Morse Description and Geometric Encoding of Digital Elevation Maps

Andrés Solé, Vicent Caselles, Guillermo Sapiro, and Francisco Arándiga

Abstract

Two complementary geometric structures for the topographic representation of an image are developed in this work. The first one computes a description of the Morse-topological structure of the image, while the second one computes a simplified version of its drainage structure. The topographic significance of the Morse and drainage structures of Digital Elevation Maps (DEM) suggests that they can been used as the basis of an efficient encoding scheme. As an application we combine this geometric representation with an interpolation algorithm and lossless data compression schemes to develop a compression scheme for DEM. This algorithm achieves high compression while controlling the maximum error in the decoded elevation map, a property that is necessary for the majority of applications dealing with DEM. We present the underlying theory and compression results for standard DEM data.

Keywords

Image compression, digital elevation maps, Morse structure, drainage structure, geometric encoding, geometric sampling.

I. Introduction

A geometric approach to encoding and compressing Digital Elevation Maps (DEM) is proposed in this paper. This is based on Morse theory and drainage structures, which lead to an efficient representation of the topographic structures of these images.

DEM data consist of a discrete digital representation of a surface terrain. Each cell in a DEM corresponds to a point (x, y, z) in 3D space. We can think of (x, y) as the

A. Solé and V. Caselles are with the Dept. de Tecnologia, Universitat Pompeu-Fabra, Passeig de Circumvalació, 8, 08003 Barcelona, Spain. E-mail: {andreu.sole,vicent.caselles}@tecn.upf.es .

G. Sapiro is with the Department of Electrical and Computer Engineering and the Digital Technology Center, University of Minnesota, Minneapolis, MN 55455, USA. E-mail: guille@ece.umn.edu .

F. Arándiga is with the Dept. Matemática Aplicada, Universitat de Valencia, C/ Dr. Moliner, s/n, 46100 Burjassot (València), Spain. E-mail: arandiga@uv.es .

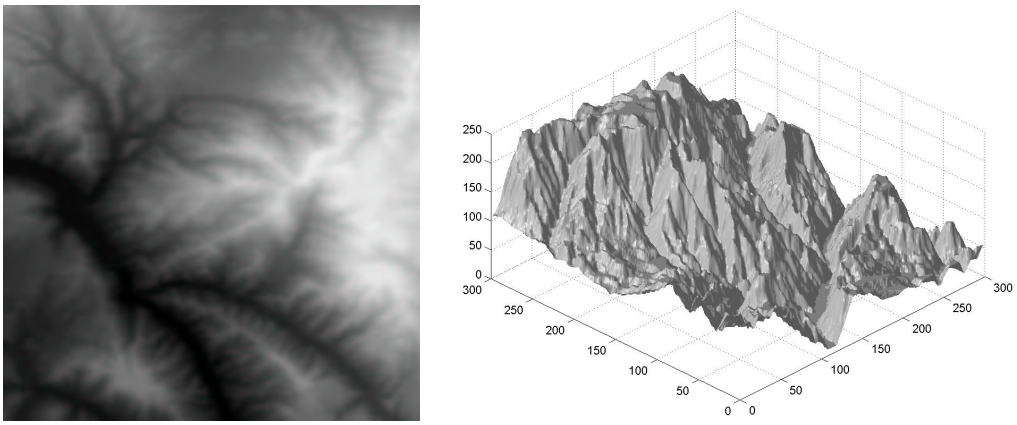

Fig. 1. Left: Original DEM images. Right: 3D representation of the image on the left.

coordinates in the image domain and the height z as the gray value of the image (see Fig. 1). The acquisition systems used to obtain a DEM have been improved during the last years in order to obtain a better resolution both in the coordinate plane and in height.

Obviously, this kind of data requires a large amount of bytes to store it. Typically a DEM image from a small terrain has 1200×1200 points, that is 1440000 bytes ($1.4Mb$) when using 8 bits for the height, or 2880000 ($2.8Mb$) when using 16 bits. If we note that for a complete terrain description of a country we need thousands of these images, storing and transmitting them requires efficient encoding and compression.

Many algorithms exist for data compression. They can be classified into lossless and lossy [1], [2], [3]. Lossless algorithms introduce no error, thereby limiting the amount of achieved data compression. Lossy algorithms achieve higher compression ratios at the expenses of introducing errors in the decoded image. It is important in this case to have a control on this error. Typically, lossy compression algorithms control the L_2 norm of the error (the root mean square error), but it is not so easy to find algorithms which allow a control on the L_∞ norm of the error (that is the sup error). This is fundamental for DEM applications. Without an L^∞ error control the error in individual pixels may be of the same order of magnitude as the image gray value resolution. For DEM applications, e.g., navigation and landing, this leads to an error in terrain height that makes the algorithm forbidden. A standard algorithm allowing the desired L_∞ control is JPEG Lossless (JPEG-LS) [2], [3], [4], which has a near lossless mode where one can impose the maximum allowed

error. The use of JPEG-LS for DEM data has been studied in [4]. Let us mention that when using standards such as JPEG-2000 [5], and then encoding the errors greater than a given threshold, a bound in the maximal error can be achieved as well. Of course, this is not part of the standard itself.

As mentioned above, the control of the maximum error per pixel (terrain point) is required when compressing DEM data to achieve an accurate description of terrain features. On the other hand, in the case of DEM data, it seems reasonable to store only those geometric structures which are of special relevance (such as its Morse and drainage structures, see below), and interpolate, while controlling the maximal error, the rest from this non uniform geometric sampling. The goal of this paper is to efficiently compute these structures and to use them to efficiently compress DEM data, while controlling the L_∞ norm of the error. The geometric description of the data is based on Morse theory and drainage structures, which provide, as we will show below, an efficient topographic representation of the image. This natural geometric representation of DEMs is not obtained with algorithms such as JPEG-LS or JPEG-2000.

This paper is structured as follows. Section II describes the current literature in topographic representation of images and will help to further motivate our geometric approach for DEM encoding. Section III is devoted to the computation of the Morse structure of an image. Some basic notions, like the notion of *monotone section* of a topographic map, are introduced, and an algorithm to compute the Morse structure is proposed. We also briefly comment on the mathematical justification of the proposed algorithms. In Section IV we describe an algorithm to compute a simplified drainage structure. Section V describes the coding and interpolation strategy and we summarize the compression algorithm used in our experiments. Section VI shows comparative results between the proposed algorithm and the standards JPEG-LS and JPEG-2000. Finally, in Section VII we present the main conclusions of this work.

II. Background on the topographic description of images

The use of a topographic description of images, surfaces, or $3D$ data has been introduced and motivated in different areas of research, including image processing, computer graphics, and geographic information systems (GIS), e.g., [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22]. The motivations for such a description differ depending on the field of application. In all cases, the aim is to achieve an efficient description of the basic shapes in the given image and their topological changes as a function of a physical quantity that depends on the type of data (height in data elevation models, intensity in images, etc.). In our brief literature review below, which will help to motivate our current contribution, we have separated the works into two main areas of research: computer graphics and image processing. In some cases this separation is somewhat arbitrary, some papers, if not all, could be included in both areas, since the application could be oriented to one or the other.

In computer graphics and geographic information systems, topographic maps represent a high level description of the data. Topographic maps are represented by the contour maps, i.e., the isocontours of the given scalar data. The description of the varying isocontours requires the introduction of data structures, like the *topographic change tree* or *contour tree* which can represent the nesting of contour lines on a contour map (or a continuous topographic structure) [23], [15], [16]. In all cases, the proposed description can be considered as an implementation of Morse theory, in the sense that Morse theory describes the topological change of the isocontours of scalar data or height function as the height varies, and relates these topological changes to the criticalities of the function. Given the scalar data u defined in a domain Ω of $\mathbb{R}^N$ ($u : \Omega \to \mathbb{R}$), the contour map is defined in the literature as the family of isocontours $[u = \lambda] = \{x \in \Omega : u(x) = \lambda\}$, $\lambda \in \mathbb{R}$, or in terms of the boundaries of upper (or lower) level sets $[u \geq \lambda] = \{x \in \Omega : u(x) \geq \lambda\}$ ($[u \leq \lambda]$). The first description is more adapted to the case of smooth data while the second description can be adapted to more general continuous data where there are plateaus of constant elevation or discontinuous data. The second description has been addressed in [12], [15], while the first description has been used in [9], [10], [16], where an apriori

interpolation of the discrete data is required so that the regularity assumptions permit the isocontour description.

The contour map is organized in a data structure, either the contour tree [15], [16], or the Reeb graph [24], [25]. The contour tree represents the nesting of contour lines of the contour map. According to [15], each node represents a connected component of an upper (or lower) level set $[u \geq \lambda]$ ($[u \leq \lambda]$), and links between nodes represent a parent-child relationship, a link going from the containing to the contained set in the upper tree, or viceversa if we consider the lower tree. Each node has a list of descendants, its corresponding elevation value, a list of boundary points, and its parent. The contour tree encodes the topological changes of the level curves of the data. Critical values and its associated features, peaks (maxima), pits (minima), or passes (saddles), can be extracted from the contour trees [15]. The description of the topographic changes requires the use of both upper and lower trees, and the contour tree can also be used as a tool to compute other terrain features such as ridges and ravines [15]. For practical applications, the data structure has to be implemented with a fast algorithm and with minimal storage requirements. In [16] this is accomplished with a variant of the contour tree where the criticalities (maxima, minima, saddles, computed in a local way) are computed first. In [9] several attributes have been added to the contour data which can be used to select a subsampled family of contours which are representative of the data. As examples of such attributes the authors choose the length or area of the isocontours, the ratio length of the isocontour/area of the enclosed set, or the integral of the gradient along the isocontour.

A related data structure is the *Reeb graph*, which represents the splitting and merging of the isocontours. The *Reeb graph* of the height function u is obtained by identifying two points $p, q \in \Omega$ such that $u(p) = u(q)$ if they are in the same connected component of the isocontour $[u = u(p)]$. Thus, a cross-sectional contour corresponds to a point of an edge of the Reeb graph, and a vertex represents a critical point of the height function u. The Reeb graph was proposed in [24] as a data structure for encoding topographic maps. The authors proposed to compute it following the computation of the so-called surface network. The surface network is also a topological graph, i.e., a graph that represents the

relations among critical points, whose vertex are critical points and the edges represent either a ridge or a ravine line. A ridge (ravine) line is a line with steepest gradient which joins a pass to a peak (pit) [24], [26]. The critical points are computed with an algorithm based on local computations, so that the Euler's formula relating the number of peaks, pits and passes, holds. Then ridges and ravines are computed following the steepest lines. In the context of computer graphics, Morse theory has also been used to encode surfaces in $3D$ space [22]. In [22], the authors also use a tree structure like the Reeb graph complemented with information about the Morse indexes of the singularities and including enough (information about) intermediate contours to be able to reconstruct by interpolation the precise way in which the surface is embedded in $3D$ space.

In image processing, the topographic description was advocated as a local and contrast invariant description of images (i.e., invariant under illumination changes), and has lead to an underlying notion of shapes of an image as the family of connected components of upper or lower level sets of the image [11], [20], [21], [27]. An efficient description of the family of shapes in terms of a tree was proposed in [18], [19] and further developed in [17]. The tree of shapes as proposed in [18], [19] fuses the information of both the trees of upper $[u \geq \lambda]$ and lower $[u \leq \lambda]$ level sets of the scalar image u. The key idea for this fusion is the notion of shape as a connected component of an upper $[u \geq \lambda]$ or lower $[u < \lambda]$ level set in which the holes are filled-in. This topographic structure has been further studied in [6], [7], [19], where a Morse description of this topographic structure was developed. The mathematical description permitted to include the case of images as upper semicontinuous functions. In [17], following bilinear interpolation of the discrete data, the image could be treated as a continuous function and a tree of bilinear level lines $[u = \lambda]$ was computed. The tree of bilinear level lines is more related to the contour tree computed with the isocontours of the interpolated image. A subtree containing the so-called meaningful level lines [28] can be extracted which contains the main level lines according to the distribution of gradient values of the image in a statistical way [17], [28]. The work in [29] can be considered as a mathematical description of the (iso) contour tree in the case of two-dimensional functions. In [12], Morse theory has also been used as a

basic model to describe the geometric structures of $2D$ and $3D$ images, and in general, of multidimensional data. Applications have been given in different domains, in particular, to visualize structures in $3D$ medical images. The data are typically multi-dimensional sampled data, and it cannot be assumed that the function is Morse in a traditional sense, even if interpolated. Thus, the authors adapt Morse theory using combinatorial methods. The authors assume that the given data are interpolated by a continuous real valued functions u. The basic geometric objects studied are the boundaries of the connected components of upper level sets $[u \geq \lambda]$ of u and their variation with the level λ. In their set of axioms, the authors assume that those boundaries are compact, oriented manifolds in $\mathbb{R}^N$, they precise their local structure, and its connection with the original sampled data. In particular, those axioms imply that the topological structure of the sampled data is reflected by any interpolating function satisfying their axioms. Then critical points and critical values are defined, obtaining maximum, minimum, and saddle critical points (and values). Criticalities are defined by local analysis of the function, including the case of degenerate sets (i.e., connected regions of the sampled data with the same values). The authors prove that the topology of their basic objects change at a critical level, and does not change between critical levels. Then the criticality graph is defined, the vertices of the graph are the criticalities and the edges go between criticalities in such a way that no further criticalities are located between them. The regions represented by each edge are called zones of the critical point with higher value. In each zone the boundaries of the upper level sets are homeomorphic [12]. The topology change at a critical value is computed by combinatorial methods if the critical value is a saddle or a minimum, and by the genus in the case of a maximum value. In some sense, this structure is related to a Reeb graph. Efficient algorithms are proposed which compute the criticality graph [12].

Let us finally mention that a morphological approach to image compression has been proposed by several authors, for instance [30], [31], [32], [33], [34]. In [30],[32] the authors propose to use binary partition trees to select the level curves which have to be encoded. The trees take into account the cost in bits to encode the selected level boundaries and the approximation error (measured with an L^2 norm). In [34], the author selected the

level lines taking into account its perceptual significance which was measured in terms of the number of T and X junctions contained in it. Because of our application to the encoding of DEM data, we use a description more adapted to the topographic features of the data.

III. Morse theory and monotone sections

As we explained in the previous section, the aim of Morse theory is to describe the topological changes of the (iso)level sets of a function in terms of its critical points. We consider here the case of 2D real valued data (gray level images, or height data), and describe a simple and efficient algorithm to compute its Morse structure from its upper and lower level sets.

To make the ideas precise, let us first recall some definitions. Let D be a subset of $\mathbb{R}^N$. Given a function $u : D \to \mathbb{R}$, we call upper (lower) level set of u any set of the form $[u \geq \lambda] := \{x \in D : u(x) \geq \lambda\}$ or $[u > \lambda] := \{x \in D : u(x) > \lambda\}$ ($[u \leq \lambda] := \{x \in D : u(x) \leq \lambda\}$ or $[u < \lambda] := \{x \in D : u(x) < \lambda\}$), where $\lambda \in \mathbb{R}$. The (upper) topographic map of a function u is the family of the connected components of the level sets of u, $[u \geq \lambda]$, $\lambda \in \mathbb{R}$, the connected components being understood in the relative topology of D. We shall assume that our image is a continuous function and each upper or lower level set has a finite number of connected components which is true for a discrete image or for a continuous image after some basic filtering [6]. It was proved in [11] that the topographic map is the structure of the image which is invariant under local contrast changes, a notion also defined in [11]. In [6], [7] the authors studied the Morse structure of the topographic map for continuous functions (a similar study can be done for bounded upper semicontinuous functions). They defined a notion of nonsingular region of the topographic map trying to express the fact that the level lines of the topographic map in a nonsingular region do not change topology, as it happens for smooth functions where singularities are understood in the usual way [35]. A first version of this notion appeared in [27].

Let $u : D \to I\!R$ be a function. For each $\lambda, \mu \in I\!R$, $\lambda \leq \mu$ we define

$$U_{\lambda,\mu} = \{x \in D : \lambda \leq u(x) \leq \mu\}.$$

The connected components of a set $X \subseteq I\!R^N$ will be denoted by $\mathcal{CC}(X)$. If $x \in X$, the connected component of X containing x will be denoted by $cc(X, x)$

Definition 1: Let $u : D \to I\!R$ be a continuous function. A monotone section of the topographic map of u is a set of the form

$$X_{\lambda,\mu} \in \mathcal{CC}(U_{\lambda,\mu}), \tag{1}$$

for some $\lambda, \mu \in I\!R$ with $\lambda \leq \mu$, such that for any $\lambda', \mu' \in [\lambda, \mu]$, $\lambda' \leq \mu'$ the set

$$\{x \in X_{\lambda,\mu} : \lambda' \leq u(x) \leq \mu'\}$$

is a connected component of $U_{\lambda',\mu'}$.

Due to small oscillations in the image, its Morse structure is too complex, i.e., there are many criticalities. To simplify the structure of the topographic map while preserving its main features we filter the image with the Vicent-Serra filters [36],[37], also called extrema filters [38]. Extrema filters eliminate the small connected components of upper and lower level sets of the given image [6], [39]. The resulting image has a simplified topographic map structure. If u has been filtered with the extrema filters with an area threshold ϵ and X is a connected component of $[u \geq \lambda]$ or $[u < \lambda]$, then $|X| > \epsilon$ [6],[39]. This implies that, for any $\lambda \leq \mu$, there are a finite number of connected components of $U_{\lambda,\mu}$. In particular, there are a finite number of connected components of $[u \geq \lambda]$ and $[u < \lambda]$ [6], [39]. In all what follows we shall assume that

(**H**) u has been filtered with the extrema filters.

Assuming property (**H**), if $X_{\lambda,\mu}$ is a monotone section, then the family of sets $X \cap [u = \alpha]$, $\alpha \in [\lambda, \mu]$, is a family of nested connected sets [40], [41]. Moreover, the union of monotone sections which intersect is a monotone section [6], Proposition 2, [40]. This permits to define the notion of maximal monotone section containing a given point. Let $x \in D$ and $\lambda = u(x)$. For each $\eta \geq \lambda$, let $X_{\lambda,\eta} = cc(U_{\lambda,\eta}, x)$. We define

$$\eta_+(x, \lambda) = \sup\{\eta : \eta \geq \lambda \quad \text{s. t. } X_{\lambda,\eta} \text{ is a monotone section}\}.$$

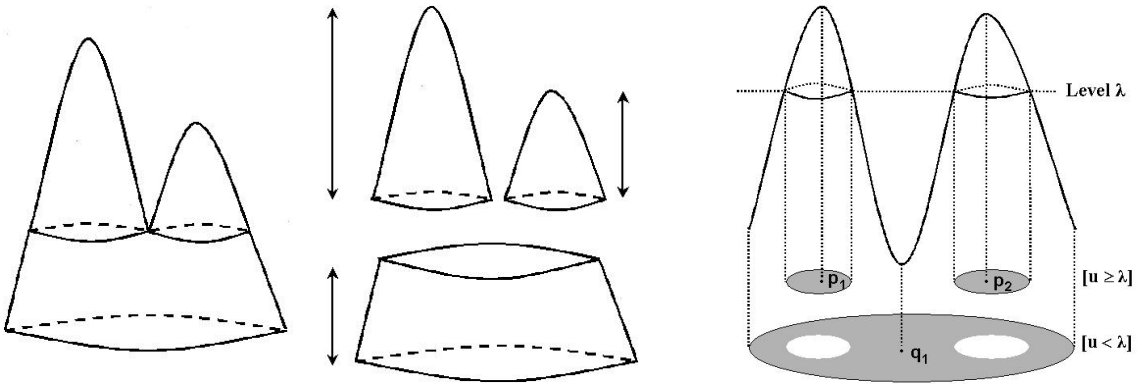

Fig. 2. Left: Original function and its decomposition in its maximal monotone components. The vertical arrows denote the degree of contrast of each monotone section. Right: A function u and its upper and lower level sets at level λ with its assigned signature.

Similarly, we define

$$\eta_-(x,\lambda) = \inf\{\eta : \eta \leq \lambda \quad \text{s. t. } X_{\eta,\lambda} \text{ is a monotone section}\}.$$

Note that both numbers are well defined since $X_{\lambda,\lambda}$ is always a monotone section. Note that, by definition, $\eta_-(x,\lambda) \leq \eta_+(x,\lambda)$. By Proposition 2 in [6] (see [40]), we may define the (open, closed, half-open, half-closed) interval $I(x,\lambda)$ containing λ whose end-points are $\eta_-(x,\lambda), \eta_+(x,\lambda)$ and which determines a monotone section containing x maximal with respect to inclusion, which we denote by $X_{I(x,\lambda)}$. Note that $\lambda \in I(x,\lambda)$ for all $\lambda \in (-\infty, \sup_D u(x)]$.

Maximal monotone sections represent the largest sections of the topographic map containing no topological changes (see Fig. 2, left), and our purpose is to compute them. Intuitively monotone sections are topologically equivalent to truncated cones and the maximal monotone ones are the largest truncated cones contained in the graph of the image. The interpolation algorithms described in Section V are able to re-interpolate these truncated cones from the curves bounding them.

To compute the maximal monotone sections we require some additional definitions.

Definition 2: Let $M \subseteq \overline{\Omega}$. We say that M is a zonal maximum (resp., minimum) of u at height λ if M is a connected component of $[u = \lambda]$ and, for all $\epsilon > 0$, the set $[\lambda - \epsilon < u \leq \lambda]$ (resp., $[\lambda \leq u < \lambda + \epsilon]$) is a neighborhood of M.

Definition 3: We say that $\lambda \in I\!R$ is a singular value of the topographic map of u if it

corresponds to a zonal maximum, minimum, or it corresponds to a level where it begins or ends a maximal monotone section, i.e., there is a point $x \in \overline{\Omega}$ such that $\eta_+(x,\lambda) = \lambda$ or $\eta_-(x,\lambda) = \lambda$.

We will present a simple algorithm to compute the singular values of the topographic map of u. For this we introduce a notion of critical value of u and we observe that it is indeed equivalent to the notion of singular value (detailed proofs are given in [40], [41]).

Definition 4: Let $\lambda \in I\!R$. Assume that (**H**) holds. The signature of the level set $[u \geq \lambda]$ (resp. $[u < \lambda]$) consists of a finite family of points $\{p_i : i = 1, ..., r\}$ (resp. $\{q_j : j = 1, ..., s\}$) such that each p_i (resp. q_j) is a point in a different connected component $X^{\lambda,i}$ (resp. $X_{\lambda,j}$) of $[u \geq \lambda]$ (resp. $[u < \lambda]$). The points p_i, q_j are selected so that

$$u(p_i) = \sup_{x \in X^{\lambda,i}} u(x), \text{and} \quad u(q_j) = \inf_{x \in X_{\lambda,j}} u(x).$$

We denote the signature of $[u \geq \lambda]$ by $sig([u \geq \lambda])$, the signature of $[u < \lambda]$ by $sig([u < \lambda])$. We define $sig([u \geq \lambda], [u < \lambda]) = sig([u \geq \lambda]) \cup sig([u < \lambda])$.

An example of the above definition is given in Fig. 2 (right). To stress the marker point in the notation, we write X^{λ,p_i} instead of $X^{\lambda,i}$. As it is proved in [40], [41], under assumption (**H**), given $\lambda \in I\!R$, there is $\epsilon > 0$ such that $sig([u \geq \mu], [u < \mu])$ is constant for all $\mu \in (\lambda - \epsilon, \lambda]$.

Definition 5: We say that $\lambda \in I\!R$ is a critical value for u if there is a sequence $\mu_n \downarrow \lambda$ such that $sig([u \geq \mu_n], [u < \mu_n]) \neq sig([u \geq \lambda], [u < \lambda])$ for each $n = 1, 2, ...$

The following result is proved in [40], [41].

Proposition 1: Let $\lambda \in I\!R$. Assume that (**H**) holds. Then λ is a critical value of u if and only if λ is a singular value of the topographic map of u.

This permits us to compute the singular values of the topographic map of u by computing the critical values of u. Moreover, at the discrete level, a topological change occurs when going from level λ to level $\lambda - 1$ if and only if either *i)* $sig([u \geq \lambda]) \neq sig([u \geq \lambda - 1])$, or *ii)* $sig([u < \lambda]) \neq sig([u < \lambda - 1])$. This is equivalent to saying that *i)* two connected components of $[u \geq \lambda]$ merged at level $\lambda - 1$, a phenomenon which we call merging of upper connected components, or a new connected component of the upper level sets was

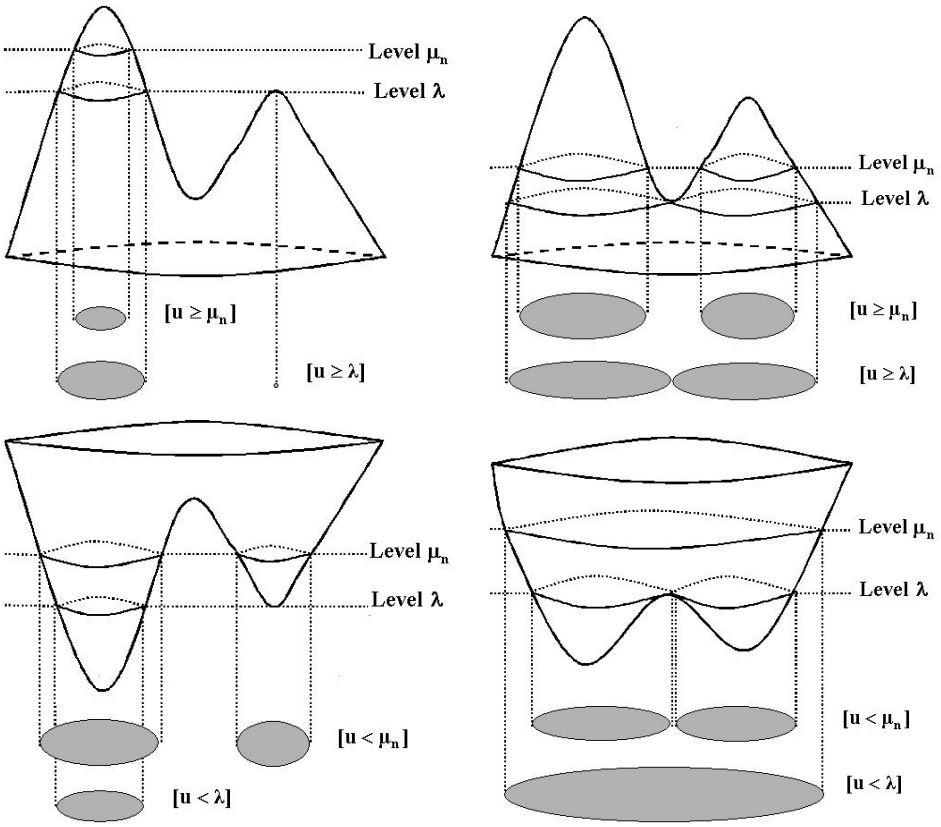

Fig. 3. From left to right and up to down the four cases explained in the text.

born at level $\lambda - 1$, or *ii*) two connected components of $[u < \lambda - 1]$ merged at level λ, a phenomenon which we call merging of lower connected components, or a new connected component of the lower level sets was born at level λ (see Fig. 3). We call the first type of criticality of upper type, while the second is called of lower type.

Several notions of criticalities have appeared in the literature (see previous section). Our aim has been to show that, essentially, they are equivalent for $2D$ images. Full proves will be given in subsequent papers.

A. The computational algorithm

For the sake of simplicity we only describe the algorithm to compute the critical values of upper type. The critical values of lower type can be computed using the same algorithm applied to the inverted image $\max(u) - u$. Note that it is possible to compute both types of criticalities at the same time. Anyway, the computational cost will be similar either if we compute the sup and inf monotone sections separately or at the same time. As we

will see in Algorithm 1, the computational cost derives mainly from the computation of the connected components at each level. Since we perform this computation by means of a region growing strategy, it makes no difference to do both computations (upper and lower connected components) simultaneously or not.

As before, let D be a subset of $\mathbb{R}^N$ and $u : D \to \mathbb{R}$ a continuous function. To simplify our terminology we call sup-monotone section a connected component X of an upper level set $[u \geq \lambda]$ which contains no criticalities of upper type due to merging of connected components of upper level sets. Note that it contains a single critical value due to the birth of the connected component corresponding to a zonal maximum. As already mentioned, for simplicity we only describe how to compute the sup-monotone sections. The corresponding inf-monotone sections can be computed with the same algorithm applied to $max(u) - u$.

We have observed that due to low oscillations in the image, a large number of criticalities may appear. In order to avoid this problem we have used two different strategies. The first one consists on making a pre-filtering step to reduce the number of low oscillations present in the image.[1] This pre-filtering step can be done by means of the extrema filters which preserves as much as possible the topographic map structure [6], [39]. The second strategy (which can be combined or not with the pre-filtering step) consists on discarding those critical sections which do not satisfy a minimum contrast criterion, in other words, low contrasted sup-monotone sections will be discarded. The contrast criteria is specified by means of a parameter $MinContrast$.

We need some additional simple notation in order to make explicit an algorithm taking into account the contrast criterion. Denote by $(X^{\lambda,p_i}, \beta^{\lambda,p_i})$ the sup-monotone section of $[u \geq \lambda]$ beginning in β^{λ,p_i}, thus it contains no criticality of upper type due to merging. We can define now a measure of contrast C for the couple $(X^{\lambda,p_i}, \beta^{\lambda,p_i})$ simply by $C(X^{\lambda,p_i}, \beta^{\lambda,p_i}) = \beta^{\lambda,p_i} - \lambda$. Note that at a critical level μ the couple $(X^{\mu,p_i}, \beta^{\mu,p_i})$ defines a maximal sup-monotone section of the topographic map starting at level β^{μ,p_i} and ending

[1] Note that this filtering is done only for the computation of the Morse structure. All other operations, and in particular the maximal encoding error, refer to the original image.

Algorithm 1

1 *Set* $\lambda = M$ *and compute* X^{λ,p_i} *and set* $\beta^{\lambda,p_i} = M$, $\forall i$.

2 *Store the couples* $(X^{\lambda,p_i}, \beta^{\lambda,p_i})$ *in* $L[\lambda]$.

3 *If* $(\lambda - 1) > m$, *set* $\lambda = \lambda - 1$, *and recompute* X^{λ,p_i} *(note the abuse of notation on* λ *and* p_i*); else go to step* ***8****.*

4 $\forall p_i \in sig([u \geq \lambda])$ *if* $p_i \in sig([u \geq \lambda + 1])$ *and* X^{λ,p_i} *was not marked as critical, then* $\beta^{\lambda,p_i} = \beta^{\lambda+1,p_i}$*; else* $\beta^{\lambda,p_i} = \lambda$

5 *Let* $A = sig([u \geq \lambda + 1]) \backslash (sig([u \geq \lambda + 1]) \cap sig([u \geq \lambda]))$.

6 *If* $A \neq \phi$ *then* $\forall p \in A$ *search* $q \in sig([u \geq \lambda])$ *such that* $p \in X^{\lambda,q}$ *and mark* $X^{\lambda,q}$ *as critical.*

7 *Return to step* ***2****.*

8 *Select from* L *the list of couples* $(X^{\lambda,p}, \beta^{\lambda,p})$ *marked as critical and verifying that* $C(X^{\lambda,p}, \beta^{\lambda,p}) \geq MinContrast$.

9 *The output of the algorithm will be the set of curves* Γ_M *corresponding to the boundaries of the selected sections* $X^{\lambda,p}$ *and its markers.*

TABLE I

ALGORITHM FOR COMPUTING THE MAXIMAL SUP-MONOTONE SECTIONS OF AN IMAGE.

at level μ (see Fig. 2).

We assume that our image u ranges from $m = min(u)$ to $M = max(u)$ (for example from 0 to 255). To store the computed connected components we use a dynamical data structure L consisting on a vector, ranging from m to M, of lists of couples $(X^{\lambda,p_i}, \beta^{\lambda,p_i})$. The algorithm computing the maximal sup-monotone sections is summarized in Table I.

It is important to remark that the parameter $MinContrast$ allows us to discard low contrasted sup-monotone sections in a *causal* way. That is, if we denote as $MS(u, MinContrast)$ the set of maximal sup-monotone sections of u with contrast $\geq MinContrast$, then the inclusion principle $MS(u,0) \supset MS(u,1) \supset \ldots \supset MS(u,n)$ holds. An example of the performance of the preceding algorithm is presented in Fig. 4, where we have computed the critical sections of a low resolution synthetic image.

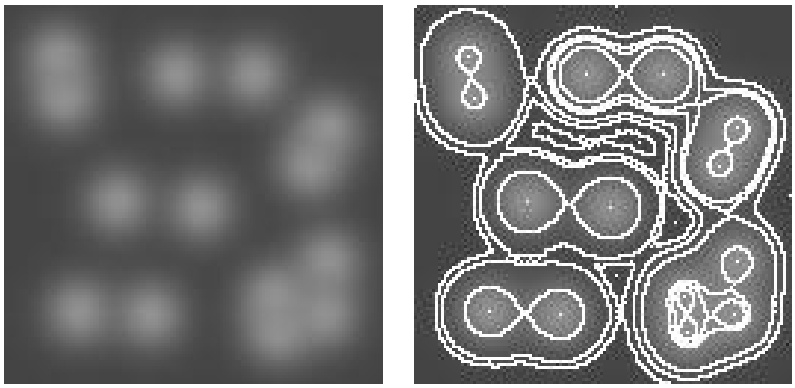

Fig. 4. Left: Synthetic image representing a set of peaks. Right: Computed saddle structures and criticalities.

As we noted in the Introduction, many different approaches exist to compute the basic Morse structure of images or 3D data. We have presented a simple computational approach and have studied its mathematical properties in [40], [41]. Longer writing space would be required to prove that all the approaches are essentially equivalent in theory. Let us also note that the singularities of the tree of shapes of an image as introduced in [19], [7], coincide with the notion of critical value defined here [40].

IV. Drainage structures

In the previous section we have developed an algorithm to compute the Morse structure of an image. This Morse structure consists on the maxima, minima, and the level lines where a topology change occurs (i.e., the boundaries of the maximal monotone sections). This can be considered as a reasonable *global* description of a DEM. There are also other structures which are of special interest due to its topographic significance in DEM data. These structures correspond mainly to the drainage structures (e.g., rivers and ravines). There exists many different algorithms accurately computing such structures, see [42] and references therein. We will present an approach which is related to the one in [43]. Strictly speaking, we do not compute the drainage structures but a version of them which is adapted to our purposes. In a simplistic way we can think of the drainage structures as the set of points for which there exists at least one direction in which the flow of water is accumulated or repealed. We can write down this definition mathematically by considering the set of points $\{\vec{x} : \exists \vec{v} \in I\!R^2$ and $\varepsilon \geq 0$ such that $u(\vec{x}) \leq u(\vec{x} + t\vec{v})$

$\forall t \in (-\varepsilon, \varepsilon)$, or $u(\vec{x}) \geq u(\vec{x} + t\vec{v})\ \forall t \in (-\varepsilon, \varepsilon)\}$. Observe that this set of points will contain in particular the maxima and minima of u. It has been shown in [40], [41] that this set of points is contained in an at most a countable family of curves. Intuitively, these curves contain the drainage structures (ridges and valleys), giving also information about boundaries of plateaus for example. In the discrete case we only consider 4 different directions (values of v) corresponding to 4 different profiles in the image. Concretely, we search maxima and minima of the vertical, horizontal and diagonal profiles. In [43] only two directions were used, namely the horizontal and vertical ones.

Algorithm 2

1 *Set* $(v1, p1) \leftarrow ComputeExtrema(x)$

2 *Set* $v^f = v1$

3 *for* $j = 1$ *to* $length(v1)$ *do*
  compute $c_{left} = |v(j) - v(j-1)|$
  compute $c_{right} = |v(j) - v(j+1)|$
  if $c_{left} < thr$ *or* $c_{right} < thr$ *then*
    set $c_{left} = |v^f(j) - v^f(j-1)|$
    if $c_{left} < thr$ *then*
      $v^f(j) = v^f(j-1)$
    else
      $v^f(j) = v^f(j+1)$
    end if
  end if
end for

4 *Set* $(v2, p2) \leftarrow ComputeExtrema(v^f)$

5 *Set* $p(i) = p1(p2(i))\ \forall i = 1..length(p2)$. *This vector* p *contains the* x *coordinates of the selected extrema. We can think of* p *as the vector defining the intervals of monotonicity of* x *after the filtering. Due to the flatness of* v^f, *in each interval* $(p(i-1), p(i+1))$ *the extrema* $v2(p(i))$ *could not be the absolute maximum (minimum) so we have to recompute it in each of these intervals.*

6 *Finally, if an extrema* $v2(i)$ *lies in a flat zone we redefine the extrema as the boundaries of this flat zone.*

TABLE II

ALGORITHM FOR COMPUTING THE SIGNIFICANT EXTREMA OF A $1D$ PROFILE.

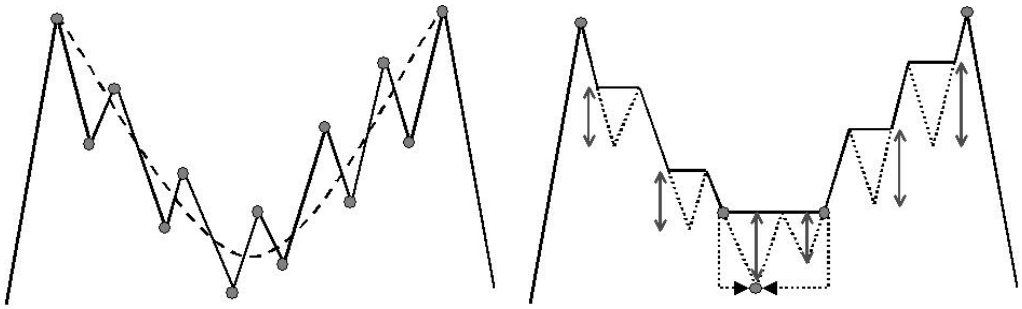

Fig. 5. Left: Computed maxima and minima of a 1D signal. The dotted curve denotes the ideal profile (before introducing electronic and quantization noise which is always present in digital images). Right: Signal obtained by filtering the less contrasted extrema according to **Algorithm 2** (light gray arrows depict the contrast of the filtered extrema). Note that if we simply reject those extrema instead of filtering them with **Algorithm 2**, then the only computed extrema would be the two maxima.

In order to specify the algorithm we assume that we have a simple subroutine, *ComputeExtrema*, which receives as input a vector $x = [x(1), ..., x(n)]$ and returns two vectors (v, p), where $v = [v(1), ..., v(k)]$ contains the maxima and minima of x consecutively and $p = [p(1), ..., p(k)]$ its relative positions in such a way that $v(i) = x(p(i))$. The problem is that a large number of extrema can appear due to low oscillations, mainly due to noise. In order to solve this problem we have to choose the most significant extrema, in our case those which have maximum *contrast*. The contrast of an extremum $v(i)$ can be defined simply by $C(v(i)) = min(|v(i) - v(i-1)|, |v(i) - v(i+1)|)$. Using this criteria directly, for example by eliminating those extrema with contrast less than a threshold thr, produces undesirable results (see Fig. 5). Instead of using the contrast to eliminate extrema points we use it to filter them. The filtering step simply consists on replacing the value of $v(i)$ by $v(i-1)$ if $C(v(i)) = |v(i) - v(i-1)| < thr$ or $v(i+1)$ if $C(v(i)) = |v(i) - v(i+1)| < thr$. These filtering step produces a new vector v^f and recomputing the extrema of v^f produces a more desirable result, see Fig. 5. If any computed extremum lie in a flat zone (i.e., it is a zonal extremum), we shall replace it by the boundary of the flat zone. The algorithm computing these significant extrema from a 1D-profile x is summarized in table II.

Figure 5 illustrates the extrema filtering process. On the left we show the real (solid line) and the ideal (dotted line) profile without noise and its extrema. On the right we show the profile after the filtering process. Note that the filtered profile contains a new

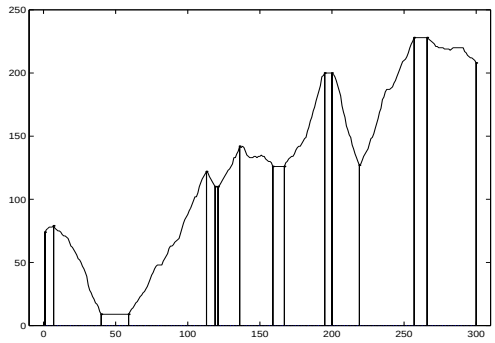

Fig. 6. Computed zonal maxima and minima of one image profile using the algorithm described above with a contrast parameter of 10.

minimum (zonal minimum in fact) which does not coincide with any of the minima of the original signal. This is solved easily by recomputing the absolute minimum in the interval where the filtered signal is convex (see step **5** of **Algorithm 2**).

Figure 6 shows the output of the algorithm just presented using a real profile of DEM data. In order to obtain the desired curves we must compute the maxima and minima of all the selected profiles of the image. As we have previously said, we compute these curves using the horizontal, vertical and diagonal profiles of the image.

Figure 7 illustrates the whole geometric sampling process. From left to right and top to bottom we show the original DEM image, the level lines Γ_M corresponding to its Morse structure ($MinContrast = 10$), the curves Γ_D corresponding to the extrema of the profiles ($thr = 10$), and the final sampling $\Gamma_{xy} = \Gamma_M \cup \Gamma_D$. A thinning step has been performed in order to obtain one pixel width curves.

V. The interpolation and coding step

We have described two algorithms that compute important points and curves from an image, thereby providing the basic geometric description of DEM data. These algorithms can be considered as a non uniform geometric sampling of the image. The next step is to interpolate the missing data from our sampling. There exists several algorithms to interpolate data from curves and/or points. We have in particular tested three of them: the Laplacian model (which corresponds to $min \int \|\nabla u\|^2 dxdy$), the AMLE model (which corresponds to $min \lim_{p\to\infty} \int \|\nabla u\|^p dxdy$), and the thin plate model (which corresponds

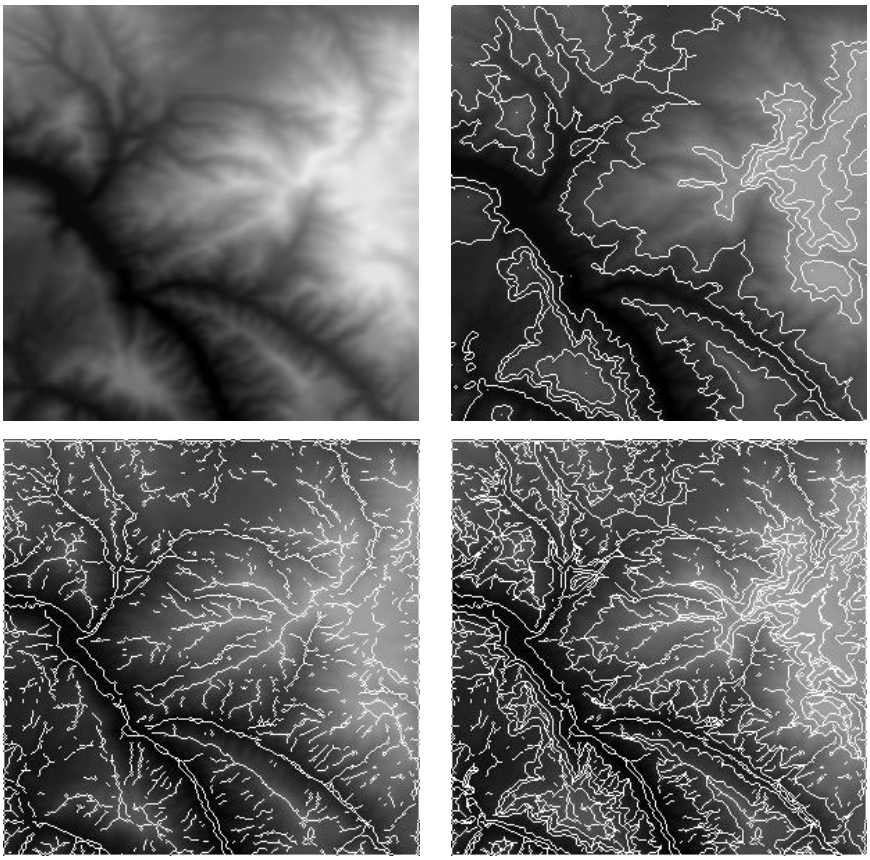

Fig. 7. From left to right and up to down: The original DEM image, computed level lines (white), computed ridge/valley structure (white) and the whole image sampling (white).

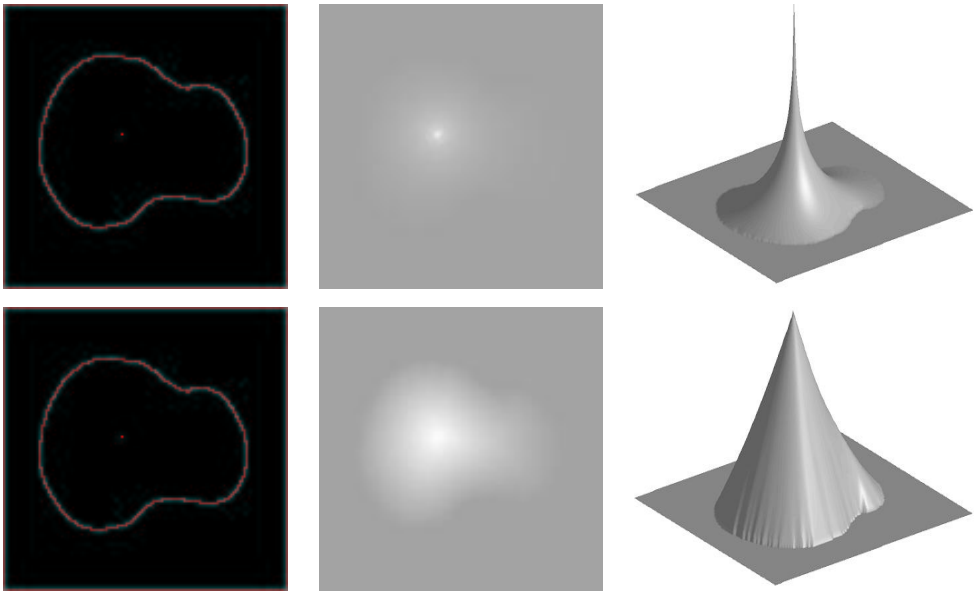

Fig. 8. Top row: From left to right the initial data, the interpolated data using Laplacian interpolation and its $3D$ representation. Bottom row: From left to right the initial data, the interpolated data using AMLE interpolation and its $3D$ representation.

to min $\int(\triangle(u))^2 dxdy)$. We remark that, rigorously speaking, only the AMLE model can be used to interpolate values specified on points [44] (see Fig. 8). In spite of this, we shall also use the Laplacian since there are many curves in the data and we may think of points as small regions. In order to evaluate these interpolation schemes we have chosen as a measure of goodness the entropy of the residual between the original image and the interpolated one. This is a natural choice since we want to minimize the number of bits used to encode the errors between the interpolated and the original images. After several tests we have discarded the thin plate approximation since it needs to store not only the gray values at the curves but also its derivatives in order to obtain a good interpolation. The AMLE model and the simple Laplacian model were both tested using the whole image sampling (Γ_{xy} and $u|_{\Gamma_{xy}}$) and using only the level lines (Γ_M and $u|_{\Gamma_M}$). In the second case the AMLE model performed better than the Laplacian (the entropy of the residual, the maximum, and $RMSE$ errors were lower than in the Laplacian interpolation). Surprisingly, in the case of the whole sampling structure the winner was the Laplacian, although the interpolation for the AMLE model looked visually better and the maximum errors were almost the same. After these tests we have decided to use the Laplacian interpolation to obtain the first estimation of the image from the selected curves and points. In order to control the maximum (sup) error we simply store/encode the quantized error information (that is why the entropy of the residual was a natural measure of goodness for the interpolant). See the Appendix for more details on these interpolation processes.

At this point we need to consider how to encode both the initial curves (geometrically sampled data) and the residuals once a sup error e is specified. We proceed to address this now. The geometry of the sampled curves (Γ_{xy}) and their gray levels ($u|_{\Gamma_{xy}}$) are encoded separately. To encode the geometry we use a differential chain coding strategy, see [45], [46], [47]. In the future we plan to explore an encoding based on rate-distortion theory, as in [48]. For the gray levels we use an ENO (Essentially Non Oscillatory) based encoding scheme [49] which also controls the sup error, a fundamental requirement of the application as stated before. Finally we compress both the geometry and the gray values

of the curves using an arithmetic coder. Having these curves and the data on them, we can interpolate them by means of the Laplace equation to obtain the first estimate of the image.

Finally, to control the maximum error, we need to store the residuals r. Encoding the residuals r can be simply done by quantizing them using

$$r^q = sign(r)\left\lfloor \frac{|r|}{e} \right\rfloor e, \tag{2}$$

and then coding the resulting r^q with an arithmetic coder.

Algorithm 3

1	*From the original image, eventually filtered to reduce its bandwidth, u compute a subsampled image* $u_l(i,j) = u(li, lj)$ *where* l *is the reducing factor.*
2	*Compute* Γ_{xy} *and* $u_l|_{\Gamma_{xy}}$. *Encode* Γ_{xy}. *Compute* $\tilde{u}_l|_{\Gamma_{xy}}$ *by applying an ENO encoding scheme, with maximum error* e, *to* $u_l|_{\Gamma_{xy}}$.
3	*Compute the interpolated image* $\tilde{u}_l$ *by solving Laplace equation with initial data* $(\Gamma_{xy}, \tilde{u}_l|_{\Gamma_{xy}})$.
4	*Compute and quantize the residual* r_l^q *between* u_l *and* $\tilde{u}_l$. *Let* $\hat{u}_l = \tilde{u}_l + r_l^q$ *be the approximation of* u_l *satisfying* $\sup\{|u_l - \hat{u}_l|\} = e$.
5	*Zoom out* $\hat{u}_l$*and compute and quantize the new residual* r^q *between* u *and* $\tilde{u}$ *in order to satisfy* $\sup\{|u - (\tilde{u} + r^q)|\} = e$.
6	*Finally compress* Γ_{xy}, $\tilde{u}_l|_{\Gamma_{xy}}$, r_l^q *and* r^q *using an arithmetic coder.*

TABLE III
ALGORITHM FOR COMPRESSING DEM DATA.

The compression ratios using this approach were already satisfactory. We observed that the encoding of the geometry represented the main cost in bits. This is due mainly to the irregularity of the curves and the inefficiency of the differential chain coding approach (3 bits/pixel with 8-connected curves or 2 bits/pixel with 4-connected curves). To further improve the encoding of the geometry we have adopted a simple multiscale approach. We

compute and encode the curves and the residuals in a subsampled image and then zoom out the result and recompute new residuals. If required, before sampling, we may filter the given image with a low pass filter or with an anisotropic filter like motion by mean curvature or affine invariant smoothing [38], [50]. Indeed, it is convenient to apply these filters while keeping some points fixed, namely, the points corresponding to the extrema values and the saddle points. In practice we may fix the multiple points of the sampled curves Γ_{xy}. Anisotropic filtering fixing points was studied in [51].

The zoom out process can be done by using a bicubic spline interpolation, although this can create new maxima and minima due to the well known oscillation problem of splines. In order to avoid this kind of errors we have used a *shape preserving spline*, which avoids the oscillation problem of classical splines, that is, respects the monotonicity of the original data (no new maxima or minima are created). Concretely, we have implemented the algorithm proposed in [52].

The complete algorithm for compressing DEM data is summarized in Table III.

VI. Compression results

In order to compare our results, we use the JPEG-LS standard for lossless and controlled lossy image compression, being this the only standard that permits a control on the maximal per pixel error [3]. We also compare with JPEG-2000 [5], in which, by reintroducing the errors, we are able to control the maximum error, while deviating from the standard specifications (that is, the complete compressed bit stream is not in standard format due to the error control part). In the comparison tables below, JPEG-LS is denoted by JLS, while ours is denoted by ME, standing for morphological encoding. We also report the $RMSE$ as frequently done for lossy image compression algorithms. We report results on a set of 10 DEM images of size 1200 by 1200 pixels with 8 bits per pixel. We also report the results of our algorithm when applied to the same set of images with a resolution of 16 bits per pixel (note that JPEG-2000 doesn't support this range, and the image has to be subdivided).

Figure 9 shows 10 plots (corresponding to a set of 10 different DEM images), each one

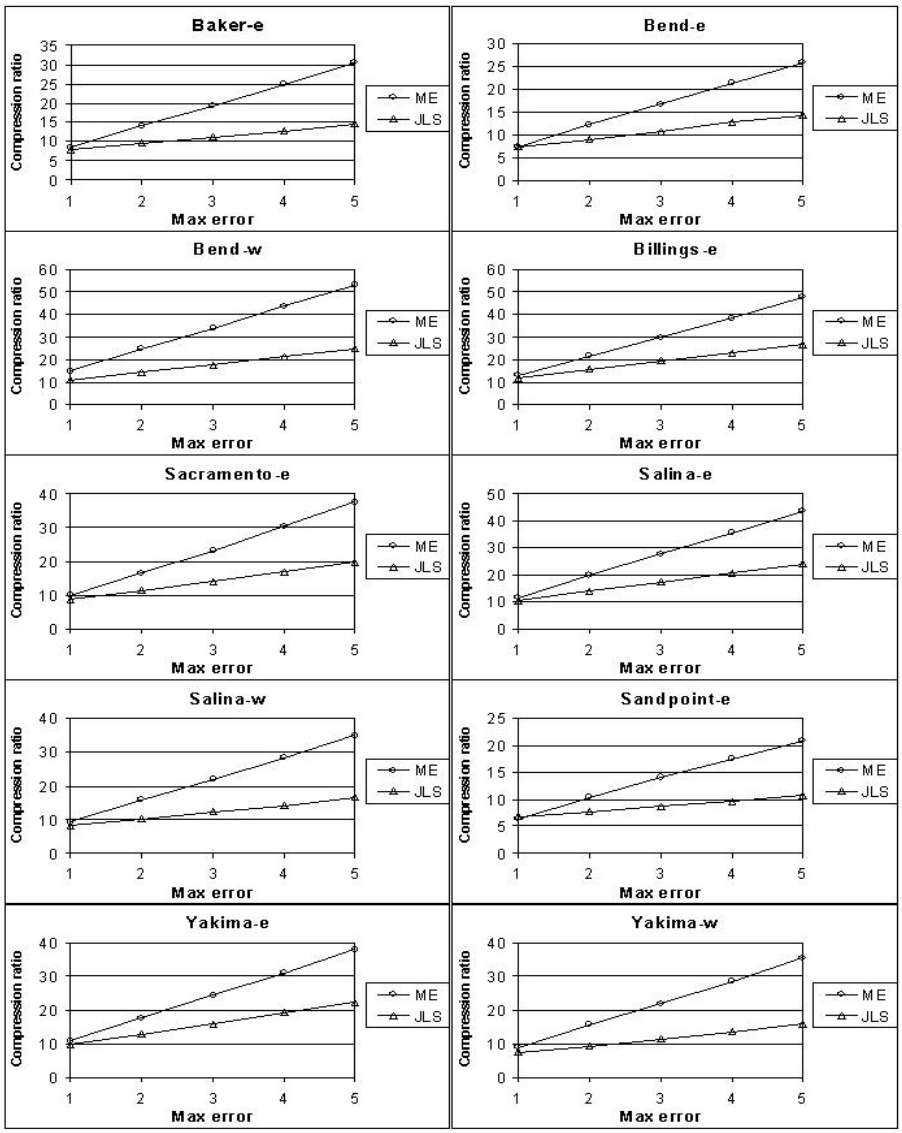

Fig. 9. Performance of the compression ratio of ME and JPEG-LS methods when allowing maximum errors ranging from 1 to 5. The amount in meters corresponding to a difference in gray level of 1 depends on the image, but typically this value may be of 10 meters.

showing the compression ratio (vertical axis) versus the maximum error allowed (horizontal axis) for ME and JPEG-LS. When $e = 1$ there is no significant difference between both methods, though as e increases ME outperforms JLS. In fact when e is greater than 1 the plots show that ME reaches almost twice (or more in some cases) the compression ratio (CR) of JLS.

Figure 10 shows a more detailed study for the case $e = 5$. In the left plot we can see the compression ratio of both methods for the set of 10 images, the right plot corresponds to

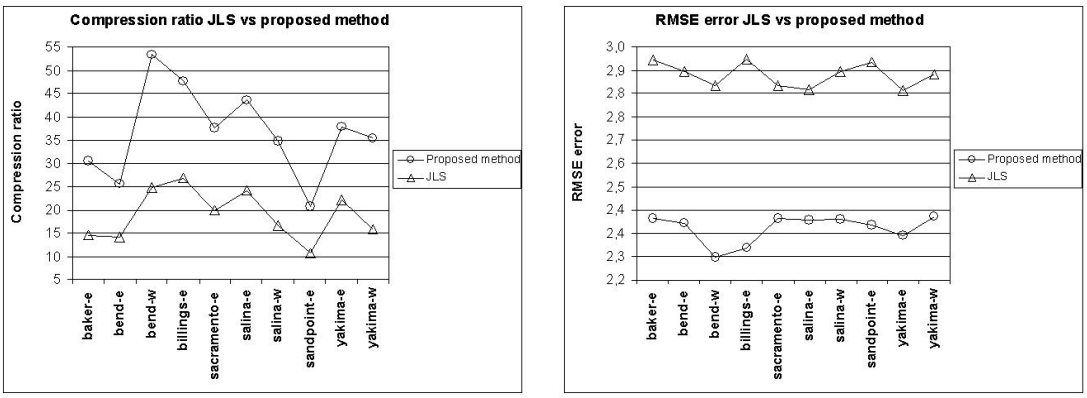

Fig. 10. Left: Compression ratio of JLS vs the proposed method for a set of ten images compressed with a maximum error of 5. Right: Corresponding root mean square error ($RMSE$) for the two methods.

the $RMSE$ for both methods. Table IV shows the exact values for the compression ratios and $RMSE$ errors of both methods. The last row shows the average values for the CR and $RMSE$ of JLS and ME. The average CR for the case of JLS and ME are **18,9872** and **36,7245** respectively. That is, our proposed scheme ME reaches almost twice the compression ratio of JLS. In addition, the average $RMSE$ for the cases of JLS and ME is **2,9038** and **2,3782**, respectively. The last two columns contain the compression ratio and the $RMSE$ in meters, respectively, corresponding to the above set of images quantized with 16 bits per pixel when compressed with ME (with $e = 5$ meters).

Table V shows the results obtained using $JPEG-2000$. Column 1 displays the maximum error obtained when using $JPEG-2000$ at the same compression rate as the one obtained with ME (which corresponded to a maximum error of 5), and displayed in the first column of Table IV. Recall that strictly following the JPEG-2000 specifications, this algorithm can not automatically control the maximal error. We can correct the error to be the same as the one obtained with ME (i.e., 5), as displayed in column 4. Column 2 contains the RMS error. Although this average error is lower than for ME, recall that the significant geometry of the DEM as encoded by ME is losslessly represented, outperforming JPEG-2000 in those critical areas.

Figure 11 displays the original image and its compressed versions using JLS, ME and $JPEG-2000$. The figure shows both the gray scale images and its level sets. Note that the topographic structures are better preserved in the case of the ME and $JPEG-2000$

	CR (ME) (8 bits)	CR (JLS) (8 bits)	$RMSE$ (ME)	$RMSE$ (JLS)	CR (ME) (16 bits)	RMSE (meters)
baker-e	30,4549	14,6585	2,4099	2,9546	11,1249	2,5825
bend-e	25,7138	14,2491	2,3946	2,9152	11,5988	2,5523
bend-w	53,4223	24,7324	2,2787	2,8677	15,8313	2,6217
billings-e	47,6631	26,8321	2,3121	2,9588	19,2838	2,5408
sacramento-e	37,6028	19,8254	2,4127	2,8664	9.0562	2,6763
salina-e	43,5954	24,2159	2,4062	2,8538	13,7127	2,5936
salina-w	34,8702	16,5692	2,4093	2,9166	11,5446	2,5988
sandpoint-e	20,7807	10,6721	2,3901	2,9491	9,6334	2,5501
yakima-e	37,7655	22,2050	2,3521	2,8488	17,0275	2,4659
yakima-w	35,3764	15,9119	2,4166	2,907	7,3900	2,7532
AVERAGE	**36,7245**	**18,9872**	**2,3782**	**2,9038**	**12,6203**	**2,5935**

TABLE IV

COLUMNS 1 TO 4: COMPRESSION RATIO (CR) AND $RMSE$ ERRORS FOR JLS AND ME ON 8 BIT IMAGES WHEN COMPRESSING WITH A MAXIMUM ERROR OF 5 GRAY VALUES. COLUMNS 5 AND 6 SHOWS THE PERFORMANCE OF THE ALGORITHM WHEN CONSIDERING THE FULL RANGE (16 BITS), IN THIS CASE THE ERROR CORRESPOND TO ELEVATION METERS. THE LAST ROW SHOWS THE AVERAGE VALUES FOR THE SET OF 10 IMAGES.

compression while it is severely distorted in the case of JLS compression.

VII. Conclusions

In this work we have presented techniques to compute a basic geometric representation of images. This representation is given by the Morse and drainage structures of the image. As an application, this geometric image representation was used to derive a non uniform sampling strategy that when combined with standard interpolation and coding techniques, provided a novel DEM compression algorithm. This algorithm produces compression ratios similar to a modified JPEG-2000 and permits to control the maximal error in the decoded image, a property which is fundamental for most DEM applications. In addition, a natural geometric representation of the DEM is obtained, which includes a

	L_∞ no correction	$RMSE$	L_∞ corrected	increment (bytes)
baker-e	11	1,0016	5	550
bend-e	11	1,0258	5	708
bend-w	9	0,8057	5	391
billings-e	12	0,8224	5	825
sacramento-e	10	0,9689	5	568
salina-e	11	0,8961	5	615
salina-w	10	0,9635	5	630
sandpoint-e	10	1,0081	5	628
yakima-e	9	0,8748	5	504
yakima-w	9	0,9599	5	424

TABLE V

First and second columns correspond to the L_∞ and $RMSE$ errors for the set of images in the first column compressed with the standard JPEG-2000 at the same compression ratio as the one achieved by our method (see text for more details on these RMSE). The third column corresponds to the L_∞ error after correcting the result achieved by JPEG-2000. The fourth column corresponds to the increment in bytes due to the correction.

lossless representation of its topology and drainage structure, unique characteristics of the method here proposed.

In order to improve the compression results here reported, the lossy compression of the topographic representation needs to be investigated. It is important to investigate how errors in the position of the critical level-lines and the drainage curves affect the maximal error in height. Further study of the particular lossless encoders used once the structure has been computed will help to improve in the compression.

We are currently investigating the use of geometric sampling techniques for the compression of natural images. Some of the criteria used for the compression of DEM data, e.g., the use of an L_∞ norm, are too stringent here. Other concepts, like the selection of drainage structures, are natural for DEM data, but not as much for other types of data. The question then is what is a good geometric representation of natural images

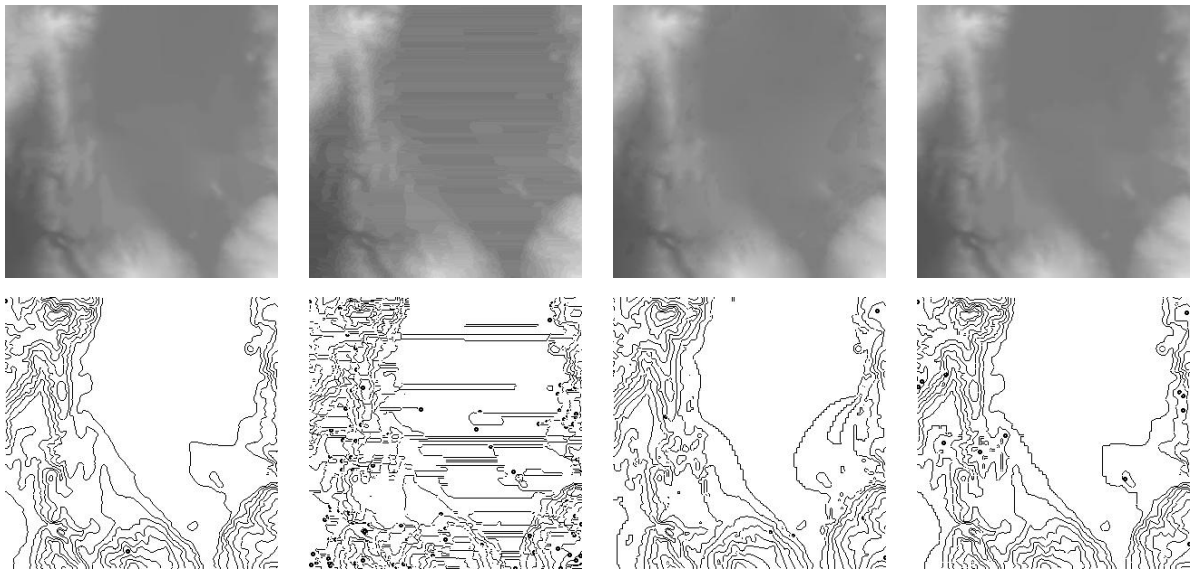

Fig. 11. Top row: From left to right selected region of the original DEM image *baker-e*, the same region after compressing with JLS with *sup* error 5, after compressing with ME with *sup* error 5 and after compressing with the corrected $JPEG-2000$ with *sup* error 5. Bottom row: Level lines of the above images.

that will lead to compression results as the ones obtained for DEM with the techniques here introduced.

Appendix

In this Appendix we give a result which explains why the Laplacian and the AMLE interpolations are in some sense *shape preserving*. For the detailed proof we refer to [40], [41]. We shall consider an interpolation operator as a transformation E which associates to each open bounded set Ω and each function $\varphi \in C(\partial\Omega)$ a function $E(\varphi,\Omega) \in C(\overline{\Omega})$ such that $E(\varphi,\Omega)|_{\partial\Omega} = \varphi$. We shall say that the interpolation operator satisfies the stability principle if

$$E(E(\varphi,\Omega)|_{\partial\Omega'},\Omega') = E(\varphi,\Omega)|_{\Omega'}$$

for any open bounded set Ω, any $\varphi \in \partial\Omega$, and any open bounded set $\Omega' \subseteq \Omega$. Suppose that the interpolation operator E satisfies the stability property, we say that E satisfies the maximum principle if

$$\inf_{\partial\Omega'} \varphi \leq \inf_{\overline{\Omega'}} E(\varphi,\Omega') \leq \sup_{\overline{\Omega'}} E(\varphi,\Omega') \leq \sup_{\partial\Omega'} \varphi \tag{3}$$

for any open bounded set Ω, any open bounded set $\Omega' \subseteq \Omega$, and any $\varphi \in C(\partial\Omega')$. If E satisfies the maximum principle and $\varphi = \alpha$ in $\partial\Omega$, where $\alpha \in I\!R$, then $E(\varphi, \Omega) = \alpha$ in $\overline{\Omega}$, and the same is true in any open bounded set $\Omega' \subseteq \Omega$.

Theorem 1: Let Ω_2 be an open simply connected set. Let Ω_1 be an open set whose boundary $\partial\Omega_1$ is connected and $\overline{\Omega_1} \subseteq \Omega_2$. Let $\Omega = \Omega_2 \setminus \overline{\Omega_1}$. Assume that $u|_{\partial\Omega_1} = \lambda$, $u|_{\partial\Omega_2} = \mu$ with $\lambda < \mu$ or $\lambda > \mu$. Let E be an interpolation operator satisfying the stability and the maximum principle. Then $E(u|_{\partial\Omega}, \Omega)$ contains only a monotone section in the sense that $\overline{\Omega}$ is a monotone section of $E(u|_{\partial\Omega}, \Omega)$. In particular, if $E(u|_{\partial\Omega}, \Omega)$ is the $AMLE$ extension of the boundary data inside Ω, then $E(u|_{\partial\Omega}, \Omega)$ contains only a monotone section.

VIII. Acknowledgments

The first and second authors acknowledge partial support by PNPGC project, reference BFM2000-0962-C02-01. G. Sapiro is partially supported by a grant from the Office of Naval Research ONR-N00014-97-1-0509, the Presidential Early Career Award for Scientists and Engineers (PECASE), and a National Science Foundation CAREER Award. G. Sapiro thanks Dr. Wen Masters from ONR and Dr. Carey Schwartz from DARPA for introducing him to the problem of DEM compression. F. Arándiga is partially supported by MCYT BFM 2001-2814.

References

[1] A. Said and W. A. Pearlman, "An image multiresolution representation for lossless and lossy comression," *IEEE Transaction on Image Processing*, vol. 5, no. 9, pp. 1303–1310, September 1996.

[2] M.J. Weinberger, G. Seroussi, and G. Sapiro, "LOCO-I: A low complexity, context-based, lossless image compression algorithm," *Proc. IEEE Data Compression Conf., Snowbird, Utah*, April, 1996.

[3] M.J. Weinberger, G. Seroussi, and G. Sapiro, "From LOCO-I to the JPEG-LS standard," *Proceedings of the International Conference on Image Processing*, vol. 4, pp. 68–72, 1999.

[4] S.D. Rane and G. Sapiro, "Evaluation of JPEG-LS, the new lossless and controlled-lossy still image compression standard, for compression of high-resolution elevation data.," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 10, pp. 2298–2306, October 2001.

[5] D. Taubman and M. Marcellin, *Jpeg2000 : Image Compression Fundamentals, Standards, and Practice*, Kluwer, The Netherlands, 2001.

[6] C. Ballester and V. Caselles, "The m-components of level sets of continuous functions in wbv," *Publicacions Matemàtiques*, vol. 45, pp. 477–527, 2001.

[7] C. Ballester, V. Caselles, and P. Monasse, "The tree of shapes of an image," *Preprint CMLA, Cachan, Paris*, 2001.

[8] P.J. Besl and R.C. Jain, "Segmentation through symbolic surface descriptions," *CVPR, May*, 1986.

[9] C. Bajaj, V. Pascucci, and D.R. Schikore, "The contour spectrum," *Proceedings Visualization '97*, pp. 167–173, 1997.

[10] C. Bajaj, V. Pascucci, and D.R. Schikore, "Fast isocontouring for improved interactivity," *In Proc. IEEE Symposium on Volume Visualization, San Francisco, Oct. 7-8*, pp. 39–46, 1996.

[11] V. Caselles, B. Coll, and J.M. Morel, "Topographic maps and local contrast changes in natural images," *Int. J. Comp. Vision*, vol. 33, no. 1, pp. 5–27, 1999.

[12] J.L. Cox and D.B. Karron, "Digital morse theory with suggested applications," *Manuscript available at http://www.casi.net*, 1998.

[13] W.R. Franklin and A. Said, "Lossy compression of elevation data.," *7th Int. Symposium on Spatial Data Handling*, 1996.

[14] R. Haralick, L. Winston, and T. Laffey, "The topographic primal sketch," *Int. J. Rob. Research*, vol. 2, 1983.

[15] I.S. Kweon and T. Kanade, "Extracting topograpic terrain features from elevation maps," *CVGIP: Image Understanding*, vol. 59, no. 2, pp. 171–182, 1994.

[16] M. van Kreveld, R. va Oostrum, C. Bajaj, V. Pascucci, and D. Schikore, "Contour trees and small seed sets for isosurface traversal," *In 13th ACM Symposium on Computational Geometry*, pp. 212–220, 1997.

[17] J.L. Lisani, "Comparaison automatique d'images par leurs formes," *Ph.D Thesis, Université de Paris-Dauphine, July*, 2001.

[18] P. Monasse and F. Guichard, "Fast computation of a contrast invariant image representation," *IEEE Trans. on Image Proc.*, vol. 9, pp. 860–872, 2000.

[19] P. Monasse, *Représentation morphologique d'images numériques et application au recalage*, Ph.D. thesis, Université de Paris-Dauphine, 2000.

[20] P. Salembier, "Morphological multiscale segmentation for image coding," *Signal Processing, Special Issue on Nonlinear Signal Processing*, vol. 38, pp. 359–386, 1994.

[21] P. Salembier and J. Serra, "Flat zones filtering, connected operators and filters by reconstruction," *IEEE Trans. Image Processing*, vol. 4, pp. 1153–1160, 1995.

[22] Y. Shinagawa, T.L. Kunii, and Y.L. Kergosien, "Surface coding based on morse theory," *IEEE Computer Graphics and Appl.*, vol. 11, no. 5, pp. 66–78, 1991.

[23] J. Roubal and T.K. Poiker, "Automated contour labelling and the contour tree," *In Auto-Carto, March*, 1985.

[24] S. Takahashi, T. Ikeda, Y. Shinagawa, T. Kunii, and M. Ueda, "Algorithms for extracting correct critical points and constructing topological graphs from discrete geographic elevation data," *Eurographics 95*, vol. 14, no. 3, pp. 181–192, 1995.

[25] G. Reeb, "Sur les poits singuliers d'une forme de pfaff completement integrable ou d'une fonction numérique," *Comptes Rendus Acad. Sciences Paris*, vol. 222, pp. 847–849, 1946.

[26] E.G. Johnston and A. Rosenfeld, "Digital detection of pits, peaks, ridges and ravines," *IEEE Trans. Systems Man Cybernetics*, vol. 472, no. July, 1975.

[27] V. Caselles, J. L. Lisani, J.M. Morel, and G. Sapiro, "Shape preserving local histogram modification," *IEEE Transactions on Image Processing*, vol. 8, no. 2, pp. 220–230, 1999.

[28] A. Desolneux, L. Moisan, and J.M. Morel, "Edge detection by helmholtz principle," *To appear at the Journal of Mathematical Imaging and Vision*, 2001.

[29] A.S. Kronrod, "On functions of two variables," *Uspehi Mathematical Sciences (NS)*, vol. 35, no. 5, pp. 24–134, 1950.

[30] L. Garrido P. Salembier, "Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval," *IEEE Transactions on Image Processing*, vol. 9, no. 4, pp. 561–576, 2000.

[31] F. Meyer P. Salembier, L. Torres and C. Gu, "Region-based video coding using mathematical morphology," *Proceedings of IEEE (Invited paper)*, vol. 83, no. 6, pp. 843–857, 1995.

[32] P. Salembier and F. Marqus, "Region-based representations of image and video: Segmentation tools for multimedia services," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1147–1169, 1999.

[33] E. Reusens, "Joint optimization of representation model and frame segmentation for generic video compression.," *EURASIP Signal Processing*, vol. 46, no. 11, pp. 105–117, 1995.

[34] J. Froment, "A functional analysis model for natural images permitting structured compression," *COCV*, vol. 4, pp. 473–495, 1999.

[35] J. Milnor, "Morse theory," *Annals of Math. Studies 51, Princeton University Press*, 1963.

[36] L. Vincent, "Morphological area openings and closings for gray-scale images," *Proc. of the Workshop "Shape in Picture",1992, Driebergen, The Netherlands, Springer-Berlin*, pp. 197–208, 1994.

[37] L. Vincent, "Gray scale area openings and closings, their efficient implementation and applications," *Proc. Workshop Mathematical Morphology and Applications to Signal Processing, Barcelona, Spain, May*, pp. 22–27, 1993.

[38] F. Guichard and J.M. Morel, "Partial differential equations and image iterative filtering," *Ceremade, 9535, Université Paris-Dauphine, France*, 1995.

[39] V. Caselles and P. Monasse, "Grain filters," *to appear in Journal of Mathematical Imaging and Vision*, 2002.

[40] Andrés Solé, *Geometric Image Coding, Filtering and Restoration*, Ph.D. thesis, Universitat Pompeu Fabra, 2002.

[41] V. Caselles, G. Sapiro, and A. Solé, "Morse description and morphological encoding of continuous data," *Preprint*, 2002.

[42] A. López, F. Lumbreras, J. Serrat, and J. Villanueva., "Evaluation of methods for ridge and valley detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, 1999.

[43] W.W. Seemuller, "The extraction of ordered vector drainage networks from elevation data," *Comp. Vision Graphics and Image Processing*, vol. 47, 1989.

[44] V. Caselles, J.M. Morel, and C. Sbert, "An axiomatic approach to image interpolation," *IEEE Transactions on Image Processing, Special Issue on PDE's, Geometry Driven Diffusion and Image Processing*, vol. 7, no. 3, pp. 376–386, 1998.

[45] R. R. Estes Jr. and V. Ralph Algazi, "Efficient error free chain coding of binary documents," *Proceedings of the Data Compression Conference, Snowbird, Utah*, pp. 122–132, April 1995.

[46] C. Lu and J. Dunham, "Highly efficient coding schemes for contour lines based on chain code representation," *IEEE Trans. Commun.*, vol. 39, no. 10, pp. 1511–1514, 1991.

[47] B. B. Chaudhuri and S. Chandrashekhar, "Neighboring direction runlength coding: An efficient contour coding scheme.," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, no. 4, pp. 916–921, July 1990.

[48] G.M. Schuster and A.K. Katsaggelos, *Rate-distortion based video compression*, Kluwer Academic Publishers, 1997.

[49] F. Aràndiga and R. Donat, "Nonlinear multiscale decompositions: The approach of A. Harten," *Numerical Algorithms*, vol. 23, pp. 175–216, 2000.

[50] G. Sapiro and A. Tannenbaum, "Affine invariant scale-space," *International Journal of Computer Vision*, vol. 11, pp. 25–44, 1993.

[51] V. Caselles, B. Coll, and J.M. Morel, "A kanisza programme," *Progress in Nonlinear Diferential Equations and their Applications*, vol. 25, 1996.

[52] J. M. Hyman, "Monotonicity preserving cubic interpolation," *SIAM J. Sci. Stat. Comp.*, vol. 4, no. 4, 1983.