights the strength of our method, which combines the stability brought by connected components and the accuracy of the original mean-shift algorithm.

VII. Conclusion

In this paper, we have formally defined a stability property for segmentation algorithms and proposed a method for measuring the actual stability of different segmentation algorithms. We experimentally showed that the mean-shift algorithm that is widely used in remote sensing applications is not stable according to our definition and proposed a modified version of the algorithm with guaranteed stability. We then proved that the stability property allows deriving a simple solution for piecewise processing of the segmentation results and applied this principle to the proposed stable mean shift, leading to a segmentation methodology scalable to real-world data. We then experimentally demonstrated the capabilities of this new segmentation methodology by showing that results are independent of the tiling scheme, and we showed an example of applying the method to large imagery. While this method works around the scientific locks of segmenting data at large scale, it does not, of course, resolve the well-known issues of robustness with respect to scene variability and parameter tuning. However, allowing for real and larger data processing might allow for a better knowledge of these issues in the future. Additional work may also include applying this methodology to other segmentation algorithms, and better sketching the intrinsic properties those algorithms must exhibit to be compatible large-scale processing. Further work to reduce processing time by optimizing or porting to GPU some steps of our methodology might also be needed to meet expectations of time-constrained applications. Please note that the complete source code and ready-to-use binaries are available in the Orfeo ToolBox open-source software 3.20 or later. The Orfeo ToolBox cookbook also provides a step-by-step guide to large-scale segmentation using these tools [35].

Acknowledgment

The authors thank A. Jaen and S. Harasse from CS-SI for their help with Section II, J. Nabucet from Costel for providing the pan-sharpened image of Rennes used in Section VI-B, and J. Inglada and A. Giros from the Centre National d'Études Spatiales (French Space Agency) for carefully reviewing this paper before submission. All Pleiades images are copyright of CNES (2012 and 2013), distribution Airbus DS/SpotImage.

REFERENCES

[1] T. Blaschke, "Object based image analysis for remote sensing," *ISPRS J. Photogramm. Remote Sens.*, vol. 65, no. 1, pp. 2–16, Jan. 2010.

[2] J. Inglada and J. Michel, "Qualitative spatial reasoning for high-resolution remote sensing image analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 2, pp. 599–612, Feb. 2009.

[3] M. Vanegas, I. Bloch, and J. Inglada, "Alignment and parallelism for the description of high-resolution remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 6, pp. 3542–3557, Jun. 2013.

[4] E. Christophe, J. Michel, and J. Inglada, "Remote sensing processing: From multicore to gpu," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 4, no. 3, pp. 643–652, Sep. 2011.

[5] D. Commaniciu, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[6] D. J. Crisp, P. Perry, and N. J. Redding, "Fast segmentation of large images," in *Proc. 26th Austral. Comput. Sci. Conf.*, 2003, vol. 16, pp. 87–93.

[7] A. Grote and C. Heipke, "Road extraction for the update of road databases in suburban areas," in *Proc. Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, 2008, pp. 563–568.

[8] R. Goffe, "Pyramides irrégulières descendantes pour la segmentation de grandes images histologiques," Ph.D. dissertation, Univ. Poitiers, Poitiers, France, 2011.

[9] Z. Yang, "Tiling and merging framework for segmenting large images," U.S. Patent 8 086 037, Dec. 27, 2011.

[10] P. Happ, R. Ferreira, C. Bentes, G. Costa, and R. Feitosa, "Multiresolution segmentation: A parallel approach for high resolution image segmentation in multicore architectures," in *Proc. Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, 2010, vol. 38, p. 4.

[11] J. Wassenberg, W. Middelmann, and P. Sanders, "An efficient parallel algorithm for graph-based image segmentation," in *Proc. Comput. Anal. Images Patterns*, 2009, pp. 1003–1010.

[12] T. S. Korting, E. F. Castejon, and L. M. G. Fonseca, "Divide and segment—An alternative for parallel segmentation," in *Proc. GeoInfo*, 2011, pp. 97–104.

[13] J. C. Tilton, "Split-remerge method for eliminating processing window artifacts in recursive hierarchical segmentation," U.S. Patent 7 697 759, Apr. 13, 2010.

[14] J. Michel *et al.*, "Open tools and methods for large scale segmentation of very high resolution satellite images," in *Proc. OGRS*, 2012, pp. 179–184.

[15] G. Meinel and M. Neubert, "A comparison of segmentation programs for high resolution remote sensing data," in *Proc. Int. Arch. Photogramm. Remote Sens.*, 2004, vol. 35, pp. 1097–1105.

[16] H. Zhang, J. E. Fritts, and S. A. Goldman, "Image segmentation evaluation: A survey of unsupervised methods," *Comput. Vis. Image Understanding*, vol. 110, no. 2, pp. 260–280, May 2008.

[17] J. S. Cardoso and L. Corte-Real, "Toward a generic evaluation of image segmentation," *IEEE Trans. Image Process.*, vol. 14, no. 11, pp. 1773–1782, Nov. 2005.

[18] X. Jiang, C. Marti, C. Irniger, and H. Bunke, "Distance measures for image segmentation evaluation," *EURASIP J. Appl. Signal Process.*, vol. 2006, pp. 209–209, Jan. 2006.

[19] A. Hoover *et al.*, "An experimental comparison of range image segmentation algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 7, pp. 673–689, Jul. 1996.

[20] A. Ortiz and G. Oliver, "On the use of the overlapping area matrix for image segmentation evaluation: A survey and new performance measures," *Pattern Recogn. Lett.*, vol. 27, no. 16, pp. 1916–1926, Dec. 2006.

[21] L. Di Stefano and A. Bulgarelli, "A simple and efficient connected components labeling algorithm," in *Proc. Int. Conf. Image Anal. Process.*, 1999, pp. 322–327.

[22] H. J. Johnson, M. McCormick, and L. Ibanez, "The ITK software guide," in *Updated for ITK 4.5*, 3rd ed. Chapel Hill, NJ, USA: Insight Software Consortium, 2013.

[23] E. Christophe and J. Inglada, "Open source remote sensing: Increasing the usability of cutting-edge algorithms," *IEEE Geosci. Remote Sens. Newslett.*, pp. 9–15, Mar. 2009.

[24] T. Blaschke, S. Lang, E. Lorup, J. Strobl, and P. Zeil, "Object-oriented image processing in an integrated gis/remote sensing environment and perspectives for environmental applications," in *Environmental Information Planning, Politics and the Public*, 3rd ed, vol. 2. Berlin, Germany: Metropolis Verlag, 2000, pp. 555–570.

[25] N. Jiang, J. Zhang, H. Li, and X. Lin, "Semi-automatic building extraction from high resolution imagery based on segmentation," in *Proc. Int. Workshop EORSA*, 2008, pp. 1–5.

[26] X. Huang and L. Zhang, "An adaptive mean-shift analysis approach for object extraction and classification from urban hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 12, pp. 4173–4185, Dec. 2008.

[27] D. Ming *et al.*, "Semivariogram-based spatial bandwidth selection for remote sensing image segmentation with mean-shift algorithm," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 5, pp. 813–817, Sep. 2012.

[28] D. Comaniciu, V. Ramesh, and P. Meer, "The variable bandwidth mean shift and data-driven scale selection," in *Proc. IEEE 8th ICCV.*, 2001, vol. 1, pp. 438–445.

[29] A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," in *Proc. Comput. Vis.-ECCV*, 2008, pp. 705–718.

[30] Edison: Code for the Edge Detection and Image Segmentation. [Online]. Available: http://www.caip.rutgers.edu/riul/research/code/EDISON

[31] E. Christophe and J. Inglada, "Object counting in high resolution remote sensing images with otb," in *Proc. IEEE IGARSS*, 2009, vol. 4, p. IV-737.

[32] J. Michel, J. Malik, and J. Inglada, "Lazy yet efficient land-cover map generation for hr optical images," in *Proc. IEEE IGARSS*, 2010, pp. 1863–1866.

[33] J. Michel, M. Grizonnet, and O. Canevet, "Supervised re-segmentation for very high-resolution satellite images," in *Proc. IEEE IGARSS*, 2012, pp. 68–71.

[34] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Trans. Inf. Theory*, vol. IT-21, no. 1, pp. 32–40, Jan. 1975.

[35] *The Orfeo ToolBox Cookbook, a Guide for Non-Developers Updated for OTB-3.20*, OTB Development Team CNES, Paris, France, 2013.

Julien Michel (M'14) received the Telecommunications Engineer degree from the École Nationale Supérieure des Télécommunications de Bretagne, Brest, France, in 2006.

From 2006 to 2010, he was with Communications et Systèmes, Toulouse, France, where he worked on studies and developments in the field of remote sensing image processing. He is currently with the Centre National d'Études Spatiales (French Space Agency), Toulouse, where he is in charge of research and development in image processing algorithms for remote sensing images.

David Youssefi was born in Toulouse, France, in 1990. He received the Electrical Engineering degree from the École Nationale Supérieure d'Electronique, Informatique, Télécommunications, Mathématique et Mécanique, Bordeaux, France.

He is currently a Study Engineer at Communications et Systèmes, Toulouse, where he works on remote sensing image processing.

Manuel Grizonnet received the Mathematical Modeling, Vision, Graphics, and Simulation Engineer degree from the École Nationale Supérieure d'Informatique et Mathématiques Appliquées, Grenoble, France, in 2007.

From 2007 to 2009, he was with BRGM (French Geological Survey), Niamey, Niger, as a Systems and Geological Information System (GIS) Engineer in the frame of the SYSMIN project, which aim to give the Niger ways to promote its mining potential by establishing a GIS. He is currently with the Centre National d'Études Spatiales (French Space Agency), Toulouse, France, where he is developing image processing algorithms and software for the exploitation of Earth observation images.