

# Fast Road Network Extraction in Satellite Images Using Mathematical Morphology and Markov Random Fields

**Thierry Géraud**

*EPITA Research and Development Laboratory, 14-16 Rue Voltaire, 94276 Le Kremlin-Bicêtre Cedex, France  
Email: thierry.geraud@lrde.epita.fr*

**Jean-Baptiste Mouret**

*EPITA Research and Development Laboratory, 14-16 Rue Voltaire, 94276 Le Kremlin-Bicêtre Cedex, France  
Email: jean-baptiste.mouret@lrde.epita.fr*

*Received 1 September 2003; Revised 27 August 2004*

We present a fast method for road network extraction in satellite images. It can be seen as a transposition of the segmentation scheme “watershed transform + region adjacency graph + Markov random fields” to the extraction of curvilinear objects. Many road extractors which are composed of two stages can be found in the literature. The first one acts like a filter that can decide from a local analysis, at every image point, if there is a road or not. The second stage aims at obtaining the road network structure. In the method, we propose to rely on a “potential” image, that is, unstructured image data that can be derived from any road extractor filter. In such a potential image, the value assigned to a point is a measure of its likelihood to be located in the middle of a road. A filtering step applied on the potential image relies on the area closing operator followed by the watershed transform to obtain a connected line which encloses the road network. Then a graph describing adjacency relationships between watershed lines is built. Defining Markov random fields upon this graph, associated with an energetic model of road networks, leads to the expression of road network extraction as a global energy minimization problem. This method can easily be adapted to other image processing fields, where the recognition of curvilinear structures is involved.

**Keywords and phrases:** Markov random field, mathematical morphology, pattern recognition, satellite image.

## 1. INTRODUCTION

Many different methods have been proposed to segment curvilinear structures in 2D images, some of them being dedicated to road network extraction. We recall some of them which are, to our humble opinion, the most promising ones:

- (i) tracking by active testing by [1];
- (ii) unifying snakes, region growing, and energy/Bayes/MDL, so-called region competition, by [2];
- (iii) defining Markovian field on a set of segments by [3];
- (iv) dynamic programming for saliency optimization by [4];
- (v) using a Markov point process by [5].

These methods suffer from drawbacks. The saliency approach does not rely on a global optimization process. Tracking-like approaches plainly cannot take into account features extracted from image regions and require a starting point; these approaches are thus limited to rather easy segmentation problems. Markovian approaches are often computationally expensive due to the high number of primitives—small segments—they have to handle.

Finally, region competition is also an expensive approach, where both the region and variational flavors are not often pertinent when objects are not regions.

In this paper, we detail the detection method for road network previously introduced in [6], which overcomes these drawbacks. Compared to the previous contribution, the present document describes our road model, adds algorithmic considerations, and shows how the method can be adapted to other problems in pattern recognition.

This paper is organized as follows. First, Section 2 recalls some classical tools used in pattern recognition. Then, Section 3 details the method we propose. Section 4 discusses our results and compare them to the ones obtained using a Markov point process. Finally, Section 5 concludes and gives perspectives.

## 2. PRELIMINARIES

### 2.1. Watershed transform and minima removing

The watershed transform (WT) is the classical method used for image segmentation in the field of gray-scale

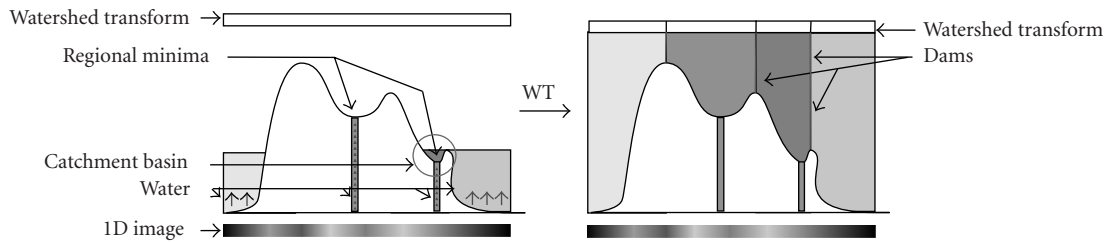


FIGURE 1: Watershed transform of a 1D gray-scale image. We begin by piercing the regional minima of the surface then we slowly flood it with water. Where water coming from different basins meets, a dam is built. The resulting watershed transform is the set of dams.

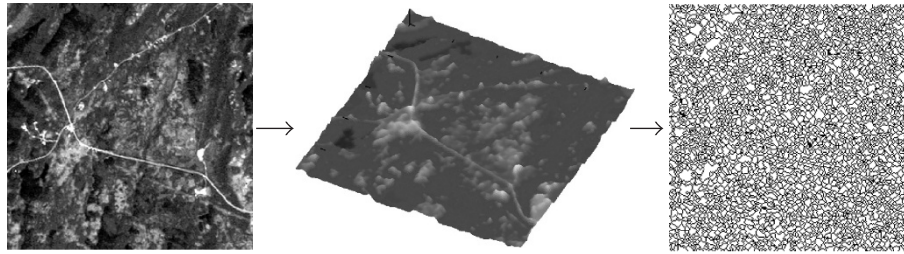


FIGURE 2: Watershed transform of gray-scale images; the image is viewed as a 3D landscape and the watershed lines are the crest lines.

mathematical morphology. Figure 1 recalls the basic principle of the watershed transform by immersion.

A 2D gray-scale image is considered as a 3D landscape (Figure 2). In this case, the watershed lines are the *crest lines* and they separate adjacent catchment basins, that is, regions. Since images have numerous minima, applying the watershed transform to an image leads to an over-segmentation.

A classical algorithm to suppress minima in images is the morphological closing operator. When there is no prior information about the shape of image objects, closing is usually performed with a disk as structural element in order to preserve isotropy. However, artifacts appear in resulting images: crest lines can strongly move when one wants to remove many minima, that is, when filtering strength (i.e., the disk radius) increases.

Conversely, an area closing operator does not present this drawback. The underlying idea of this operator, defined by Vincent in [7], is to remove connected components whose area is smaller than a parameter  $\lambda$ ,  $\lambda \in \mathbb{N}$ . This morphological filter is said to be “shape preserving”: it acts on connected components thus it does not change the shape of the structures in the image. The gray-scale formulation relies on the threshold superposition principle [7]; a fast implementation of this operator is provided by [8].

Figure 3 illustrates the contour shifting/unshifting properties of both “classical” and area closing operators. Starting from the classical house image, closing operators are applied to its gradient norm image (GNI); the negatives of the results are depicted by images in Figures 3a and 3b. The watershed transform algorithm is then applied, which respectively leads to images in Figures 3c and 3d. Please note that these segmentation results contain the same number of regions. However, contours are shifted when the classical closing is

involved, which is not the case with the area closing. Moreover, in the former case, regions have more uniform sizes and are spread more uniformly over the image space than in the latter case. This is another drawback since we prefer segmentations that are more adapted to original image data.

## 2.2. Region adjacency graph and Markov random fields

A presently common framework to segment an image  $I$  or to extract objects from  $I$  is based on the WT; it can be summarized as follows.

- (1) An image  $G$  of the gradient norm of  $I$  is computed. Contours in the GNI,  $G$ , have high intensity values, whereas regions have low intensity values.
- (2) The WT is applied to  $G$  which results in a partition of  $I$  into regions. The watershed lines follows crest lines of  $G$ , that is, objects contours. This partition,  $P$ , is an over-segmentation since  $G$  contains a greater number of minima than the effective number of objects/regions to segment.
- (3) The region adjacency graph (RAG) is extracted from  $P$ . In such a graph, a node corresponds to a region (more precisely, a catchment basin) and an edge between two nodes indicates that these regions are adjacent. Extra information may be put into the graph like statistical estimations concerning regions of  $I$  which are then enclosed in graph nodes, or saliency values of contours estimated from  $I$  and added to graph nodes.
- (4) The last step aims at grouping regions according to given criteria in order to get a final segmentation. To that aim, a Markov random field (MRF) is defined onto the RAG and the segmentation process is handled by a Markovian relaxation.

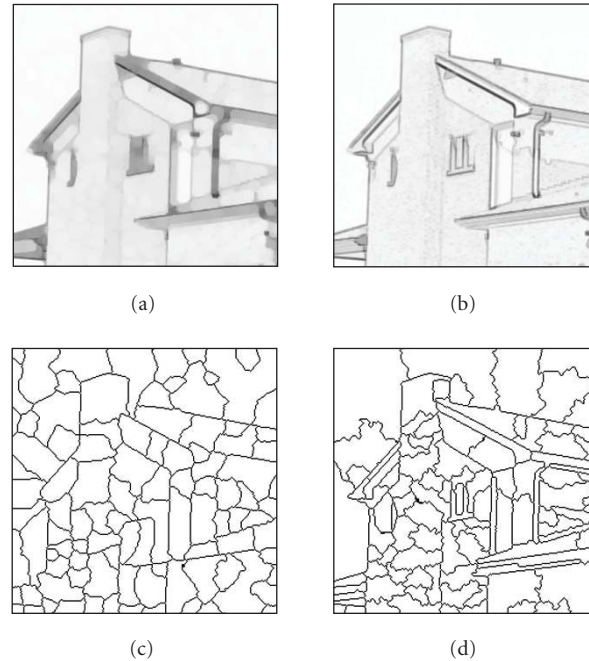


FIGURE 3: Watershed transform results with the same final number of regions. (a) Closing of GNI (inverted) with a disk ( $r = 4, 3$ ). (b) Area closing of GNI (inverted) ( $a = 50$ ). (c)  $W$  obtained from image (a). (d)  $W$  obtained from image (b).

This framework is powerful since it remains general—it can be applied to various image segmentation problems—and since the final segmentation results from a global process on high-level image primitives (regions in that case). Moreover, it enables operational segmentations even when images are over-sized and when objects are difficult to segment; for instance, Géraud et al. [9] succeeded in segmenting internal brain structures from magnetic resonance images. This framework has been discussed by many authors (see, e.g., [10, 11, 12, 13]), and a multiscale version of this framework has been proposed in [14].

### 3. PROPOSED METHOD

The method we propose is composed of four steps, derived from the ones described in Section 2.2 (see Figure 4 for an overview of the proposed framework). They are illustrated with a small part ( $700 \times 380$  pixels) of a Landsat image from St. Johns city, Canada, having a 25 m resolution and 7 spectral channels. This original image is under the following copyright: “© 2000. Government of Canada with permission from Natural Resources Canada”; it can be fetched from <http://geogratis.cgdi.gc.ca/>.

#### 3.1. Preprocessing

The preprocessing step of our method aims at providing a potential image. Having a high potential value assigned to a point means that this point is very likely located in the center of a road. On the contrary, having a low potential value means that this point may not belong to a road. A potential value can be either a fuzzy degree (a membership value,

a possibility or necessity value, or a belief or plausibility value), a probability, or a value resulting from estimators of road presence.

In particular, the literature proposes a lot of road extractor filters. Some of them have been designed to a specific type of satellite images, for a given imagery system, a given resolution, and sometimes a given kind of scene—urban, rural, and so forth. Some other filters are more or less adaptable to the variability that can be found in satellite images. Depending on the resolution of images to be processed, these filters are line detectors such as in [3, 15, 16] or more elaborated filters when roads appear to be thick such as in [17, 18]. In all cases, we can derive from these filters a gray-level image corresponding to the magnitude of their response, which gives us a potential image to work with.

In the case of low-resolution images, pixels with high potential values indicate the possible presence of a road, and the expected road network is one pixel thick. In the case of high-resolution images, the highest values of filter response are observed in the middle of roads. The road network topology can then be represented by a one-pixel thick object, passing through the middle part of roads. Please note that we do not aim at getting any local attributes of roads such as its width, and so forth.

The main property of such potential images is that, in both cases, the road network is located on crest lines of the potential image.

As an illustration, we have chosen a very simple potential image, the red channel ( $0.63\text{--}0.69\ \mu\text{m}$ ); it is depicted in Figure 5. We stress again that more elaborated potential images can also be used.

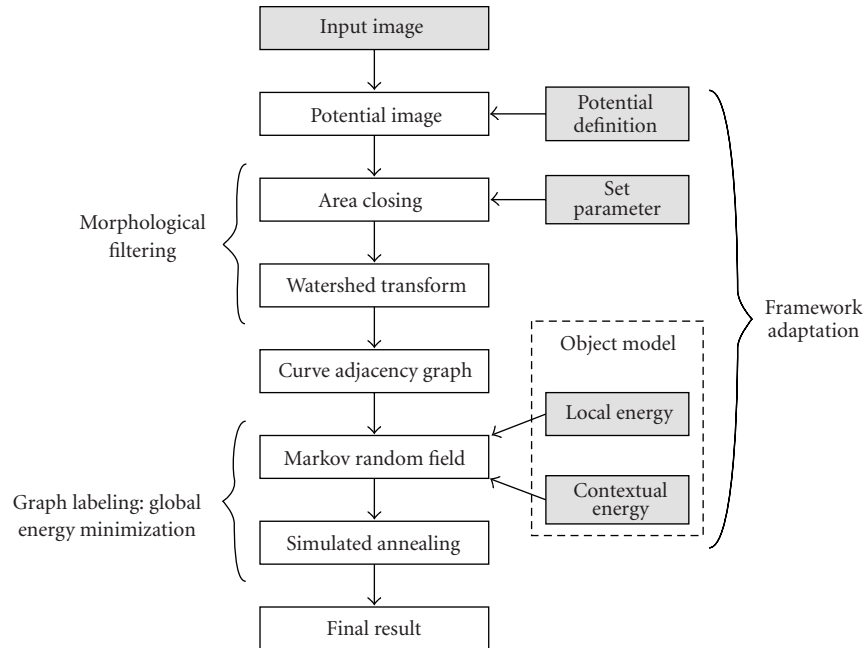


FIGURE 4: Overview of the proposed framework; gray boxes symbolize parameters to be defined.

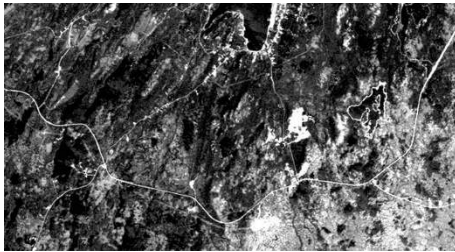


FIGURE 5: Gray-level potential field.

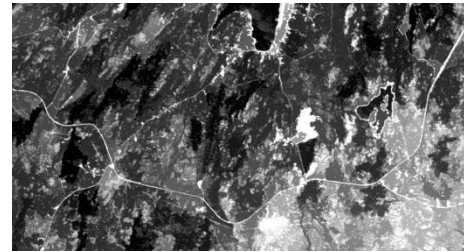


FIGURE 6: Result of morphological closing (area = 500).

### 3.2. Filtering

The filtering step consists in computing an area closing of the potential image and then running the WT (Section 2.1). The potential image, once closed (Figure 6), has much fewer minima than the “original” potential image (Figure 5) while properly retaining crest lines location. Therefore, the resulting watershed lines (Figure 7) include the road network.

In Figure 7, the watershed lines resulting from different values of area are depicted. Multiscale properties of this morphological filtering can be observed: new curves (features) do not appear when area (scale) increases and a feature which is present at a given scale (watershed lines obtained with a given area) is still present at a lower scale (in the watershed lines obtained with a smaller area). This property is very important since the only parameter of this filtering step is the *area*; even with a large value of the area, we are guaranteed to have important roads included in watershed lines.

Please note that applying the WT to segment curvilinear objects is not an original contribution. This idea is at least

mentioned in [19],<sup>1</sup> and in [20]. However, the original part of our work comes from the use of a graph built from the result of the WT. This new step is explained in the following section.

### 3.3. Curve adjacency graph

Using the watershed lines, we build a curve adjacency graph (CAG). A node of this graph (circles on Figure 8) represents a shed, that is, a connected part of the watershed lines separating two adjacent basins. An edge (gray lines on Figure 8) is drawn between two nodes/sheds if one end of the first shed is connected with an end of the second one through the watershed lines.

Edges coming to a node are divided into two groups, depending on the shed end they are connected to (Figure 8a).

<sup>1</sup>See “3D segmentation of brain structures from MR images using morphological approaches,” English version of part III-7 and Appendix C of [19]; it is electronically available from <http://www.lrde.epita.fr/people/thierry.geraud/publications.html>.

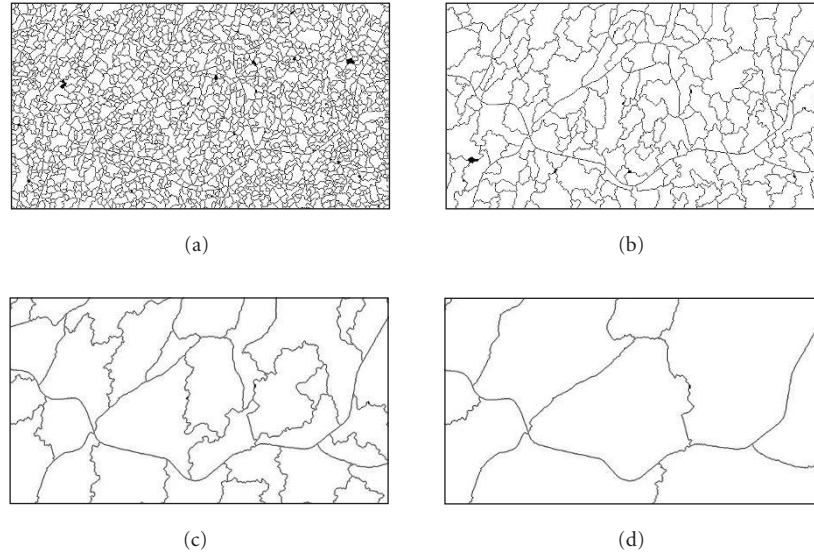


FIGURE 7: Different strengths of filtering. (a) area = 10. (b) area = 100. (c) area = 1000. (d) area = 10000.

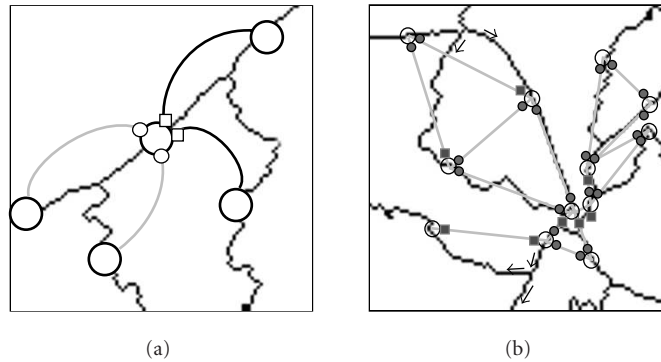


FIGURE 8: (a) For every node, edges are divided into two groups (gray and black lines) depending on the shed end they are connected to. (b) Complete CAG.

This distinction, symbolized by disk-shaped and square-shaped anchors in Figure 8, permits to properly handle in the next step (Section 3.4) some geometrical constraints about the road network.

Our algorithm to build this graph starts from the result of a WT *with labels*. This watershed variant assigns a unique label (an integer) to each region; the label “white” is assigned to watershed lines. The basic idea of this algorithm consists in visiting all the white pixels from neighbors to neighbors and in creating new nodes when reaching a crossing point. During this search, a label is assigned to each point of the node, it is then used to link nodes together.

We denote by  $W$  the set of pixels which belong to watershed lines and by  $V_p$  the neighborhood of a point  $p \in W$ .

*Definition 1* (crossing point). Let  $N_a$  denote the number of adjacent regions of a point  $p \in W$ .  $p$  is a crossing point if and only if  $N_a > 2$ .

$N_a$  is computed by comparing the label assigned to each pixel in the neighborhood of  $p$ . Figure 9 contains the values of  $N_a$  for an example image.

Using this definition, it is now easy to describe an iterative algorithm which creates the CAG (Algorithm 1). The presented version uses seeds to keep track of pixels where it has to create new nodes. When it reaches a crossing point, seeds are emitted for each neighbor, and when a seed is hit, a new node is created. Figure 10 shows the result of the node creation step, circles are seeds and crosses denote crossing points.

Multiple crossing points and crossing points with too many neighbors may cause problems during the graph building process. Examples of these kinds of configuration are circled in Figure 11.

The multiple crossing points configuration occurs when two crossing points are adjacent; in this case Algorithm 1 would create a useless node and nodes would not be linked

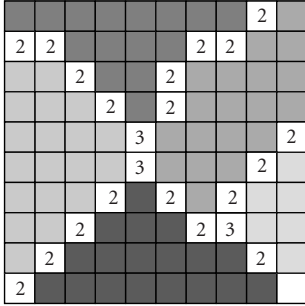


FIGURE 9: Number of adjacent regions.

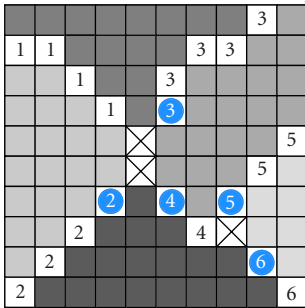


FIGURE 10: Labeling of the sheds.

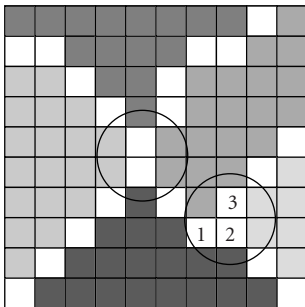


FIGURE 11: Potential difficulties.

correctly. The selected solution is not to emit any seed at a crossing point  $p_1$  if another crossing point  $p_2$  is in its neighborhood and to add the neighbors of  $p_1$  in the neighbors list of  $p_2$ . Thus the seed will be emitted once.

In the second potentially dangerous configuration, the visiting order can lead to the omission of a link. Indeed if, for example, the algorithm visits—in this order—pixels labeled “1,” “3,” and “2” of Figure 11, the adjacent labels of the crossing point (pixel “2”) would be (following the labeling of Figure 10) only “4” and “6”: we would have lost the information needed to create the edge with the node “5.” We now suppose that the algorithm visits pixels “1,” “2,” and “3,” the adjacent labels of “2” would be “4,” “5,” and “6”: we would be able to link the node “5” with its adjacent nodes. The retained solution is to sort neighbors in such a way that the algorithm visits crossing points before any other neighbor.

Figure 12 shows the graph built using our algorithm.

```

(1) Let  $I$  denote the input image,
 $T$  a stack of points (will be used to store the pixels to
explore),  $E$  a 2D array of booleans initialized to false
(will be used to store the seeds),
 $L$  a 2D array of nodes (will be used to store the label of
each of the points of the watershed lines),
 $n$  a node,
 $p$  the first pixel of a watershed line,
 $G$  the resulting graph
(2) push( $T, p$ )
(3) /* create nodes */
(4) while  $T \neq \emptyset$  do
(5)  $p \leftarrow \text{pop}(T)$ 
(6)  $L[p] \leftarrow \text{label}(n)$ 
(7) if  $p \in E$ , then
(8)  $n \leftarrow \text{new\_node}(G)$ 
(9) end if
(10) let  $\text{Neigh}_W$  denote the set of neighbors of  $p$  in  $W$ 
(11) for all  $s \in \text{Neigh}_W$  do
(12) if  $E[s] = \text{true}$ , then
(13)  $E[s] = \text{false}$ 
(14) end if
(15) end for
(16) push( $T, \text{Neigh}_W$ )
(17) if  $p$  is a crossing point, then
(18) for all  $s \in \text{Neigh}_W$  do
(19)  $E[s] \leftarrow \text{true}$ 
(20) end for
(21) end if
(22) end while
(23) /* create edges */
(24) for all crossing points  $p$  do
(25) let  $V_p$  denote the set of neighbors of  $p$  in  $W$ 
(26) for all  $s_i \in V_p$  do
(27) for all  $s_j \in V_p$  do
(28) if  $i \neq j$ , then
(29) add_edge( $G, L[s_i], L[s_j]$ )
(30) end if
(31) end for
(32) end for
(33) end for

```

ALGORITHM 1: Building a CAG: basic algorithm.

### 3.4. Markovian relaxation

Extracting road networks now turns out to be a graph labeling problem that can be solved by modeling it using MRFs. Two types of information can be used to distinguish whether a node of the CAG is a road or not. The first one concerns the shape of the considered node and the values of the pixels of the potential image. For instance, we can assume that roads are straight lines with a high and almost constant potential value along the shed. The way roads are organized constitutes

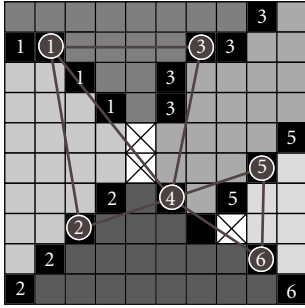


FIGURE 12: Built graph.

another type of information which can be used to recognize them. For instance, a road is, most of the time, linked to another road and does not stop abruptly. This idea can be expressed locally since the observation of the adjacent sheds of a node  $s$  is enough to know if  $s$  is realistically linked to the road network.

As we will later see in more detail, this feature of our graph labeling problem made it suitable to be modeled with a MRF. MRFs have been widely used in image processing since 1984 (see [21]); our use is close to Tupin's one [3] who used them to label a graph made up of small straight segments extracted from SAR images in order to extract road networks.

We first recall some basic results about MRFs. We denote by  $S$  the set of sites, by  $Y$  the observation field, by  $X$  the result field, by  $X_s$  and  $Y_s$  their respective restriction to a given node  $s$ , by  $x_s$  and  $y_s$  their realization, and by  $X_{N_s}$  the restriction of  $X$  to the neighborhood of  $s$ . The variable  $x_s$  has a Boolean realization where 1 means *is\_road* and 0 means *not\_road*.

**Definition 2** (MRF).  $Y$  is a MRF if and only if

$$\begin{aligned} P(Y_s = y_s | X_r = x_r, r \in S, r \neq s) \\ = P(Y_s = y_s | x_r, r \in Y_{N_s}). \end{aligned} \quad (1)$$

**Definition 3** (Gibbs distribution). Let  $U(x)$  denote an energy function. A Gibbs distribution takes the following form:

$$P(X = x) = \frac{1}{Z} \exp(-U(x)), \quad (2)$$

where  $Z$  is a normalizing constant called the *partition function*. The energy  $U(x)$  can be defined as the sum of *clique potentials*  $V_c(x)$  over all possible cliques  $C$ :

$$U_c(x) = \sum_{c \in C} V_c(x). \quad (3)$$

In this case, we call a *Gibbs random field* (GRF) a random field whose distribution obeys a Gibbs distribution:

$$\begin{aligned} P(X = x) &= \frac{1}{Z} \exp(-U_c(x)) \\ &= \frac{1}{Z} \exp\left(-\sum_{c \in C} V_c(x)\right). \end{aligned} \quad (4)$$

Finally, the Hammersley-Clifford theorem establishes the equivalence of MRF and GRF.

We can now switch back to our labeling problem. The labeling is an inverse problem: finding a realization  $x$  of  $X$ , assigning the labels to the nodes, using  $y$ , and the input data. We can use the *maximum a posteriori* (MAP) and search for the configuration  $\hat{x}$  which maximizes the probability  $P(X = x | Y = y)$ .

Using Bayes' theorem, we can write

$$P(X = x | Y = y) = \frac{P(Y = y | X = x)P(X = x)}{P(Y = y)}. \quad (5)$$

$P(Y = y)$  is a constant which does not depend on the realization of  $x$ . We suppose that in the observation field every node is independent:

$$P(Y = y | X = x) = \prod_s P(Y_s = y_s | X_s = x_s). \quad (6)$$

As  $X$  is a MRF, we can write  $P(X = x) = (1/Z) \exp(-U(x))$ . Combining (2), (5), and (6) we get

$$\begin{aligned} P(X = x | Y = y) &\propto P(Y | X)P(X) \\ &\propto e^{\ln P(Y | X) - U(X)} \\ &\propto e^{-\mathcal{U}(x | y)}, \end{aligned} \quad (7)$$

where

$$\mathcal{U}(x | y) = \sum_{s \in S} -\ln P(y_s | x_s) + \sum_{c \in C} V_c(x). \quad (8)$$

Consequently the a posteriori distribution is a Gibbs distribution and from the Hammersley-Clifford theorem the conditional field  $P(X | y)$  is a MRF. We are looking for the configuration which maximizes  $P(X = x | Y = y)$ , therefore the one which minimizes  $\mathcal{U}(x | y)$ : the road extraction is expressed as an energy minimization problem.

By defining  $P(y_s | x_s) = \exp(U_D(y_s | x_s))$ , (6) and (8) lead to

$$P(X_s = x_s | Y_s = y_s) \propto \exp(-U_D(y_s | x_s) + U_c(Y_{N_s})). \quad (9)$$

The first energy term,  $U_D(y_s | x_s)$ , models a priori knowledge about roads; we will call it the *data energy*. As a node designates a small piece of the watershed lines, that is, a set of points, we can have measures associated with every node, such as the curvature of this piece of line, its "potential" mean value (measured from the potential image), its contrast with respect to adjacent regions, its saliency [22], and so forth.

We first define a measure of the curvature. Let  $N$  be the number of points in a shed, we assume that the list of points is ordered (from the start of the shed to its end). The local

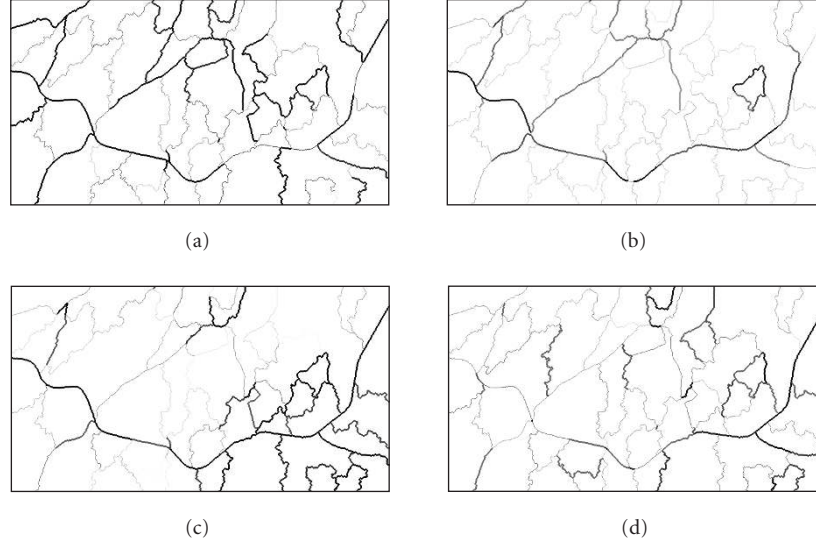


FIGURE 13: Values  $U_{\sigma_{\kappa_s}}$ ,  $U_{\bar{c}_s}$ ,  $U_{\bar{p}_s}$ , and  $U_{\Delta \bar{p}_s}$  associated with each shed (darkest sheds have the lowest energy); sheds with an energy lower than the mean energy are two pixels wide. (a) Variance deviation of the local curvature:  $U_{\sigma_{\kappa_s}}$ . (b) Mean contrast:  $U_{\bar{c}_s}$ . (c) Mean potential:  $U_{\bar{p}_s}$ . (d) Mean variation of potential:  $U_{\Delta \bar{p}_s}$ .

discrete curvature at the  $i$ th point of the list  $p_i = (x_i; y_i) \in \mathbb{N}^2$  can be computed using

$$\kappa_i = \sqrt{(y_{i-1} - 2y_i + y_{i+1})^2 + (x_{i-1} - 2x_i + x_{i+1})^2}. \quad (10)$$

Curvature is not sufficient to characterize the shape of a road: for example, a portion of a circle could be a regular road but it has a constant nonzero curvature. On the other side, highly oscillating sheds are not roads and have a nonzero curvature. Another criterion is the *variance*  $\sigma_{\kappa}$  of the curvature  $\kappa$ :

$$\sigma_{\kappa} = \frac{1}{N} \sum_{i=1}^N (\kappa_i - \bar{c}_i)^2, \quad (11)$$

where  $\bar{c}_i$  denotes the mean of the curvature for a shed.

This measure is combined with some other ones based mainly on the potential values  $p_i$  at point  $i$ . In our model the data energy  $U_D(x)$  for a node  $x$  is a function of

- (i) mean potential  $\bar{p}_s = (1/N) \sum_{i=1}^N p_i$  along the shed;
- (ii) mean variation  $\Delta \bar{p}_s = (1/N) \sum_{i=0}^N |p_i - p_{i-1}|$  of potential along the shed;
- (iii) mean curvature  $\bar{\kappa}_s = (1/N) \sum_{i=0}^N \kappa_i$  of the shed;
- (iv) variance  $\sigma_{\kappa_s}$  of the local curvature;
- (v) mean contrast between each point of the shed and its neighbors:

$$\bar{c}_s = \frac{1}{N} \sum_{i=1}^N \left( \frac{1}{\text{card}(V_i)} \sum_{v \in V} (p_i - p_v) \right), \quad (12)$$

where  $V_i$  denotes the neighborhood of point  $i$ .

Energy terms are then built using these measures. For instance if  $\bar{\kappa}$  and  $\bar{p}$  denote the mean curvature and the mean potential for all the sheds, we can build  $U_{\kappa}$  and  $U_p$  as follows:

$$U_{\kappa}(x_s | y_s) = \begin{cases} \kappa_s & \text{if } y_s = 1, \\ \bar{\kappa} & \text{otherwise,} \end{cases} \quad (13)$$

$$U_p(x_s | y_s) = \begin{cases} 1 - p_s & \text{if } y_s = 1, \\ 1 - \bar{p} & \text{otherwise.} \end{cases}$$

Using these notations, the data energy term can be written as

$$U_D(x_s | y_s) = a_1 \cdot U_p + a_2 \cdot U_{\Delta p} + a_3 \cdot U_{\kappa} + a_3 \cdot U_{\sigma_{\kappa}} + a_4 \cdot U_{\bar{c}_s}, \quad (14)$$

where  $a_1$ ,  $a_2$ ,  $a_3$ , and  $a_4$  denote coefficients which need to be adjusted. Values of  $U_{\sigma_{\kappa_s}}$ ,  $U_{\bar{c}_s}$ ,  $U_{\bar{p}_s}$ , and  $U_{\Delta \bar{p}_s}$  for the image of Figure 5 are depicted in Figure 13. In order to obtain a low energy for each shed belonging to the road network, we need the four criteria. But the distinction power of  $U_{\sigma_{\kappa_s}}$  and  $U_{\bar{c}_s}$  is better than that of  $U_{\bar{p}_s}$  and  $U_{\Delta \bar{p}_s}$ . Consequently, best results are obtained when  $a_3$  and  $a_4$  are greater than  $a_1$  and  $a_2$ .

The second energy term,  $U_c(Y_{N_s})$  deals with labeling contextual information; we will call it the *prior energy*. We use a model rather close to the Potts model. We first recall the latter for a set of labels  $l \in \Lambda$  and a clique  $c$ :

$$U_c = \sum_{c=(i,j)} \beta \delta(l_i, l_j), \quad (15)$$

where

$$\delta(l_i, l_j) = \begin{cases} 1 & \text{if } l_i = l_j, \\ 0 & \text{otherwise.} \end{cases} \quad (16)$$



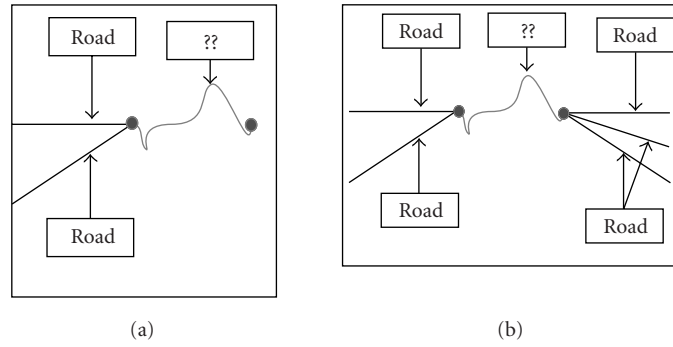


FIGURE 14: Prior energy: configurations with a good energy in Potts' model but undesirable in our road model.

TABLE 1

Configuration	Probability	Prior energy
	High	0.0
	Medium	0.2
	Low	1.0
	Low	1.0

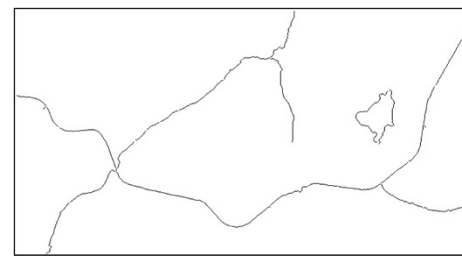


FIGURE 15: Final MRF labeling.

With  $\beta < 0$ , energy decreases when the neighbors of a node  $n$  have the same label as that of  $n$ .

To model a road network, we have a similar need: if a node is linked to two nodes labeled as “road” but has high data energy, we would like to label it as road. However some configurations like those of Figure 14 would have a low energy level in the Potts’ model but can not occur in a road model. This is why we need to distinguish connections coming from one side of the shed and those which come from the other side. Table 1 summarizes the different cases and the corresponding prior energy.

Finally, the simulated annealing algorithm [23] is employed to find a configuration of minimal energy. The metropolis sampler and a linear temperature decrease schedule have been used in our experiments.

Details about these classical algorithms can be found in any optimization textbook.

## 4. RESULTS

### 4.1. Commented results and implementation issues

We have validated our method on about a dozen of Landsat images. Applying the whole road extraction process to an image having  $2 \times 10^6$  pixels takes less than twenty seconds on a 1.7 GHz personal computer running GNU/Linux. A representative result is depicted in Figure 15.

While a rigorous comparative study would require a whole paper by itself, some clues allow us to hypothesize that our method is among the fastest and most accurate available. The morphological filtering step relies on the area closing and the WT, which are well-studied algorithms that have efficient implementations. The optimization step, which relies on the simulated annealing algorithm, can be slow if the graph used contains too many nodes. However, our framework builds—and that’s one of its main aspects—a small, though accurate, graph.

In particular, our method is sped up when the area parameter increases, since the CAG is becoming smaller. Filtering often leads to information loss, but with  $\text{area} = 500$  for our 25 m resolution test image, we obtain a very lightweight graph (about 126 nodes; see Table 2) and the filtering step only prevents us from extracting loops in road network whose area is less than  $0.3 \text{ km}^2$ . Our method is therefore rather insensitive to data simplification.

As a reference, the framework proposed by Tupin et al. [3] typically works on graphs made up of tens of thousands of nodes. This leads to a slower optimization process. Similarly, Markov points processes like the one used in [24] (see Section 4.2) require a longer optimization step than the proposed method.

It should be noted that contrary to some other methods (for instance, in [25]), no attempt has currently been made to recover partially occluded roads. If the road followed by a watershed line stops, the latter will move towards the closest high potential (Figure 16), which has no reason to be correlated with the path of the occluded road. This situation leads

TABLE 2

Area	Number of sheds	Number of links
2000	33	51
1000	72	120
500	126	211
250	239	434
100	535	984
50	1038	1948
10	5074	9533

to sheds with a part corresponding to a road and a chaotic part, therefore the associated data energy is not significant.

A way to address this problem is to split nodes associated to an incoherent energy. For example, a node can have a low energy if we take into account its first ten points, and a high one for the remaining points. We could split this node in two, then create an artificial node which would link the first ten points with the closest shed in the direction they point to.

Another conceivable improvement is to add a postprocessing step to the framework. This step would “reconstruct” sheds which have an associated energy below a given threshold or create the artificial nodes previously described.

We have developed our method with our C++ image processing library Olena [26]. Olena is a free software under the GNU public licence (GPL) and can be downloaded from our web site <http://www.lrde.epita.fr>. Olena provides a wide range of objects:

- (i) images structures: 1D, 2D, or 3D images and graphs;
- (ii) safe data types: integers, floating values and different color encodings;
- (iii) utility objects: points, iterators, and so forth.

Olena also provides fast implementation of algorithms and is a *generic* library. An algorithm is written once while accepting various input types. Olena can thus be used in different fields of image processing and pattern recognition. Source code of our method implementation is available on the Internet on our web site.

#### 4.2. Comparison with another approach

Figure 17 is the result of the method proposed in [24] applied to the image depicted in Figure 5. This method is based on stochastic geometry and reversible jump Monte Carlo Markov chains (RJMCMC) dynamic.

The authors considered that roads consist of a thin network which can be approximated by connected lines segments. They build a marked point process using the Candy model as a prior model and a likelihood term based on statistical hypothesis tests. This leads to a MAP estimator of the road network which is used in conjunction with a simulated annealing algorithm to get the final result.

The result depicted in Figure 17 needed about one hour of computation on a 1 GHz computer running GNU/Linux to be extracted. Since the method proposed in this work found the result shown in Figure 15 in less than thirty seconds on a similar computer, we can safely argue that our approach is faster.

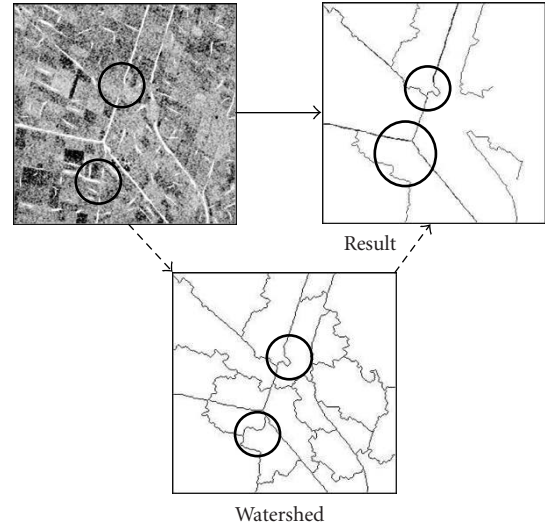


FIGURE 16: Roads partially occluded are not recovered.



FIGURE 17: Results from a Markov object process.

Another significant advantage of the proposed method is that we do not assume that roads can be approximated by connected straight lines; this leads to more natural results in rural areas, where roads are often curved.

However, relying on a Markov object process is a more flexible approach than ours since it is easier to model and introduce various kinds of information in such a process.

#### 4.3. Framework adaptation

The proposed recognition framework is not limited to road network extraction; it can also easily be adapted to segment some other curvilinear objects, as depicted in Figure 18. We briefly show in this section the steps that need to be modified for a given segmentation application.

The preprocessing step depends on the particular application and on the original image data. For instance, when the original image contains a curve to be segmented and when this curve is dark pixels on white background, the potential image can be as simple as the original image once inverted.

Setting the area parameter of the morphological filtering step also depends on both application and data. As explained in Section 2.1, this parameter removes image local minima.

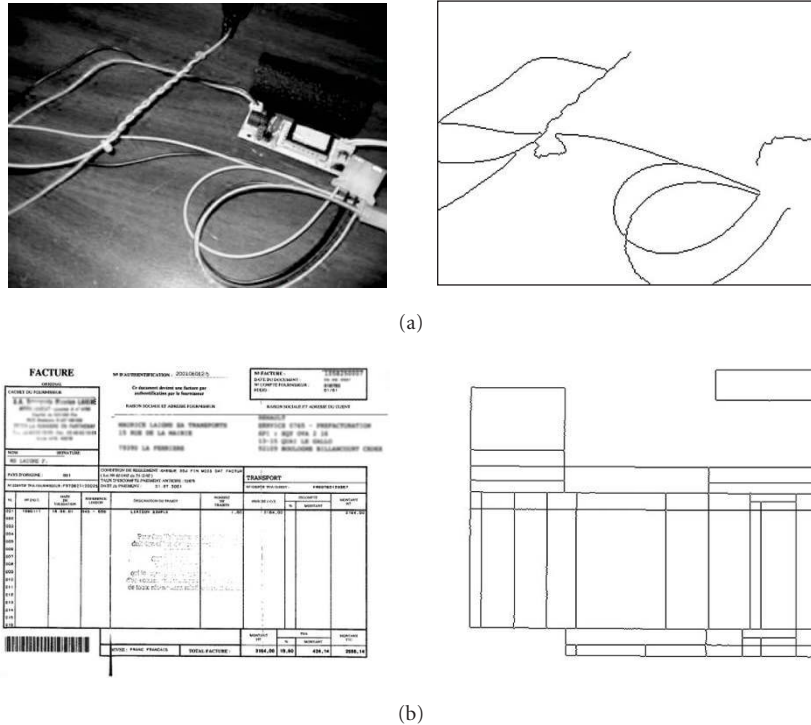


FIGURE 18: Examples of other applications of the proposed framework: (a) wires segmentation, (b) form segmentation.

Thus, considering the WT result, this parameter has an influence on merging small catchment basins. When a curvilinear object contains a loop, this loop can disappear if its area is lower than the area parameter value.

Finally, defining the energies for the MRFs is also data dependent. Features associated with nodes—a priori knowledge about piece of curvilinear objects—are numerous; the ones related to prior energy express knowledge about the global shape of the curvilinear objects and the connections between its different parts; for instance, a feature can be a continuity measure when the object is a smooth curve or, in the contrary, a measure that ensures that the object is only composed of straight lines and  $\pi/2$  breaks.

## 5. CONCLUSION

We have presented a method to extract road networks from satellite images. We have transposed the recognition scheme “watershed transform + region adjacency graph + Markov random fields,” described in Section 2.2 and which is dedicated to image segmentation, to the problem of road network recognition. To that aim, we propose a recognition scheme that is, as far as we know, original: “area opening + watershed transform + curve adjacency graph + Markov random fields.”

This recognition scheme is a global optimization process so it provides robust and reproducible results. Moreover, it is general and can easily be adapted to other image processing fields where the recognition of curvilinear structures is involved.

## ACKNOWLEDGMENT

The authors would like to thank Caroline Lacoste and Florence Tupin for both their help and support, and the reviewers for their relevant comments and suggestions.

## REFERENCES

- [1] D. Geman and B. Jedynak, “An active testing model for tracking roads in satellite images,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 1, pp. 1–14, 1996.
- [2] S. C. Zhu and A. Yuille, “Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 884–900, 1996.
- [3] F. Tupin, H. Maitre, J.-F. Mangin, J.-M. Nicolas, and E. Pechersky, “Detection of linear features in SAR images: application to road network extraction,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, no. 2, pp. 434–453, 1998.
- [4] M. Lindenbaum and A. Berengolts, “A probabilistic interpretation of the saliency network,” in *Proc. 6th European Conference on Computer Vision (2)*, vol. 1843 of *Lecture Notes in Computer Science*, pp. 257–272, Dublin, Ireland, June–July 2000.
- [5] R. Stoica, X. Descombes, and J. Zerubia, “A Markov point process for road extraction in remote sensed images,” Tech. Rep. 3923, INRIA, Sophia Antipolis, France, April 2000.
- [6] T. Géraud, “Fast road network extraction in satellite images using mathematical morphology and Markov random fields,” in *Proc. IEEE EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP '03)*, Trieste, Italy, June 2003.
- [7] L. Vincent, “Morphological area openings and closings for grey-scale images,” in *Proc. of the Workshop Shape in Picture:*

- Mathematical Description of Shape in Gray-Level Images*, Y.-L. O, A. Toet, D. Foster, H. J. A. M. Heijmans, and P. Meer, Eds., vol. 126 of *NATO ASI Series. Series F. Computer And System Sciences*, pp. 197–208, Driebergen, The Netherlands, September 1992.
- [8] A. Meijster and M. H. F. Wilkinson, “A comparison of algorithms for connected set openings and closings,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 484–494, 2002.
- [9] T. Géraud, J.-F. Mangin, I. Bloch, and H. Maître, “Segmenting internal structures in 3D MR images of the brain by Markovian relaxation on a watershed based adjacency graph,” in *Proc. of IEEE International Conference on Image Processing*, vol. 3, pp. 548–551, Washington, DC, USA, October 1995.
- [10] I. Y. Kim and H. S. Yang, “A systematic way for region-based image segmentation based on Markov random field model,” *Pattern Recognition Letters*, vol. 15, no. 10, pp. 969–976, 1994.
- [11] K. Haris, S. N. Efstratiadis, N. Maglaveras, and A. K. Katsaggelos, “Hybrid image segmentation using watersheds and fast region merging,” *IEEE Trans. Image Processing*, vol. 7, no. 12, pp. 1684–1699, 1998.
- [12] A. Bleau and L. J. Leon, “Watershed-based segmentation and region merging,” *Computer Vision and Image Understanding*, vol. 77, no. 3, pp. 317–370, 2000.
- [13] A. Sarkar, M. K. Biswas, and K. M. S. Sharma, “A simple unsupervised MRF model based image segmentation approach,” *IEEE Trans. Image Processing*, vol. 9, no. 5, pp. 801–812, 2000.
- [14] J. M. Gauch, “Image segmentation and analysis via multiscale gradient watershed hierarchies,” *IEEE Trans. Image Processing*, vol. 8, no. 1, pp. 69–79, 1999.
- [15] M. A. Fischler, J. M. Tenenbaum, and H. C. Wolf, “Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique,” *Computer Graphics and Image Processing*, vol. 15, no. 3, pp. 201–223, 1981.
- [16] A. Katartzis, H. Sahli, V. Pizurica, and J. Cornelis, “A model-based approach to the automatic extraction of linear features from airborne images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 9, pp. 2073–2079, 2001.
- [17] C. Steger, “Extracting curvilinear structures: a differential geometric approach,” in *Proc. 4th European Conference on Computer Vision*, vol. 1064 of *Lecture Notes in Computer Science*, pp. 630–641, Cambridge, UK, April 1996.
- [18] C. Steger, “An unbiased detector of curvilinear structures,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 2, pp. 113–125, 1998.
- [19] T. Géraud, *Segmentation des structures internes du cerveau en imagerie par résonance magnétique tridimensionnelle*, Ph.D. thesis, École Nationale Supérieure des Télécommunications, Paris, France, June 1998.
- [20] C. Steger, *Unbiased extraction of curvilinear structures from 2D and 3D images*, Ph.D. thesis, Fakultät für Informatik, Technische Universität München, München, Germany, 1998.
- [21] S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, 1984.
- [22] L. Najman and M. Schmitt, “Geodesic saliency of watershed contours and hierarchical segmentation,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 12, pp. 1163–1173, 1996.
- [23] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [24] C. Lacoste, X. Descombes, and J. Zerubia, “Road network extraction in remote sensing by a Markov object process,” in *Proc. of IEEE International Conference on Image Processing*, vol. 3, pp. 1017–1020, Barcelona, Spain, September 2003.
- [25] I. Laptev, H. Mayer, T. Lindeberg, W. Eckstein, C. Steger, and A. Baumgartner, “Automatic extraction of roads from aerial images based on scale space and snakes,” *Machine Vision and Applications*, vol. 12, no. 1, pp. 23–31, 2000.
- [26] J. Darbon, T. Géraud, and A. Duret-Lutz, “Generic implementation of morphological image operators,” in *Proc. 6th International Symposium on Mathematical Morphology (ISMM '02)*, pp. 175–184, Sydney, Australia, April 2002.

**Thierry Géraud** received an Engineering degree in telecommunications from École Nationale Supérieure des Télécommunications (ENST), Paris, France, in 1993 and the Ph.D. degree in signal and image processing from ENST in 1997. He is currently working at EPITA Research and Development Laboratory, Paris, France. His research interests include image processing, pattern recognition, software engineering, and object-oriented scientific computing.



**Jean-Baptiste Mouret** received an Engineering degree in computer science from École pour l'Informatique et les Techniques Avancées (EPITA), Paris, France, in 2004. He worked for one year at the EPITA Research and Development Laboratory (LRDE) and he is currently a student at the AnimatLab, a research team of the Laboratoire d'Informatique de Paris 6 (LIP6), Paris, France. This team is devoted to the animat approach, that is, to the synthesis of simulated animals or real robots, whose inner workings are as much inspired from biology as possible.

