Available online at www.sciencedirect.com
ScienceDirect

ISPRS Journal of Photogrammetry & Remote Sensing 63 (2008) 156–168

PHOTOGRAMMETRY & REMOTE SENSING

www.elsevier.com/locate/isprsjprs

Using colour, texture, and hierarchial segmentation for high-resolution remote sensing

Roger Trias-Sanz [a,b,*,1], Georges Stamon [b], Jean Louchet [c]

[a] Institut Géographique National, 2-4 av. Pasteur, 94165 Saint-Mandé, France
[b] SIP-CRIP5, Université de Paris 5, 45 rue des Saints Pères, 75006 Paris, France
[c] Equipe COMPLEX, INRIA, 78153 Rocquencourt, France

Received 23 February 2006; received in revised form 3 August 2007; accepted 20 August 2007
Available online 24 October 2007

Abstract

Image segmentation can be performed on raw radiometric data, but also on transformed colour spaces, or, for high-resolution images, on textural features. We review several existing colour space transformations and textural features, and investigate which combination of inputs gives best results for the task of segmenting high-resolution multispectral aerial images of rural areas into its constituent cartographic objects such as fields, orchards, forests, or lakes, with a hierarchical segmentation algorithm. A method to quantitatively evaluate the quality of a hierarchical image segmentation is presented, and the behaviour of the segmentation algorithm for various parameter sets is also explored.

Keywords: Segmentation; Hierarchical; Colour; Cartography; Land cover

1. Introduction

Image segmentation is usually performed as a preprocessing step for many image understanding applications, for example in some land-cover and land-use classification systems. A segmentation algorithm is used with the expectation that it will divide the image into semantically significant regions, or objects, to be recognized by further processing steps.

Most image segmentation methods take a parameter (usually a threshold for the dissimilitude between adjacent regions) and output a partition of the image. This usually translates into a single-scale analysis of the image: small thresholds give segmentations with small regions and much detail, large thresholds give segmentations preserving only the most salient regions.

The problem is that, as it has been said since the early times of image analysis (Marr, 1982), meaningful structures and objects in a given image appear at different values of the controlling parameter (or in other words, at different scales). For a high-resolution aerial image, for example, at coarse scales we may find fields, while at finer scales we may find individual trees or plants. With

* Corresponding author. Institut Géographique National, 2-4 av. Pasteur, 94165 Saint-Mandé, France. Tel.: +33 1 4398 8345; fax: +33 1 4398 8581.

E-mail addresses: roger.trias.sanz@ieee.org (R. Trias-Sanz), stamon@math-info.univ-paris5.fr (G. Stamon), jean.louchet@inria.fr (J. Louchet).

[1] Dr. Trias-Sanz is currently at Google Inc., 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA.

0924-2716/$ - see front matter © 2007 International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS). Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.isprsjprs.2007.08.005

that in mind, several authors have proposed *multi-scale* segmentation algorithms (Guigues et al., 2003; Jolion and Montanvert, 1992; Koepfler et al., 1994; Salembier and Garrido, 2000). These analyse the image at several different scales of analysis at the same time, giving, for example, a hierarchy of regions.

Another issue is that of what input data to use for segmentation. The raw radiometric data given by the sensor is the most obvious candidate, but it might be possible that transformations of this data give better results. If the images are at high spatial resolution, the use of textural features instead of, or in addition to, radiometric data might improve the results.

This article tries to clarify these issues for the precise case of segmenting high spatial resolution (about 50 cm per pixel), low spectral resolution (red, green, and blue channels) images of rural areas, with the aim of obtaining segments that correspond to fields, orchards, forests, vines, lakes, and other such medium-size cartographic objects. We begin in Section 2 with a brief explanation of hierarchical and multi-scale segmentation methods, in particular Guigues' scale-sets method (Guigues et al., 2003), and in Section 3 we present quality measures that we will use to evaluate the segmentation algorithm's performance for different inputs. Next, in Section 4, we describe the colour space transformations and textural features that we have tested. The core of the article is Section 5, where we explore which combination of input channels gives best results.

2. Hierarchical segmentation

For the tests in this article we decided to use Guigues' *scale-sets* hierarchical segmentation algorithm (Guigues et al., 2003, 2006) because it makes both the segmentation criterion and the scale parameter explicit. The scale parameter becomes an output in multi-scale or hierarchical methods. It is then up to the post-segmentation stages to either select the most appropriate scale, or to analyse the results as a whole. This section briefly introduces Guigues' segmentation algorithm.

A partition P over a domain $\mathcal{D}$ is a division of $\mathcal{D}$ into separate pieces $p_i \in 2^{\mathcal{D}}$ (also named *cells*), as $P=\{p_1, p_2,\ldots\}$, such that $\cup_i p_i=\mathcal{D}$ (the whole domain is covered) and for all i and j, $i \neq j \Rightarrow p_i \cap p_j = \emptyset$ (the cells are disjoint). We name part $\mathcal{D}$ the set of all possible partitions of $\mathcal{D}$. For an $x \in \mathcal{D}$, let $P(x)$ denote the cell of P which contains x. A partition $P=\{p_i\}_i$ is *coarser than* another partition $Q=\{q_j\}_j$ (both over the same domain $\mathcal{D}$), and we write $P \geq Q$, if, for all $x \in \mathcal{D}$, $P(x) \supseteq Q(x)$. Conversely, Q is *finer than* P, $Q \leq P$.

Let's use as domain $\mathcal{D}$ a finite subset of $\mathbb{Z}^2$, the support for our image. Consider a partitioning algorithm A_I applied to an image I, whose result depends on a one-dimensional parameter λ, the *scale parameter*

$$A_I : \mathbb{R} \rightarrow \text{part}(D), \tag{1}$$

and which is *causal*, that is,

$$\lambda_2 \geq \lambda_1 \Rightarrow A_I(\lambda_2) \geq A_I(\lambda_1). \tag{2}$$

so we can represent the partitions $A_I(\lambda)$, for all $\lambda \in \mathbb{R}$, as a tree or *hierarchy* H. Because A_I is causal, the set $\Lambda(c)$ of values λ for which a given cell c is found in the partition $A_I(\lambda)$ is an interval $\Lambda(c)=[\lambda^+(c), \lambda^-(c)[$ (and $\lambda \in \Lambda(c) \Leftrightarrow c \in A_I(\lambda)$). We call $\lambda^+(c)$ the scale of appearance of c.

For each λ, Guigues' algorithm efficiently finds the partition that optimizes the balance between *goodness-of-fit* to the data and *complexity*. Theories of model inference (variational (Mumford and Shah, 1989), Bayesian (Geman and Geman, 1984), minimum encoding (Leclerc, 1989)) indicate the need for a *regularisation* that takes into account the size or complexity of the solution. Usually, the end goal is finding a partition $P=A_I(\lambda)$ which minimizes an energy such as

$$E(P,\lambda) = D(P) + \lambda C(P) \tag{3}$$

where $D(P)$ measures how badly P represents the original image and $C(P)$ measures the complexity of P. To avoid a problem known as "spill-over", in which all the edges of an interesting region are in the hierarchy H, but the region itself is not a cell in H (i.e., it is not found as a single piece there), we consider, instead of this hierarchy, a flattened version, a graph G consisting of the finest (lowest) level of the hierarchy, with edges attributed with the maximum level (λ) they reach in the hierarchy (see Trias-Sanz (2005) for details).

3. Evaluating the quality of a multiscale segmentation

As with classical, single-scale, segmentation algorithms, the need arises to evaluate the quality of a multiscale segmentation against a reference, in order to compare different algorithms, and to select for an algorithm the parameters which are optimal for a given application. Most current segmentation evaluation methods (Liu and Yang, 1994; Segui Prieto and Allen, 2003) handle only single-scale segmentations, that is, partitions of an image. They usually work by finding correspondences between points in the reference and points in the edges of the regions given by the segmentation.

However, because multi-scale algorithms can deliver arbitrarily fine segmentations—at the finer end of the scale range—the concepts of "correspondence between reference points and segmentation edge points" and of "distance between segmentation edge and reference edge" cannot be easily transposed to the multi-scale case. In this section we present a method for evaluating the quality of a multi-scale or hierarchical image segmentation against a reference, which was briefly sketched in Trias-Sanz (2005). In the following sections, we will use these quality measures to determine the most appropriate input data to use for the segmentation. The reference segmentation is given as a set of edges of two kinds, *compulsory* and *optional*. The method computes two measures, a *false detection* (or commission) measure, and a *missed detection* (or omission) measure, by considering that a falsely detected segmentation edge is a more important error if it appears at a coarse scale of analysis; conversely, the missed detection measure takes into account the fact that a reference edge can be completely missed—by there not being a corresponding segmentation edge—but also "nearly missed" if only fine-scale corresponding segmentation edges are found. This assumes that the most visually salient edges are found at the coarsest scales, which is the case. These two measures can be minimized jointly, or an aggregate such as their average can be calculated. We have found that they can be used to compare two different multi-scale segmentations of the same image.

The procedure to compute the segmentation quality measures operates by searching for pixels belonging to segmentation edges and for reference pixels. The first step is therefore to convert the inputs into a suitable discrete (pixel-based) form.

The segmentation reference is given as two sets of segments, one for the compulsory edges and one for the optional edges. A rasterisation algorithm—such as Bresenham's line drawing method—is applied to convert these sets of edges into two sets of pixels. Note that information about what edge each pixel belongs to is lost. Let $D \subset \mathbb{Z}^2$ be the usually rectangular image domain. Pixel coordinates are elements of D. Let $R_c \subset D$ be the set of pixels given by the compulsory reference edges. Let $R_o \subset D$ be the set of pixels given by the optional reference edges, and $R = R_c \cup R_o$.

The flattened graph G of Section 2 obtained from a multi-scale segmentation is also converted to a pixel-based form. To improve processing time, we first discard the 30% of edges with smallest λ^+; we experimentally determined, by observing real data, that all such edges are due to noise or insignificant details only, and thus can be safely removed. We then sort the remaining edges according to their values of λ^+, and attribute each edge with a weight w, which is 1 for the first edge (that with highest λ^+) and decays exponentially for the remaining edges. The graph is rasterized by giving to each pixel the maximum of the weights w of all edges that occupy the pixel. We name the set of pixels with an edge is $S \subset D$, and the weight of a pixel $\mathbf{s} \in S$ as $w^+(\mathbf{s})$.

Computing the missed detection and false detection measures involves searching in small circular neighbourhoods of pixels, of fixed radius ϵ. The neighbourhood radius ϵ is used as a distance threshold when determining if a segmentation pixel and a reference pixel could correspond and, therefore, it must be set to take into account positional accuracy in the reference and acceptable positional errors in the segmentation edges, and must keep the same value for all tests in a comparison of segmentations.

Let $B_{\boldsymbol{x}}$ be the ball in D centred on $\boldsymbol{x}$ with radius ϵ,

$$B_{\boldsymbol{x}} = \{\boldsymbol{y} \in D : ||\boldsymbol{y} - \boldsymbol{x}|| \leq \epsilon\}. \tag{4}$$

Given the R and R_c reference sets, the S segmentation set, the weight map w^+, and the neighbourhood radius ϵ, the missed detection penalty for a single pixel $\boldsymbol{x} \in R_c$ is defined as

$$p_m(\boldsymbol{x}) = \begin{cases} 1 - \max\limits_{\boldsymbol{y} \in B_{\boldsymbol{x}} \cap S} w^+(\boldsymbol{y}) & \text{if } B_{\boldsymbol{x}} \cap S \neq \emptyset, \\ 1 & \text{if } B_{\boldsymbol{x}} \cap S = \emptyset, \end{cases} \tag{5}$$

and the false detection penalty for a single pixel $\boldsymbol{x} \in S$ as

$$p_f(\boldsymbol{x}) = \begin{cases} 0 & \text{if } B_{\boldsymbol{x}} \cap R \neq \emptyset, \\ w^+(\boldsymbol{x}) & \text{if } B_{\boldsymbol{x}} \cap R = \emptyset. \end{cases} \tag{6}$$

The global missed detection quality measure M and the global false detection quality measure F are, respectively,

$$M = \frac{\sum_{\boldsymbol{x} \in R_c} p_m(\boldsymbol{x})}{|R_c|}, \qquad F = \frac{\sum_{\boldsymbol{x} \in S} p_f(\boldsymbol{x})}{\sum_{\boldsymbol{x} \in S} w^+(\boldsymbol{x})}. \tag{7}$$

The missed detection—or omission—penalty p_m is computed for all pixels corresponding to compulsory ground truth segmentation edges. This penalty is maximum, as in single-scale segmentations, when no corresponding segmentation edge pixel is found; in this case, $p_m = 1$. However, even when a corresponding pixel is found, the penalty is non-zero; in this case, the higher the pixel's scale, the lower the value of p_m. Therefore, if the only segmentation pixel corresponding to a ground truth edge is found at very fine scales of analysis, it will be counted as an "almost-missed detection". On the contrary, if the corresponding pixel is very salient and appears at coarse scales of analysis, the penalty will be small.

Conversely, the false detection or commission penalty p_f is computed for all pixels corresponding to detected segmentation edges, that is, edges present in the segmentation algorithm's output. No penalty is given ($p_f = 0$) if a corresponding ground truth edge is found. If none is found, the pixel's scale is used for the penalty. Therefore, very salient detected edges that do no correspond to the ground truth have a high p_f penalty, while incorrectly detected edges at fine scales of analysis are better tolerated.

This provides a two-dimensional measure of the quality of a segmentation compared to a ground truth, $(M, F) \in [0, 1] \times [0, 1]$. The closer M and F are to zero, the higher the quality.

4. Segmentation energy

As mentioned in Section 2, the segmentation process optimizes a balance of a measure of the goodness-of-fit (how well the segmentation fits to the original image) and a measure of the complexity of the segmentation. The segmentation parameter λ balances between a perfect fit consisting of one segmentation region for each pixel in the original image, and the simplest segmentation consisting of a single region containing the whole image.

Guigues' basic segmenter uses Mumford and Shah's piecewise-constant model (Mumford and Shah, 1989) (also known as *cartoon model*) with images in the RGB (red, green, blue) colour space. While this model is good for the general case, we can use models better suited to our application: Fields, forests and other vegetation regions are not necessarily constant in RGB space. At first sight, it is texture and colour hue that are more or less constant.

4.1. Colour spaces

In this section we will describe the colour spaces we investigated for use in our system. Note that we may choose to use some coordinates from one colour space and some from another one. See Cheng et al., (2001) for a review of colour spaces in the context of image segmentation.

Fig. 5 (top) shows a sample fragment of the data set that will be used for the evaluation in Section 5.2. For that data set, only the red, green, and blue channels are available, so raw and transformed channels requiring near-infrared data could not be tested. Their original resolution is 80 cm, but they have been resampled to 50 cm, and the original data is not available. This downsampling does not introduce any visible blur, perhaps because the original images contain true information for all channels in each pixel, instead of using a Bayer pattern.

Some of the colour spaces that we used are well-known and will not be described further: The original red-green-blue space (r, g, b). The hue-saturation-intensity space (h, s, i). The CIE XYZ, Lab, and Luv spaces (whose coordinates we represent by X, Y, Z; L, $a*$, $b*$; L, $u*$, $v*$).

In addition, we studied a common simple haze removal technique that consists in subtracting from each radiometry channel its minimum. If $I_i(\boldsymbol{z})$ is the image value for channel i at position $\boldsymbol{z}$, the new channels i' are given by

$$I_{i'}(\boldsymbol{z}) = I_i(\boldsymbol{z}) - \min_{\boldsymbol{v}} I_i(\boldsymbol{v}). \tag{8}$$

This is applied to the raw red, green, and blue channels to give $c' = (r', g', b')$.

We tested log-opponent chromaticity coding, which has been long proposed (Berens and Finlayson, 2000; Faugeras, 1979) as a perceptually relevant colour transformation. The two coordinates (l_1, l_2) of log-opponent chromaticity space can be derived from the (r, g, b) coordinates as follows:

$$l_1 = \log r - \log g \tag{9}$$

$$l_2 = \log r + \log g - 2 \log b \tag{10}$$

Finally, we also experimented with other, less-known, colour transformations. From (r, g, b) coordinates,

$$e_{1a} = \log \frac{r}{r+g+b}, \quad e_{1b} = \log \frac{g}{r+g+b} \tag{11}$$

$$e_{2a} = \frac{r}{g}, \quad e_{2b} = \frac{b}{g}, \quad e_3 = \log \frac{b}{g}; \tag{12}$$

the red/cyan chromaticity from Van de Wouwer et al. (1999),

$$k_2 = \frac{r}{2} - \frac{b}{2}; \tag{13}$$

and Angulo (Angulo et al., 2003)'s c_1 and c_2, combining the hue, saturation, and intensity channels as

$$c_1 = s \quad h, \qquad c_2 = (1-s)i. \tag{14}$$

4.2. Texture parameters

We would also like to find other kinds of parameters that mostly remain constant across a vegetation region and present discontinuities at its edges. Vegetation is heavily textured (for example, we can distinguish between a forest, a field, and a vine from texture alone, with grayscale images), so texture parameters can be useful.

Texture is typically a local property, but not a property of single pixels. Colour, on the other hand, is a pixel property. Thus, there is no such thing as the texture of a pixel, but only the texture of a region. But the way to partition an image into regions is to use a segmentation algorithm. How can we use texture parameters as input to the segmentation algorithm, if we do not know yet over what regions we should calculate texture parameters? There are two solutions.

The first solution assumes that the actual shape of the region we use for calculating texture is mostly irrelevant. For each pixel in the image, we calculate texture parameters over a square neighbourhood centred on it. We can then use the results as a pixel-level input to the segmentation algorithm. Since the neighbourhoods centred on pixels at the boundaries of vegetation regions will not contain a single texture, at these boundaries there will not be sharp discontinuities in texture, as we would like to have, but smooth transitions, or other artifacts. With this solution we are assuming that this smoothing is not a problem. We call these texture features "pixel-based".

The second solution is to first run the segmentation algorithm—which operates by iteratively merging regions—using only colour and pixel-based texture features as input, until a certain condition on region size is met—in particular, for the tests shown in this article, two regions may be merged only if each is smaller than 400 m^2 and at least one is smaller than 200 m^2. We then calculate the texture parameters over each of these small regions instead of over square neighborhood, and continue the merging at the point we had left it, using also these newly calculated texture features this time. In this solution, we assume that when we first stop the algorithm, the merge regions will be big enough so that the texture calculation in them will be possible, and at the same time small enough that they are still homogeneous. We call these "region-based". Not all textural features can be calculated in this way.

These texture operators work on single-channel images only. We use the image intensity (from the HSI colour space) as input.

We have experimented with several common and some less-known texture features, which we list below. However, as we will see in Section 5.2, only one of these features, *structure complexity*, has given good results; in all other cases it is best to use raw or transformed radiometric data rather than texture. Therefore, for conciseness, we will describe in detail only the *structure complexity* feature, and simply list and give references to the rest.

We have not tested some classical texture features, such as those based on Haralick's co-occurrence matrices, because we had found out in previous work that these are not very discriminant for high-resolution remote sensing images. These are images whose texture may be periodic in one or in two directions, with, when periodic, large periods; the angle of the texture orientations is irrelevant but the periods are very discriminative, and whether there are no, one or two orientations is relevant. The texture features that we did test are meant to measure these discriminant values.

4.2.1. Structure complexity

Baillard (1997) and Guigues (2000) present two texture features to distinguish textures depending on the amount of structure present. Baillard uses the entropy of the histogram of gradient directions, and Guigues's variant uses a weighted sum of gradient directions based on Medioni's tensor voting; both take into account the amount of texture present, giving in effect three classes: flat texture (or untextured), unordered orientations (corresponding to natural textures) and ordered orientations (corresponding to artificial objects).

Baillard's method starts by computing the gradient of the source image. This is decomposed into module $M(z)$ and argument $\Theta(z)$. Let $R(z)$ be a mask of the region of interest. For region-based texture features, $R=0$ everywhere except in the region's support, where $R=1$. For pixel-based texture features, R will be a Gaussian kernel $\mathcal{N}_{10}$ (of $\sigma=10$) centred on the current pixel.

The number N of pixels in the region weighted by the mask R, the set S of pixels whose gradient module is higher than a threshold c (in Baillard (1997), $c=6$), and the number N_v of pixels in S weighted by the mask R, are calculated as

$$N=\sum_{z}R(z),\qquad S=\{\mathbf{w}:M(\mathbf{w})>c\},\tag{15}$$

$$N_v=\sum_{z\in S}R(z).\tag{16}$$

Then, a histogram is constructed from the values of Θ in the region S. If the angle domain $[0,\pi)$ is divided into N_a equally-sized bins, the relative histogram count for the i-th bin is $H(i)$,

$$S(i)=\{\mathbf{w}\in S:i\pi/N_a\le\Theta(\mathbf{w})<(i+1)\pi/N_a\},\tag{17}$$

$$H(i)=\frac{1}{N_v}\sum_{z\in S(i)}R(z),\tag{18}$$

and the histogram entropy is computed as

$$E=-\sum_{i}H(i)\log_2 H(i).\tag{19}$$

We have extended Baillard (1997) in two ways: First, we use N_v/N as a feature—Baillard (1997) uses only E—and second, we use a non-constant mask M in pixel analysis—Baillard (1997) uses constant masks. The ratios N_v/N and E are calculated both as pixel-based and region-based features.

Guigues's method also starts with the gradient of the source image, in polar form M and Θ. The gradient angle is then multiplied by 4 to obtain a $\pi/2$ invariance,

$$X(z) = M(z)\cos(4\Theta(z)), \tag{20}$$

$$Y(z) = M(z)\sin(4\Theta(z)), \tag{21}$$

Finally, the gradient module is smoothed, either directly to give h_b, or from its $\pi/2$-invariant form to give h_a:

$$\bar{X}(z) = X(z) * \mathcal{N}_{10}(z), \tag{22}$$

$$\bar{Y}(z) = Y(z) * \mathcal{N}_{10}(z), \tag{23}$$

$$h_a(z) = \sqrt{\bar{X}^2(z) + \bar{Y}^2(z)}, \tag{24}$$

$$h_b(z) = M(z) * \mathcal{N}_{10}(z). \tag{25}$$

The values h_b and h_a/h_b are used as pixel-based texture features.

4.2.2. Gabor filters

Gabor filters (Bülow and Sommer, 1998; MacLennan, 1991) are filters localized in space and frequency, which can be used to retrieve frequential properties of a texture. In their most common use, a bank containing a number of Gabor filters is run on an image.

4.2.3. Fractal dimension

The fractal dimension is often used as a measure of complexity (Peleg et al., 1984; Talukdar and Acharya, 1995; Tao et al., 2000). We have used as texture features the results of Tao, Lam, and Tang's implementation (Tao et al., 2000) of the *blanket* technique.

4.2.4. Local binary patterns

Combining rotational-invariance with multi-resolution analysis, Olaja et al. (2002) propose a texture classification method using *local binary patterns*: These are a reduced number of local patterns whose statistical frequencies are shown to be sufficient for texture classification.

4.2.5. Dot-pattern detector

The dot-pattern selective cell operator of Kruizinga and Petkov (2000) is a biologically motivated texture operator specialized in the detection (and characterisation) of dotted textures.

4.2.6. Scale-orientation histogram

The scale-orientation histogram of Zhou et al. (2003) gives an estimation of the main orientation of a texture, the degree or directionality, and the scale of analysis at which this directionality is found.

4.2.7. Fourier-based orientation estimator

There are many texture extraction methods using Fourier analysis. Garnier (2002) proposes a simple but effective method that gives the main orientation *and spatial period* of a texture, when they exist, and also the secondary orientation and period, for textures with at least two prominent directions. This can be used to distinguish among different types of vegetation.

4.2.8. Intensity average

In addition, the average of pixel intensity (channel i in Section 4.1) across all pixels in a region is also calculated as a "region-based" texture feature.

5. Choosing the best segmentation parameters

As suggested in the introduction, it may be that using transformed colour channels instead of, or in addition to, raw radiometry as input to a segmentation algorithm produces better results. In addition, at high resolutions it becomes possible to calculate texture features, the use of which may also be beneficial. We also proposed a two-phase segmentation procedure which makes the use of texture features—whose calculation requires, in principle, an image segmentation, which is of course not yet available at that point. In this section, we will explore which input colour and texture channels—of those described in Section 4—give the best results. For these evaluations, we will use the quality measures of Section 3 on a manually-defined ground truth of approximately 4 km^2.

The parameter space for these tests is too large. We will assume that some parameters are mutually independent and optimize them separately, and we will use Stepwise Forward Selection (SFS) for the choice of the optimal set of input channels.

Because quality measures are two-dimensional, it is not trivial to compare parameter sets to find the optimum. We will use *Pareto optimality* (Pareto, 1906; Gawthrop, 1970, Allan Schick, p. 32): Given two n-dimensional vectors $a=(a_0, a_1,\ldots,a_{n-1})$ and $b=(b_0, b_1,\ldots,b_{n-1})$ we say that,

$$a > b \iff \forall i \in \{0..n-1\}\,.\, a_i > b_i, \tag{26}$$

```
Given $(R_1, R_2, U_1, U_2, C_p, C_r)$, the current state of the SFS algorithm, in order to
obtain $T$, the list of tests to be performed in the next step, do as follows:
For each unused channel $c \in C_p \cup C_r$, do:

  1. If $R_2 = \emptyset$ and $c \in C_p$:
      If $c$ is a linear channel: $T \leftarrow (R_1 \cup \{c\}, R_2, U_1 \cup \{c\}, U_2, C_p \setminus \{c\}, C_r)$.
      Else, if $c$ is a polar channel: For each $d \in \ell(U_1) \cup \{1\}$, and for each S in $\mathcal{C}(c,d)$,
          $T \leftarrow (R_1 \cup S, R_2, U_1 \cup \{c\}, U_2, C_p \setminus \{c\}, C_r)$.

  2. If $R_1 \neq \emptyset$:
      If $c$ is a linear channel: $T \leftarrow (R_1, R_2 \cup \{c), U_1, U_2 \cup \{c\}, C_p \setminus \{c\}, C_r \setminus \{c\})$.
      Else, if $c$ is a polar channel: For each $d \in \ell(U_2) \cup \{1\}$, and for each S in $\mathcal{C}(c,d)$,
          $T \leftarrow (R_1, R_2 \cup S, U_1, U_2 \cup \{c\}, C_p \setminus \{c\}, C_r \setminus \{c\})$.
```

Fig. 1. Procedure to obtain, in the modified SFS algorithm, the list of tests to perform in one step.

Given a set of n-dimensional vectors S, the minima set (in the sense of Pareto) of S, min S, is the set

$$\min S = \{a \in S : \forall b \in S \,.\, \neg\,(a > b)\} \tag{27}$$

where ¬ is the logical “not”.

5.1. Methodology

The parameter space, taking into account colour transformations and texture parameters, is too large to be explored systematically. We will assume that some parameters are mutually independent and optimize them separately. At each optimization phase we will not select a single global optimum, but the Pareto optima set—and, because the assumption of parameter independence may not be true, also some parameter sets close to the optimum.

The parameters to be optimized are the following:

(1) Radiometry, derived radiometry, and pixel-based texture channels to use as segmentation input as the first segmentation phase. Some features, such as h, have an angular nature; how this is integrated with other features must also be taken into account.

(2) Region-based texture channels to use as segmentation input as the second segmentation phase, and the integration of angular features.

The tested features are those of Sections 4.1 and 4.2. We shall use the notation of these sections, for example, r for the raw red channel.

5.2. Choice of input channels

Because of the enormous size of the parameter space—all possible combinations of raw radiometric data, transformed colour channels, and pixel-based and region-based texture features—we cannot do any kind of exhaustive search. The parameters being non-numerical, typical gradient descent optimization techniques are not applicable here. For these reasons, we have chosen to optimize the choice of channels using SFS. This technique proceeds as follows: in the first step, each channel is tested separately; that giving the best results is retained for the second step. In the second step, we test each combination of the channel retained in the first step and another channel. The two

Table 1
Best-performing parameter sets for the first step of the SFS

R_1	R_2	M	F
g	∅	0.3194	0.2570
cos(h), sin(h)	∅	0.3186	0.3103
$\boldsymbol{i}$	**∅**	**0.3466**	**0.2629**
X	∅	0.3189	0.2621
$\boldsymbol{Y}$	**∅**	**0.3507**	**0.2602**
Z	∅	0.3466	0.2529
[l_2]	[∅]	[0.3081]	[0.3270]
[h_b]	[∅]	[0.5534]	[0.2202]
$\mathbf{c}_2$	**∅**	**0.3425**	**0.2612**
$\boldsymbol{g'}$	**∅**	**0.3361**	**0.2698**
$\boldsymbol{b'}$	**∅**	**0.3306**	**0.2573**

Sets marked in **boldface** are not Pareto-optimal but were nevertheless retained as initial solutions for the next step because of their high quality. Sets in [brackets] are Pareto-optimal but were discarded because they are too far away from the origin.

Table 2
Best-performing parameter sets for the second step of the SFS

R_1	R_2	M	F
g, Z	$\emptyset$	0.3218	0.2520
g, b'	$\emptyset$	0.3879	0.2382
X, Y	$\emptyset$	0.3131	0.2521
Z, e_{1a}	$\emptyset$	0.3279	0.2511
c_2, e_{2a}	$\emptyset$	0.3367	0.2428
g', b	$\emptyset$	0.2931	0.2562
c_2	Z	0.3948	0.1923
b'	g'	0.3907	0.2042
Y	Z	0.4228	0.1820
X	c_2	0.4145	0.1919
X	b'	0.4208	0.1823

channels giving the best result (one of which will be the channel obtained in the first step) are retained for the next step. The procedure may be stopped at a fixed step, or when the best result from one step is worse than that of the previous step. In the end, the channels giving the best overall result are selected.

We have modified the standard SFS to suit two needs: First, the distinction between the first and second segmentation phases must be taken into account. Second, "polar" channels must be properly handled. In addition, if several results at one step look promising, we may keep all of them instead of only the best. A *polar* channel, such as the *hue*, one that has an angular nature, cannot be directly used as an input to the segmentation because the segmenter would consider values close to 0 as very different from values close to 2π, when they are in fact similar. Let ϕ be a channel with angle values. Let f be a channel with non-angular values (a *linear* channel), which we will assume to have values in [0, 1]. Then, instead of $\{\phi\}$, we can use as the set of segmentation inputs one element of $C(\phi, f)$,

$$\begin{aligned}\mathcal{C}(\phi,f) = \{&\{cos\phi, sin\phi\}, \{cos\phi\}, \{sin\phi\}, \{f\,cos\phi, f\,sin\phi\}, \\ &\{f\,cos\phi\}, \{f\,sin\phi\}, \{(1-f)cos\phi\}, \\ &\{(1-f)sin\phi\}, \{(1-f)cos\phi, (1-f)sin\phi\}\}.\end{aligned} \tag{28}$$

The crucial part of the SFS algorithm is deciding which tests to perform given the results of the previous step. Let the best result of one step be described by a tuple (R_1, R_2, U_1, U_2, C_p, C_r): R_1 is the set of pixel-based segmentation inputs for the first segmentation phase, after transformation of polar channels; this will be empty in the first step. R_2 is the set of all inputs (pixel-based and region-based) for the second phase; this will be empty for single-phase segmentation. U_1 is the set of channels used in R_1 before polar transformation, and U_2 that for R_2. C_p is the set of pixel-based channels not in $U_1 \cup U_2$, and C_r is the set of region-based channels not in U_2. For example, $R_1 = \{E, \cos h\}$, $U_1 = \{E, h\}$, $R_2 = \emptyset$, $U_2 = \emptyset$. Let's note by $T \leftarrow m$ the addition of the test described by the tuple m to the list of tests T to be performed in the next step. For a set of channels U, let's note $\ell(U)$ the subset of *linear* channels. Then the list of tests to be performed in the next step can be obtained, in the adapted SFS algorithm, as described in Fig. 1. The algorithm is first run with a state tuple ($\emptyset$, $\emptyset$, $\emptyset$, $\emptyset$, C_p, C_r) with C_p the set of pixel-based channels and C_r the set of region-based channels.

Table 3
Best-performing parameter sets for the third step of the SFS, and results with the three raw radiometry channels

R_1	R_2	M	F
g', b, s	$\emptyset$	0.3061	0.2551
g', b, e_{1a}	$\emptyset$	0.3182	0.2383
X	c_2, k_2	0.4082	0.1675
X	c_2, b^*	0.4016	0.1807
X	b', e_{2a}	0.3798	0.1808
X	**b', g'**	**0.3661**	**0.1930**
c_2	Z, e_3	0.3490	0.1848
b'	**g', g' sin*(h)***	**0.3611**	**0.2139**
b'	g', e_{1a}	0.4056	0.1794
Y	Z, sin*(h)*	0.4062	0.1773
r, g, b	$\emptyset$	0.3722	0.2569

Sets marked in **boldface** are not Pareto-optimal but were nevertheless retained as initial solutions for the next step because of their high quality.

The choice of ϵ in Eq. (4) affects the actual quality measures, but has no effect on the quality itself: higher values of ϵ give lower values to the quality measures (and thus show higher quality), but this apparent improvement is, of course, caused by a higher tolerance of the quality measurement. In the tables and plots shown in this chapter, we have chosen $\epsilon = 4$.

Using a manually-defined reference for evaluation, we selected the best combination of input channels by running four iterations of the SFS algorithm described above. Tables 1, 2, 3, and 4 list the best-performing

Table 4
Best-performing parameter sets for the fourth step of the SFS

R_1	R_2	M	F
g', b, e_{1a}, e_{1a} cos(h)	$\emptyset$	0.3170	0.2409
g', b, e_{1a}, (1−e_{1a}) sin(h)	$\emptyset$	0.3069	0.2531
g', b, e_{1a}, e_{2a}	$\emptyset$	0.3123	0.2520
X	c_2, k_2, v^*	0.4042	0.1615
X	c_2, k_2, g'	0.3902	0.1689
X	c_2, b^*, g'	0.3884	0.1722
X	b', g', k_2	0.3818	0.1745

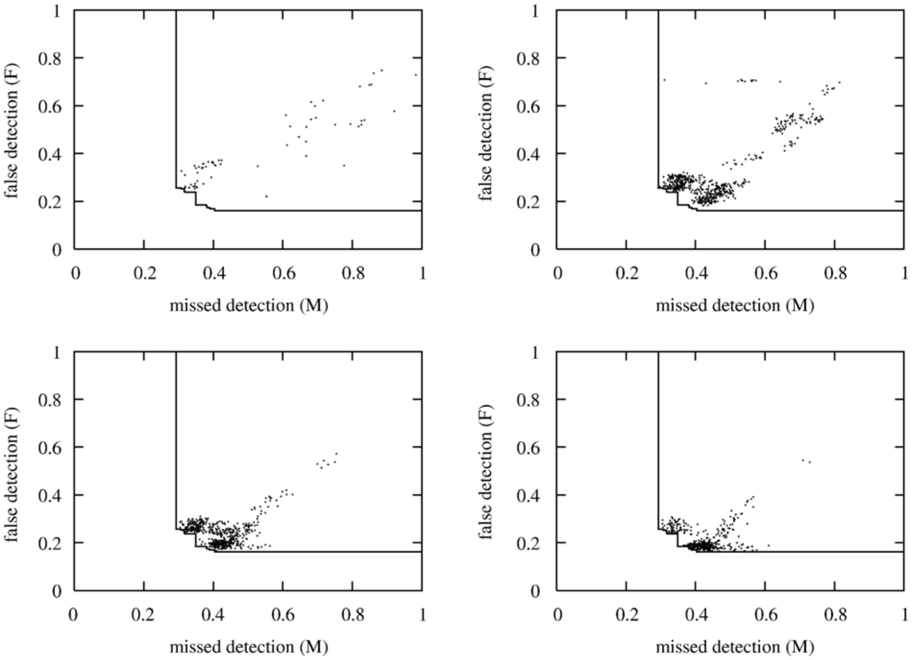

Fig. 2. Scatter plot of the segmentation quality measures for the parameter sets evaluated in each SFS step. Top left: first step; top right: second; bottom left: third; bottom right: fourth. Also shown is the final Pareto limit.

channels for each SFS iteration respectively. Table 3 also includes, for reference, results using the raw RGB channels; they perform significantly worse than other channel combinations. In these tables we indicate, for each channel set R_1 and R_2 used, the two quality measures M and F obtained. Fig. 2 shows the scatter plots for the quality measures for the test sets at each SFS iteration, and the Pareto limit at the fourth step, and Fig. 3 the results after these four iterations. Here “best-performing” means Pareto-optimal, as well as some additional close-to-optimal parameter sets (marked in **boldface**). The best-performing sets (except some Pareto-optimal sets which were too far away from the origin, and which are put in [brackets]) were retained as initial solutions for the next SFS iteration. In addition, at each iteration we removed from the available channel set $C_p \cup C_r$ those channels that exhibited the worst performances.

Since the fourth SFS step shows little improvement compared to the third step, we stopped iterating the SFS algorithm at this point. Taking into account the results at all steps, the best choice of channels is given in Table 5. We will select, as optimal parameter set, that with c_2 as input channel for the first segmentation phase, and Z and e_3 as input channels for the second segmentation phase, which has a good balance in the values of M and F. For this parameter set, $M=0.3490$ and $F=0.1848$.

These results are to be compared with those of Vansteenkiste et al. (2004). In that paper, colour channels and texture features are tested for a terrain segmentation in high-resolution satellite images. The authors conclude that texture does not add any significant advantage. In this section we have obtained similar results, with a more thorough evaluation: we have tested many more texture features and colour spaces, and have performed the tests over a larger area. We found that, indeed, texture does not seem to be useful to obtain good segmentations: no texture channel is present in any of the optimal parameter sets. However, we do see the

Table 5
Pareto-optimal segmentation parameter sets after four iterations of SFS

R_1	R_2	M	F
g', b	$\emptyset$	0.2931	0.2562
g', b, s	$\emptyset$	0.3061	0.2551
g', b, e_{1a}, $(1-e_{1a})\sin(h)$	$\emptyset$	0.3069	0.2531
g', b, e_{1a}, e_{2a}	$\emptyset$	0.3123	0.2520
g', b, e_{1a}, $e_{1a}\cos(h)$	$\emptyset$	0.3170	0.2409
g', b, e_{1a}	$\emptyset$	0.3182	0.2383
c_2	Z, e_3	0.3490	0.1848
X	b', e_{2a}	0.3798	0.1808
X	b', g', k_2	0.3818	0.1745
X	c_2, b^*, g'	0.3884	0.1722
X	c_2, k_2, g'	0.3902	0.1689
X	c_2, k_2, v^*	0.4042	0.1615

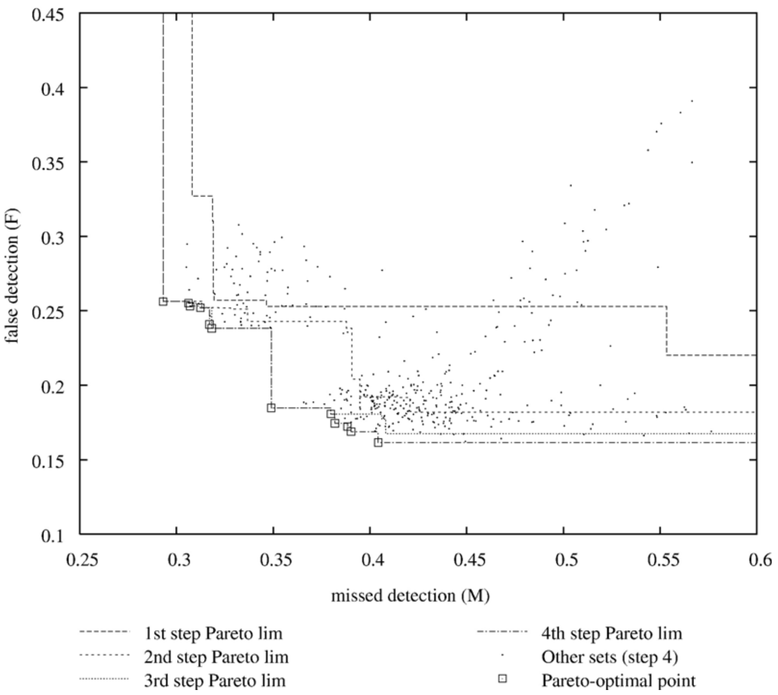

Fig. 3. Close-up of the scatter plot of the segmentation quality measures after the fourth step of the SFS, showing the Pareto-optimal parameter sets, the Pareto limit (all points in the upper-right side of this limit are worse than some Pareto-optimal parameter). The Pareto limit for the previous SFS steps is also shown. The remaining parameter sets also shown as smaller dots.

interest of using derived colour channels, or polar transformations. Of the 12 optimal parameter sets of Table 5, only one, that with $R_1=\{g', b\}$, $R_2=\emptyset$, does not contain any (non-trivially) derived or transformed colour channel. This is shown graphically in Fig. 4: sets without texture are clearly better performing.

It may seem counter-intuitive at first to notice that textural features offer no improvement over radiometry for

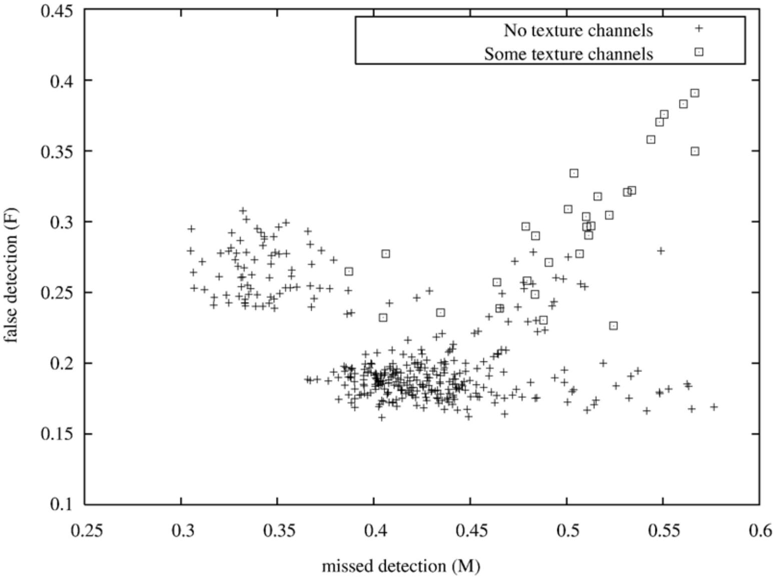

Fig. 4. Scatter plot of the segmentation quality measures after the fourth step of the SFS, separating test sets which use at least one texture feature and those that use none. Other steps have qualitatively similar results.

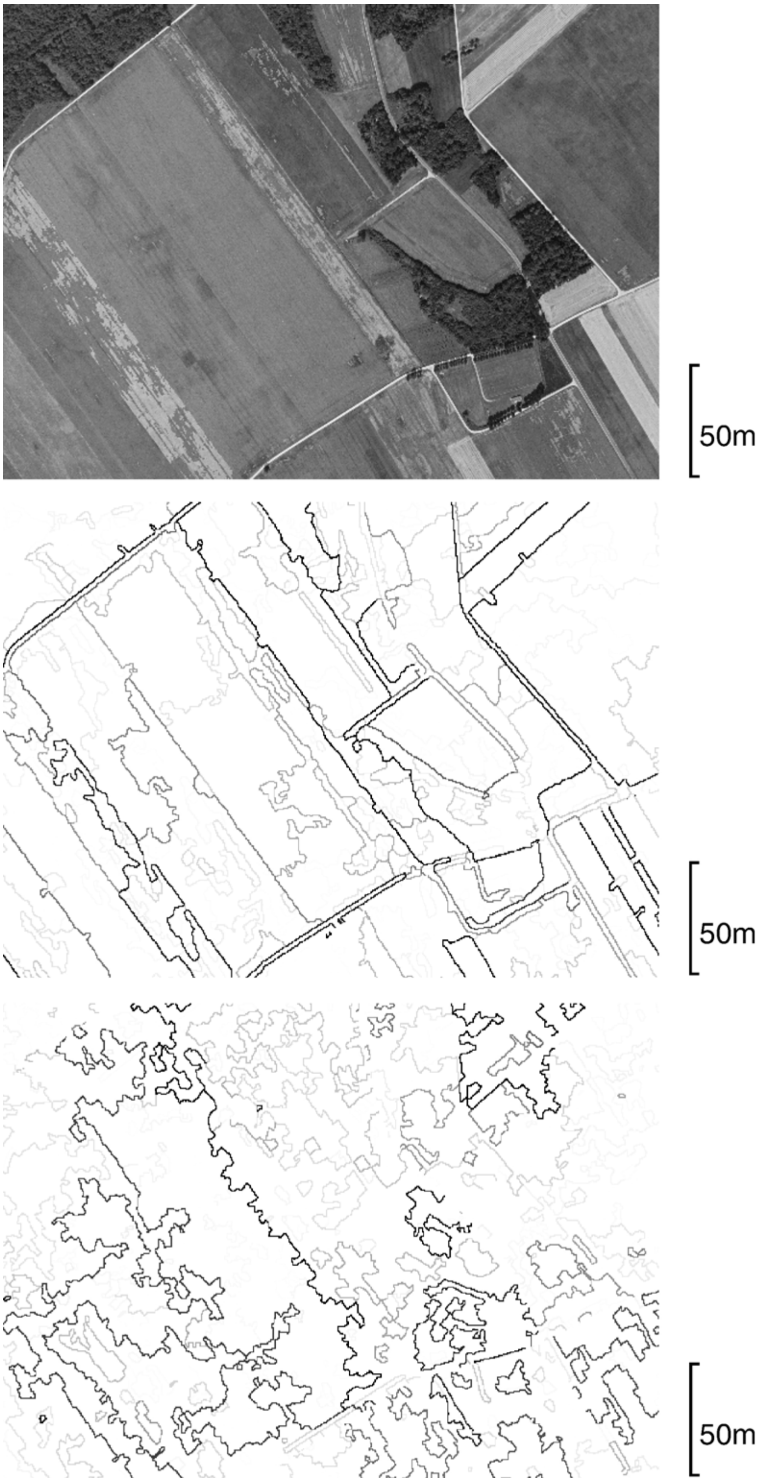

Fig. 5. Part of the test site and segmentation results. Top: input image. Center: segmentation using one of the optimal feature sets. Bottom: segmentation using one of the badly-performing feature sets. See the text for details on the parameter sets.

segmentation purposes, although we are reassured by the fact that Vansteenkiste et al. (2004) obtain similar results. It may be argued that, since textural features are inherently defined on neighbourhoods and not individual pixels, their positional accuracy is less than for radiometric data. Although texture may play a role in terrain classification, it may be that radiometry is enough for segmentation—as long as it is not necessary to determine the terrain types being separated—and that texture's inferior positional accuracy makes it less useful for segmentation.

For illustration, Fig. 5 shows some image segmentations. The input image is at the top. The center image is a segmentation with the selected optimal set, with quality measures $M=0.3490$ and $F=0.1848$. Darker edges appear at coarser analysis scales in the multiscale segmentation, and are therefore supposed to correspond —if the parameter set is well chosen—to more salient edges in the source image. The bottom image shows a segmentation with a badly-performing parameter set, with $M=0.7721$ and $F=0.6471$. Note that, whereas darker edges in the first segmentation do correspond to more salient edges in the source image, edge darkness in the second segmentation is less correlated with saliency, hence the worse quality measures. The badly-performing parameter set uses the green channel g as R_1 and, as R_2, the region-based version of the scale-orientation histogram parameter (see Section 4.2.6) referred to as $\sigma_{\theta,i}$ in Zhou et al. (2003).

6. Conclusion

Image segmentation is usually the first step of an image understanding process. Segmentation is supposed to partition an image into semantically significant objects, although in practice it is just required to obtain homogeneous patches.

While segmentation is usually performed on the raw radiometry channels, it could be that using transformed colour spaces or textural features gives improved results. To explore this question, we have performed an experiment with several transformed colour spaces and textural features which may be adequate for the problem of segmenting a high-resolution aerial image of rural areas. We have suggested a way of dealing with the fact that some colour and textural features are angular in nature. We have tested a large number of combinations of input channels, in order to determine which raw radiometry channel, transformed colour channel, or texture features give best performance for the specific problem of high-resolution aerial images of rural areas. The conclusion is that texture, which at first thought seemed promising, does not actually improve the results. We believe that this is because of the non-local nature of texture. On the other hand, using transformed colour channels gives better results than using the raw red, green, and blue channels.

In particular, we have seen that by exploring alternate input channels for the segmentation algorithm we have improved the segmentation quality from (0.3722, 0.2569) (with $R_1=\{r, g, b\}$, $R_2=\emptyset$) to (0.3490, 0.1848) (with $R_1=\{c_2\}$, $R_2=\{Z, e_3\}$). Although this particular result is likely not to be extrapolable to images other than high-resolution aerial images of rural areas, it shows that, in any particular application, it may be interesting to check whether transformed channels may give better results.

References

Angulo, J., Serra, J., Hanbury, A., 2003. Morphologie mathématique et couleur. Talk at the GDR ISIS colloquia. ENST, Paris, France. May.

Baillard, C., Oct. 1997. Analyse d'images aériennes stéréo pour la restitution 3-d en milieu urbain. Ph.D. thesis, ENST, Paris, France.

Berens, J., Finlayson, G.D., 2000. Log-opponent chromaticity coding of colour space. Proceedings International Conference on Pattern Recognition, vol. 1. IAPR, Barcelona, Spain, pp. 206–211. Sep.

Bülow, T., Sommer, G., 1998. Quaternionic Gabor filters for local structure classification. Proceedings International Conference on Pattern Recognition, vol. 1. IEEE Computer Society, Brisbane, Australia, pp. 808–810. Aug.

Cheng, H.D., Jiang, X.H., Sun, Y., Wang, J., 2001. Color image segmentation: advances and prospects. Pattern Recognition 34 (12), 2259–2281.

Faugeras, O., 1979. Digital color image processing within the framework of a human visual model. IEEE Transactions on Acoustics, Speech, and Signal Processing 27 (4), 380–393.

Garnier, A., Sep. 2002. Modélisation de textures dans les images aériennes. Master's thesis, École Supérieure en Sciences Informatiques, 930, rue des Colles; 06903 Sophia-Antipolis, France.

Gawthrop, L.C. (Ed.), 1970. The Administrative Process and Democratic Theory. Houghton Mifflin Co.

Geman, S., Geman, D., 1984. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence 6 (6), 721–741.

Guigues, L., 2000. Un algorithme pour la détection de zones de végétation sur images aériennes. Oral communication to GT6.2 "Complex systems for image analysis" of GdR ISIS. ENST, Paris, France. Oct.

Guigues, L., Le Men, H., Cocquerez, J.-P., 2003. Scale-sets image analysis. Proceedings International Conference on Image Processing, vol. 2. IEEE, Barcelona, Spain, pp. 45–48. Sep.

Guigues, L., Cocquerez, J.P., Le Men, H., 2006. Scale-Sets Image Analysis. International Journal of Computer Vision 68 (3), 289–317 (July).

Jolion, J.-M., Montanvert, A., 1992. The adaptive pyramid, a framework for 2D image analysis. CVGIP. Image Understanding 55 (3), 339–348 May.

Koepfler, G., Lopez, C., Morel, J., 1994. A multiscale algorithm for image segmentation by variational methods. SIAM Journal on Numerical Analysis 31 (1), 282–299 Feb.

Kruizinga, P., Petkov, N., 2000. A nonlinear texture operator specialised in the analysis of dot-patterns. Proceedings International Conference on Pattern Recognition, vol. 1. IAPR, Barcelona, Spain, pp. 197–201. Sep.

Leclerc, Y., 1989. Constructing simple stable descriptions for image partitioning. International Journal of Computer Vision 3 (1), 73–102.

Liu, J., Yang, Y.-H., 1994. Multiresolution color image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (7), 689–700.

MacLennan, B., 1991. Gabor representations of spatiotemporal visual images. Tech. Rep. CS-91-144. Computer Science Department, University of Tennessee, Knoxville, TN 37996, USA. Sep.

Marr, D., 1982. Vision. Freeman and Co.

Mumford, D., Shah, J., 1989. Optimal approximations by piecewise smooth functions and associated variational problems. Communications on Pure and Applied Mathematics 17 (4), 577–685.

Olaja, T., Pietikäinen, M., Mäenpää, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7), 971–987 Jul.

Pareto, V., 1906. Manuale d'economia politica. Milano, Italy.

Peleg, S., Naor, J., Hartley, R., Avnir, D., 1984. Multiple resolution texture analysis and classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 6 (4), 518–523 Jul.

Salembier, P., Garrido, L., 2000. Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. IEEE Transactions on Image Processing 9 (4), 561–576 Apr.

Segui Prieto, M., Allen, A.R., 2003. A similarity metric for edge images. IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (10), 1265–1272 Oct.

Talukdar, D., Acharya, R., 1995. Estimation of fractal dimension using alternating sequential filters. Proceedings International Conference on Image Processing. IEEE, Washington, D.C., USA, pp. 231–234. Oct.

Tao, Y., Lam, E.C.M., Tang, Y.Y., 2000. Extraction of fractal feature for pattern recognition. Proceedings International Conference on Pattern Recognition, vol. 2. IAPR, Barcelona, Spain, pp. 527–530. Sep.

Trias-Sanz, R., 2005. A metric for evaluating and comparing hierarchical and multi-scale image segmentations. Proceedings of the International Geosciences And Remote Sensing Symposium, vol. 8. Seoul, South Korea, pp. 5647–5650. Jul.

Van de Wouwer, G., Scheunders, P., Livens, S., Van Dyck, D., 1999. Wavelet correlation signatures for color texture characterization. Pattern Recognition 32 (3), 443–451 Mar.

Vansteenkiste, E., Schoutteet, A., Gautama, S., Philips, W., 2004. Comparing color and textural information in very high resolution satellite image classification. Proceedings International Conference on Image Processing. IEEE, Singapore, pp. 3351–3354. Oct.

Zhou, J., Xin, L., Zhang, D., 2003. Scale-orientation histogram for texture image retrieval. Pattern Recognition 36 (4), 1061–1062.