

Level Lines Selection with Variational Models for Segmentation and Encoding

Coloma Ballester (coloma.ballester@upf.edu)

Dept. de Tecnologia, Universitat Pompeu-Fabra, Passeig de Circumvalació 8, 08003 Barcelona, Spain

Vicent Caselles (vicent.caselles@upf.edu)

Dept. de Tecnologia, Universitat Pompeu-Fabra, Passeig de Circumvalació 8, 08003 Barcelona, Spain

Laura Igual (laura.igual@upf.edu)

Dept. de Tecnologia, Universitat Pompeu-Fabra, Passeig de Circumvalació 8, 08003 Barcelona, Spain

Luis Garrido (luis.garrido@upf.edu)

Dept. de Tecnologia, Universitat Pompeu-Fabra, Passeig de Circumvalació 8, 08003 Barcelona, Spain

Abstract. This paper discusses the interest of the Tree of Shapes of an image as a region oriented image representation. The Tree of Shapes offers a compact and structured representation of the family of level lines of an image. This representation has been used for many processing tasks such as filtering, registration, or shape analysis. In this paper we show how this representation can be used for segmentation, rate distortion optimization, and encoding. We address the problem of segmentation and rate distortion optimization using Guigues algorithm on a hierarchy of partitions constructed using the simplified Mumford-Shah multiscale energy. To segment an image, we minimize the simplified Mumford-Shah energy functional on the set of partitions represented in this hierarchy. The rate distortion problem is also solved in this hierarchy of partitions. In the case of encoding, we propose a variational model to select a family of level lines of a gray level image in order to obtain a minimal description of it. Our energy functional represents the cost in bits of encoding the selected level lines while controlling the maximum error of the reconstructed image. In this case, a greedy algorithm is used to minimize the corresponding functional. Some experiments are displayed.

Keywords: mathematical morphology, tree structure, segmentation, rate distortion, morphological encoding, minimal description length.

1. Introduction

In most image processing based applications, an image is usually viewed as a set of pixels placed on a rectangular grid. The pixel provides a extremely local information: taking it as elementary unit places the scale of representation far from the interpretation or decision scale. In recent years, an increasing number of applications rely on region based image representations. For instance, in MPEG-4 [13] or MPEG-7 [43] standards, the image is understood as a set of objects. Region-based image representations offer two advantages with respect to pixel based ones: the number of regions is much lower than the number of original pixels, and regions represent a first level of abstraction with respect to the raw information.

A key concept in region based representations is the concept of partition of the image domain. Let us mention two data structures which have proved to be useful in region based image processing: the region adjacency graph and tree based structures. The region adjacency graph (RAG) is the right data structure needed to encode a partition: the nodes represent regions and two nodes are connected by an edge if their associated regions are neighbors. Merging algorithms remove some links and merge the corresponding nodes, thus they transform a RAG into another one. Thus, starting with an initial partition organized as a RAG, and merging regions according to an homogeneity criterion we derive another partition represented by a RAG. This is the basic structure of many segmentation algorithms. The main drawback of this structure is that it can only represent a single scale of the image. To overcome it, tree based structures have

been developed to represent a hierarchy of partitions. Quadtrees, the Critical Lake Tree [27], or Partition Trees [37, 38] are examples of such structures.

One of the most sound alternatives to pixel based representations of images comes from mathematical morphology. According to Mathematical Morphology, an image u is a representative of an equivalence class of images v obtained from u via a contrast change, i.e., $v = g(u)$ where g , for simplicity, will be a continuous strictly increasing function [19, 42]. The contrast of an image depends on the sensor's properties, on the lightning conditions, on the objects' reflection properties, etc., and these conditions are usually unknown. This led the physicist and gestaltist M. Wertheimer [49] to state as a principle that the grey level is not an observable. Images are observed up to an arbitrary and unknown contrast change. Mathematical Morphology recognized contrast invariance as a basic requirement and proposed that image analysis operations should take into account this invariance principle [42]. Under this assumption, an image is characterized by its level sets (see Section 2) which constitute the basic objects for image processing and analysis. In order to account for local changes in illumination a more local description of the basic objects of an image is required. With this purpose several authors [8, 39, 42] proposed to consider the connected components of (upper or lower) level sets as basic objects of the image. In most cases, a connected component of a level set can be described in terms of its boundaries which are Jordan curves: we call them level lines. By Sard's theorem, this is the case if the image u is a smooth function, but more general cases can be included with the right definition of level lines.

The family of level lines of the image can be given a tree structure since they are ordered by inclusion. This is essentially the tree of shapes of the image [29, 31]. It gives a complete and non-redundant representation of the image and is contrast independent. The tree of shapes merges into a single tree the information contained in the trees of connected components of upper and lower level sets, called max and min tree, respectively. Even if we identify it as the tree of shapes, this structure is nothing else than a region adjacency graph for the level lines of the image.

Using level lines as basic objects of the image, many image processing tasks like edge detection, segmentation, or rate distortion browsing can be restricted to these objects. Indeed, edge detection computed as a subfamily of level lines has been the object of several works [6, 15], and the computed edges have been used for recognition purposes [7]. The main purpose of this paper is to use the family of level lines as basic atoms for segmentation and rate/distortion browsing tasks. For that, we shall rely on the tree of shapes as the basic data structure for handling them. While edge detection is based on contrasted boundaries, we base our segmentation on a homogeneity criterion expressed in terms of an energy and we compute a partition determined by level lines minimizing this energy. If we minimize it with respect to all possible partitions, the common strategies to minimize it are based on a greedy algorithm and give a local minimum, called 2-normal segmentation in [32]. The problem of finding a global minimum is exponentially complex. But, if the minimization takes place in a hierarchy of partitions, global minima can be obtained [20, 21, 37].

Binary partition trees are a first example of hierarchical partitions and have been used for a large number of applications such as filtering, segmentation, information retrieval, or visual browsing [35, 36, 37, 38, 39]. In particular, as shown in [37, 38], partition trees are appropriate to define optimum pruning strategies in the rate/distortion sense with restriction on the rate to be transmitted or the distortion of the coded image. Its usefulness and efficiency for segmentation purposes has been illustrated in [37]. Both, segmentation and browsing based on rate distortion are two examples of problems where a global optimum can be found in the set of partitions

represented by the tree, and would have an exponential complexity on the set of all partitions of the image domain (even in the set of all trees that could be constructed from an initial partition).

Assuming that a hierarchy of partitions is given as a binary partition tree \mathcal{H} , Salembier and Garrido [37] use a Lagrangian formulation to solve the rate distortion problem [40]. If $D(P)$ and $C(P)$ represent the distortion and the cost of encoding a given partition P , to solve the problem of finding a partition P minimizing $D(P)$ such that $C(P) = \gamma$, $\gamma > 0$, the authors define the Lagrangian $E^\lambda = \lambda C + D$ and use a dynamic programming strategy to find an optimal pruning of \mathcal{H} for a fixed value of $\lambda > 0$. Using a gradient descent, they find the value of λ and an associated partition P such that $C(P) \approx \gamma$ and P minimizes $D(P)$. A different type of algorithm which finds the lower boundary of the convex hull of the set of achievable distortion/rate pairs was proposed by Chou, Lookabaugh, and Gray in [14]. The proposed algorithm is a generalization of the BFOS algorithm ([5]) for optimally pruning a tree [14, 21]. In a recent work, Guigues [20] uses a dynamic programming principle to compute the optimal pruning of \mathcal{H} with respect to the energy E^λ , with an algorithm which is able to compute at the same time the optimal prunings with respect to any value of λ . The three algorithms, the dynamic programming strategies proposed in [20, 37] and the generalized BFOS algorithm [14, 21] can be used for finding an optimal segmentation as a minimum of an energy E in a hierarchy of partitions when some assumptions on E are satisfied (essentially E is the sum of a sub-additive and a super-additive criterion).

The three algorithms described above compute an optimal partition given the hierarchy of partitions \mathcal{H} . In [37], the authors suppose that the hierarchy of partitions, in their case, a binary partition tree, is constructed by using a merging algorithm with a merging criterion (based on a homogeneity criterion, for instance, applied to an initial partition of the image domain in flat zones) which has no particular relation with the optimization problem to be solved. In his work [20], Guigues proposes an strategy to simultaneously construct the hierarchy and, using a dynamic programming principle, minimize the energy E^λ on it. At each step, a region is added to the hierarchy depending on the result of the dynamic programming optimization. The construction of the hierarchy depends on the energy itself. This is called the climbing strategy. At the end, one can compute from the constructed tree the optimum of E^λ for any λ .

In our case, the regions determined by level lines are taken as a initial partition of a hierarchy which can be constructed using the simplified Mumford-Shah multiscale energy. Then, using Guigues optimization algorithm [20], the global minima of the energy in the hierarchy can be obtained at any scale. We use a similar approach to consider the problem of rate/distortion browsing.

Using level lines as basic atoms permits to obtain segmentations compatible with the level line structure of the image. This may be useful for some level set based algorithms, as in [23], where region based computation of optical flow is computed by matching the vector fields of unit normals to the level lines. It is important here that the regions of the segmentation are determined by level lines. Let us finally mention that many other image processing tasks can be efficiently addressed by means of morphological methods: contrast enhancement [9], filtering [19, 42, 44, 46], compression [17, 36], segmentation [28, 35, 47], intersection [2], or registration [30], to give some examples.

We also explore the problem of optimal selection of level lines with respect to a Minimum Description Length functional, but in this case we can only compute a locally optimal solution. For that we construct an energy functional which measures the cost in bits of encoding the family of level lines B , the gray values on them $u|_B$, and the quantized error between u and an approximate reconstruction of it based on (B, u_B) . The minimum of E provides us with a

minimal description of u in the language defined by its level lines. This method can be considered a Minimal-Description-Length functional ([26]) subordinated to the Topographic Map of the image. This minimal description represents an (lossy) encoding of u .

Finally, let us explain the plan of the paper. In Section 2 we review the basic definitions of Mathematical Morphology required to define the tree of shapes of an image. Section 3 is devoted to explain the algorithm introduced by L. Guigues [20] to minimize a multiscale energy on a hierarchy of partitions. In Section 4 we apply the above ideas to minimize the simplified version of the Mumford-Shah functional subordinated to the topographic map of the image and we display some experiments. The Mumford-Shah energy is minimized using Guigues optimization algorithm on a hierarchy of partitions obtained also with the help of simplified Mumford-Shah energy. The minimum obtained is a global one on the constructed hierarchy. In Section 5 we apply the previous algorithm on the topographic map to solve the rate distortion problem and we display some experiments comparing our results with the method used in [37]. Finally, in Section 6 we introduce a general energy functional for the selection of a minimal set of curves based on a formulation of the Minimum Description Length principle subordinated to the topographic map of the image. The functional measures the cost of encoding the selected family of level lines B , the average value (or a polynomial approximation) of u on the connected components of $\Omega \setminus B$, call it u_B , and the cost of encoding the quantized errors between u and u_B . We observe that this functional is free of parameters besides the maximum error allowed in the quantizer. Some experiments on the encoding and compression of digital elevation models are presented.

2. The Tree of Shapes of an image

A gray level image can be realistically modeled as a real function $u : \Omega \rightarrow \mathbb{R}$ where x represents an arbitrary point of a domain $\Omega \subset \mathbb{R}^N$ ($N = 2$ for usual snapshots, 3 for medical images or movies) and $u(x)$ denotes the gray level at x . To simplify, we shall assume that $N = 2$ and that the image domain Ω is a finite rectangle in \mathbb{R}^2 . As we have explained in the Introduction, images are observed up to an arbitrary and unknown contrast change and image analysis and processing operations should take into account the contrast invariance principle. Under this assumption, an image is characterized by its upper (or lower) level sets

$$[u \geq \lambda] = \{x \in \Omega, u(x) \geq \lambda\} \quad (\text{resp. } [u < \lambda] = \{x \in \Omega, u(x) < \lambda\}), \quad \lambda \in \mathbb{R}.$$

Moreover, the image can be recovered from its upper level sets by the reconstruction formula

$$u(x) = \sup\{\lambda : x \in [u \geq \lambda]\},$$

and a similar formula exists for the lower level sets. As it is easily seen, the family of level sets (upper or lower) of u is invariant under continuous strictly increasing contrast changes [19, 42].

In order to have a more local description of the basic objects of an image, we are led to consider the connected components of (upper or lower) level sets as the basic objects of the image. This led to the introduction of the Topographic Map as the family of connected components of upper (or lower) level sets $[u \geq \lambda]$ (resp. $[u < \lambda]$). Moreover, the family of connected components of upper level sets has a tree structure. And the same happens for the family of connected components of lower level sets. These two trees can be merged in a single tree: the ‘‘Tree of Shapes’’ of an image [29, 31]. In order to review this notion, we have to define the concept of hole of a set, and the notion of saturation. From that, the shapes of an image, defined as the saturation of the connected components of its level sets, have a tree structure, [3, 29, 31], and

the tree is equivalent to the image: the knowledge of the tree is sufficient to reconstruct the image. Let us briefly review these notions.

If A is a set in a topological space, $\text{int}(A)$, \bar{A} and ∂A will denote, respectively, the interior, the closure and the boundary of A .

Even if we shall restrict ourselves to the case of discrete images, let us describe the main notions of the tree in the case of images defined on a continuous domain of \mathbb{R}^2 , since this will also cover the discrete case. Let $\bar{\Omega}$ be a closed rectangle of \mathbb{R}^2 , and let Ω be the interior of $\bar{\Omega}$. To fix ideas we may assume that $\bar{\Omega} = [0, 1]^2$. We recall that the Jordan curve theorem holds in $\bar{\Omega}$ [25].

We shall assume that the image $u : \bar{\Omega} \rightarrow \mathbb{R}$ is an upper semi-continuous function, that is, we assume that its upper level sets $[u \geq \lambda]$, $\lambda \in \mathbb{R}$ are closed sets. Equivalently, the lower level sets $[u < \lambda]$ are open sets. We shall explain below how does this translate at the discrete level.

We may always transform a discrete image $u(i, j)$, $i \in \{1, \dots, N\} \times \{1, \dots, M\}$ into an upper semi-continuous function in $\bar{\Omega}$ by defining $u(x_1, x_2) = u(i, j)$ when $(x_1, x_2) \in (\frac{i-1}{N}, \frac{i}{N}) \times (\frac{j-1}{M}, \frac{j}{M})$, and taking at a common boundary between two pixels (resp., at a corner) the highest value of the two (resp., of the four).

Given a point $x \in [u < \lambda]$ (resp. $x \in [u \geq \lambda]$), we denote $cc([u < \lambda], x)$ (resp. $cc([u \geq \lambda], x)$) the connected component of $[u < \lambda]$ (resp. $[u \geq \lambda]$) that contains x .

Heuristically, the tree of shapes is a data structure to encode in a tree the family of level lines of the image. To be able to handle discontinuous functions, more specifically, upper semicontinuous ones, we define level lines as the external boundary of the level sets of the image. This leads us to the the notion of shape which consists in filling the holes of the connected components of the level sets, upper or lower, of u . The operation of hole filling was called saturation in [3, 29]. Thus, level lines are the boundaries of shapes and to give the family of level lines is equivalent to give the family of shapes. It is easy to imagine them when the image is smooth (its graph is a smooth topography).

DEFINITION 1. *Let $A \subset \bar{\Omega}$. We call holes of A in $\bar{\Omega}$ the components of $\bar{\Omega} \setminus A$. Let $p_\infty \in \bar{\Omega} \setminus A$ be a reference point, and let T be the hole of A in $\bar{\Omega}$ containing p_∞ . We define the saturation of A with respect to p_∞ as the set $\bar{\Omega} \setminus T$ and we denote it by $\text{Sat}(A, p_\infty)$. We shall refer to T as the external hole of A and to the other holes of A as the internal holes. By extension, if $p_\infty \in A$, by convention we define $\text{Sat}(A, p_\infty) = \bar{\Omega}$. Note that $\text{Sat}(A, p_\infty)$ is the union of A and its internal holes.*

The reference point p_∞ acts as a point at infinity. In all what follows, we assume that the point $p_\infty \in \bar{\Omega}$ on which the saturations are based is fixed, i.e., all saturations will be computed with respect to p_∞ . To simplify the notation, we shall write $\text{Sat}(A)$ instead of $\text{Sat}(A, p_\infty)$. We shall also speak of holes of A instead of holes of A in $\bar{\Omega}$. We refer to [3, 29] for an overview on the properties of the saturation operator.

We recall some concepts introduced in [3, 29].

DEFINITION 2. *Given an image u , we call shapes of inferior (resp. superior) type the sets*

$$\text{Sat}(cc([u < \mu], x)) \quad (\text{resp.} \quad \text{Sat}(cc([u \geq \lambda], x)))$$

where $\mu, \lambda \in \mathbb{R}, x \in \bar{\Omega}$. We call shapes of u any shape of inferior or superior type. We denote by $\mathcal{S}(u)$ the family of shapes of u .

As it is proved in [11], even if the definition of saturation depends on p_∞ , the family of shapes does not. Observe that since $\bar{\Omega}$ is unicoherent the boundary of any shape of an image is connected. Moreover, as it is proved in [3, 29], if a shape S is closed, then $S = \text{Sat}(\partial S)$. This is the mathematical translation of the fact that a shape is essentially equivalent with its boundary, the level line.

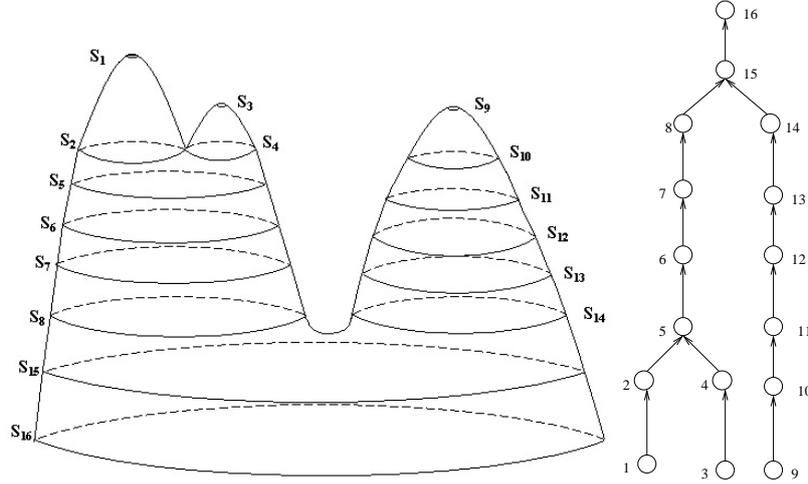


Figure 1. An example of a tree. a) Left: the graph of a function with its corresponding shapes. b) Right: the tree of shapes corresponding to the Figure on the left, the nodes of the tree are numbered following a post-order traversal.

THEOREM 1. ([3, 29]) *Any two shapes are either disjoint or nested.*

From this result, we can conclude that the set of shapes of an (upper semi-continuous) image has an inclusion tree structure. For simplicity, we assume that our image is discrete. Then we can represent the tree as a finite structure; the shapes are the tree nodes and the parent-child relationship, represented by the links between nodes, is determined by inclusion (the child A being a shape contained in the father A^f with no other shape B such that $A \subseteq B \subseteq A^f$). The root of the tree is

$$\bar{\Omega} = \text{Sat}([u \geq \min_{\Omega} u])$$

And there is no loop: if A, B_1, B_2, C are shapes and $A \subseteq B_i \subseteq C, i = 1, 2$, then $B_1 \cap B_2 \neq \emptyset$, hence the sets B_1 and B_2 must be nested.

Based on these results P. Monasse proposed in [29, 31] a data structure to encode the family of shapes of the image, or in a more informal way, its level lines. This data structure is like a region adjacency graph for the level lines of the image.

As we explained above, level lines are defined as boundaries of shapes and not as connected components of iso-level sets $[u = \lambda]$ because the image may have discontinuities. Let us express this idea with an example: If A is a connected region and u is the binary image $u(i, j) = 255$ if $(i, j) \in A$, and $u(i, j) = 0$ if $(i, j) \notin A$, then there are no iso-level sets $[u = \lambda]$ when $\lambda = 1, 2, \dots, 254$, but the boundary of the connected components of $[u \geq \lambda]$ for $\lambda = 1, 2, \dots, 254, 255$ gives us the boundary of A . Much more complex situations may appear in real images. To be able to compute these level lines which would appear at the walls of a cliff we use the definition of level lines as external boundaries of shapes. Moreover, the sets $[u = \lambda]$ may be quite dispersed and it can be difficult that its connected components have a real curve structure.

Even if the tree of shapes can be considered as a fusion of the upper and lower level sets, maxima and minima are not highlighted in this representation. All level lines are represented and highlighting one structure or another depends more on the operator used on the tree or on the functional minimized on it.

3. Optimization of a multiscale energy on a hierarchy of partitions

There are several alternative but related strategies to minimize an energy on a hierarchy of partitions, see [14, 20]. We shall follow here the approach in [20]. Let Ω be the image domain, and let $\mathcal{P}(\Omega)$, $\mathcal{Part}(\Omega)$ denote the family of subsets of Ω and partitions of Ω , respectively.

DEFINITION 3. *Let $P_0 \in \mathcal{Part}(\Omega)$. We say that \mathcal{H} is hierarchy of partitions of Ω constructed over P_0 if \mathcal{H} is a family of nonempty subsets of Ω such that*

- (i) $\Omega \in \mathcal{H}$
- (ii) Any two sets in \mathcal{H} are either nested or disjoint.
- (iii) Any set in \mathcal{H} contains a set in P_0 .

A family \mathcal{H}' of nonempty subsets of Ω satisfying (ii) and (iii) is called a pre-hierarchy over P_0 .

A cut of \mathcal{H} is a partition of Ω whose elements are in \mathcal{H} . We shall assume that the hierarchies we consider are finite, i.e., we assume that \mathcal{H} has a finite number of elements. In this case, \mathcal{H} is a tree whose nodes are the subsets of Ω in \mathcal{H} . Two nodes are related by an edge (of the tree) if one is contained in the other and no other set in the hierarchy is in between. The sets in P_0 are the leaves of the tree, Ω is the root, and the concepts of father, children and siblings apply.

DEFINITION 4. *We say that $E^\lambda : \mathcal{Part}(\Omega) \rightarrow \mathbb{R}^+$ is an affine energy on $\mathcal{Part}(\Omega)$ if there exist two functions $C, D : \mathcal{Part}(\Omega) \rightarrow \mathbb{R}^+$ and $\lambda \in \mathbb{R}^+$ such that $E^\lambda(P) = \lambda C(P) + D(P)$ for any $P \in \mathcal{Part}(\Omega)$. In this case, we denote $E^\lambda \approx (C, D, \lambda)$*

DEFINITION 5. *We say that $E : \mathcal{Part}(\Omega) \rightarrow \mathbb{R}^+$ is separable if there exists a function on the subsets of Ω which we denote by E such that*

$$E(P) = \sum_{R \in P} E(R) \quad \forall P \in \mathcal{Part}(\Omega).$$

We say that $E : \mathcal{P}(\Omega) \rightarrow \mathbb{R}^+$ is subadditive if

$$E(R \cup S) \leq E(R) + E(S) \quad \forall R, S \subseteq \Omega \text{ such that } R \cap S = \emptyset.$$

DEFINITION 6. *Let $E^\lambda \approx (C, D, \lambda)$ be an affine energy. We say that E^λ is a multiscale energy if C, D are separable and C is subadditive. The value λ is called the scale parameter of the energy.*

From now on we assume that $E^\lambda \approx (C, D, \lambda)$ is a multiscale energy. We assume that the multiscale energy is defined on the cuts of \mathcal{H} . For any λ , let $C_\lambda^*(\mathcal{H})$ be the cut of \mathcal{H} minimizing E_λ . Let us review the algorithm proposed by Guigues in [20] to compute $C_\lambda^*(\mathcal{H})$ for any $\lambda > 0$ which is based on a the dynamic programming functional relation.

For each $R \in \mathcal{H}$, let

$$\mathcal{H}(R) = \{S \in \mathcal{H} : S \subseteq R\}.$$

We call $\mathcal{H}(R)$ the partial hierarchy on the node R . As it is proved in [20], if $R \in C_\lambda^*(\mathcal{H})$ then R is locally optimal in \mathcal{H} , that is, $E^\lambda(R) \leq E^\lambda(Y)$ for any cut Y of the partial hierarchy $\mathcal{H}(R)$. Let $P_\lambda^*(\mathcal{H})$ the set of nodes of \mathcal{H} which are locally optimal in \mathcal{H} for the energy E^λ .

Let

$$\Lambda^*(R) := \{\lambda \in \mathbb{R}^+ : R \in C_\lambda^*(\mathcal{H})\}.$$

The set $\Lambda^*(R)$ represents the set of scales such that R is in the cut of \mathcal{H} minimizing E_λ .

Let

$$\Lambda_{up}^*(R) := \{\lambda \in \mathbb{R}^+ : R \in P_\lambda^*(\mathcal{H})\}.$$

The set $\Lambda_{up}^*(R)$ represents the set of scales for which R is locally optimal in \mathcal{H} for the energy E^λ . As proved in [20], $\Lambda_{up}^*(R)$ is an interval of type $[a, \infty)$. We denote by $\lambda^+(R)$ the left point of the interval and we refer to it as the scale of apparition of R in an optimal cut of the multiscale energy E^λ . Then Guigues [20] proved the following result:

PROPOSITION 1. *For any $R \in \mathcal{H}$, $\Lambda_{up}^*(R) = [\lambda^+(R), \lambda^-(R))$ where $\lambda^-(R) = \min_{S \in \mathcal{H}: R \subseteq S} \lambda^+(S)$. Thus*

$$C_\lambda^*(R) = \{R \in \mathcal{H} : \lambda^+(R) \leq \lambda < \lambda^-(R)\}.$$

We call the set $\Lambda_{up}^*(R)$ the interval of persistence of the region R . The persistent hierarchy obtained from \mathcal{H} and E^λ is

$$\mathcal{H}^* := \{R \in \mathcal{H} : \Lambda_{up}^*(R) \neq \emptyset\}.$$

On the persistent hierarchy \mathcal{H}^* we have $\lambda^-(R) = \lambda^+(R^f)$ where R^f denotes the father of R in \mathcal{H}^* .

For each $R \in \mathcal{H}$, $\lambda \in \mathbb{R}^+$, we define

$$E(\lambda, R) = \lambda C(R) + D(R).$$

We define the partial energy of the node $R \in \mathcal{H}$ as the energy of the optimal cut of $\mathcal{H}(R)$ with respect to E^λ and we denote it by $E^*(\lambda, R)$. That is

$$E^*(\lambda, R) = E^\lambda(C_\lambda^*(\mathcal{H}(R))).$$

Observe that for any leave R of the hierarchy we have $E^*(\lambda, R) = E(\lambda, R)$ for any $\lambda \in \mathbb{R}^+$.

PROPOSITION 2. *The partial energies $E^*(\lambda, R)$ of the nodes of \mathcal{H} are related by the dynamic programming equation*

$$E^*(\lambda, R) = \inf\{E(\lambda, R), \sum_{S \in \mathcal{F}(R)} E^*(\lambda, S)\} \quad \text{for any } R \in \mathcal{H},$$

where $\mathcal{F}(R)$ is the family of children of R .

PROPOSITION 3. *Assume that $E^\lambda \approx (C, D, \lambda)$ is a multiscale energy on the hierarchy \mathcal{H} . Then for any $R \in \mathcal{H}$ we have*

- (i) $E^*(\lambda, R)$ is a piecewise affine, nondecreasing, continuous and concave function of λ .
- (ii) We have $E^*(\lambda, R) = \sum_{S \in \mathcal{F}(R)} E^*(\lambda, S)$ if $\lambda < \lambda^+(R)$, while $E^*(\lambda, R) = E(\lambda, R)$ for any $\lambda \geq \lambda^+(R)$.

(iii) If C is strictly subadditive, i.e., if $C(X) < \sum_{Y \in \mathcal{F}(X)} C(Y)$ for any $X \in \mathcal{H}$, then $\lambda^+(R) \in \mathbb{R}$ and is the only solution of

$$E(\lambda, R) = \sum_{S \in \mathcal{F}(R)} E^*(\lambda, S).$$

Combining the results of Propositions 1, 2, 3 we are able to compute the λ -cuts $C_\lambda^*(\mathcal{H})$.

The above algorithm can be implemented once we have the hierarchy as it happens with the algorithms used in [14, 37]. Usually this hierarchy is constructed with a different merging algorithm [37]. On the contrary, the climbing algorithm proposed by Guigues [20] constructs the hierarchy at the same time that it implements the dynamic programming principle of Proposition 2. Our approach will be to construct the hierarchy from an initial partition using the mergings obtained with a greedy optimization algorithm for the simplified Mumford-Shah energy (see Section 4) at several scales. Then we use Guigues algorithm described in Propositions 1, 2, 3 to obtain the minimum of the energy on this hierarchy at any scale λ . A more precise description of this construction will be given in next Section. This approach can be used for computing multiscale segmentations (with the simplified Mumford-Shah energy) and to compute the solution to the rate/distortion problem.

4. The simplified Mumford-Shah functional on the Topographic Map

The idea of computing a segmentation by selecting a subset of the family of level lines of u can be applied to the simplified version of Mumford-Shah energy functional, leading to a version of it subordinated to the Topographic Map of the image.

According to Mumford-Shah, [33], a segmentation of an image $u : \Omega \rightarrow \mathbb{R}$ is defined as a pair (B, \tilde{u}) where \tilde{u} is piecewise regular function, regular in $\Omega \setminus B$, and B is a the set of boundaries where \tilde{u} is discontinuous. The set of curves B represents a partition of the image domain Ω . In particular, if we assume that \tilde{u} is piecewise constant, then $\Omega \setminus B$ is a union of regions and \tilde{u} takes a constant value on each of them which is equal to the mean value of u on it. We define the simplified Mumford-Shah functional E_{MS}^λ as

$$E_{MS}^\lambda(B, \tilde{u}|_{\Omega \setminus B}) = \lambda \mathcal{H}^1(B) + \int_{\Omega \setminus B} (u - \tilde{u})^2 \quad (1)$$

where $\mathcal{H}^1(B)$ denotes the length of the system of curves B , \tilde{u} is a piecewise constant image, i.e., constant on each region of $\Omega \setminus B$, and $\lambda > 0$ is a parameter. We observe that, given B , the minimum of E_{MS}^λ with respect to the variable \tilde{u} is explicitly given by

$$\tilde{u} = \sum_{O_i} u_{O_i} \chi_{O_i} \quad \text{where} \quad u_{O_i} = \frac{1}{|O_i|} \int_{O_i} u \, dx, \quad (2)$$

O_i being the connected components of $\bar{\Omega} \setminus B$ (as usual, for any set O , $\chi_O(x) = 1$ if $x \in O$, $\chi_O(x) = 0$, if $x \notin O$). This observation permits us to write the energy as a function of B and denote it by $E_{MS}^\lambda(B)$ instead of $E_{MS}^\lambda(B, \tilde{u}|_{\Omega \setminus B})$. This energy is a multiscale energy which can be written as $E_{MS}^\lambda \approx (C, D, \lambda)$ where

$$C(B) = \mathcal{H}^1(B), \quad D(B) = \int_{\Omega \setminus B} (u - \tilde{u})^2 \, dx. \quad (3)$$

Observe that $C(B)$ is strictly subadditive.

We shall restrict us to the case of digitized images, i.e., we assume that the domain $\Omega = \{1, \dots, N\} \times \{1, \dots, M\}$, $N, M \in \mathbb{N}$, and the image $u : \Omega \rightarrow \{1, \dots, L\}$, $L \in \mathbb{N}$. Let $\mathcal{S}(u)$ be the tree of shapes of u . Observe that any set of shapes $\mathcal{T} \subseteq \mathcal{S}(u)$ can be endowed with a tree structure whose nodes are the shapes in \mathcal{T} , two consecutive shapes of \mathcal{T} being related by an edge. Let

$$\mathcal{ST}(u) := \{\mathcal{T} : \mathcal{T} \subseteq \mathcal{S}(u)\}.$$

Let us denote

$$\partial\mathcal{T} = \cup_{A \in \mathcal{T}} \partial A.$$

We consider the minimization of (1) restricted to the set $\{\partial\mathcal{T} : \mathcal{T} \in \mathcal{ST}(u)\}$, i.e.,

$$\min_{B=\partial\mathcal{T}, \mathcal{T} \in \mathcal{ST}(u)} E_{MS}^\lambda(B). \quad (4)$$

Minimizing the simplified Mumford-Shah functional subordinated to the topographic map is a segmentation which contains a similarity criterion and computes regions whose boundaries are level lines. This is not the most general context for a segmentation, since boundaries of objects may be bounded by curves formed by pieces of level lines and may not coincide with full level lines. In spite of this, level lines are robust and contrast invariant objects, and the main edges of the image are contained in them. Let us comment on a context where this could be useful: we have used this level lines based segmentation as a tool for a region based contrast invariant optical flow estimation [23]. Our optical flow estimation is based on the matching the normals to the level lines (hence level lines) and is region based, thus it is important that the regions we use are bounded by level lines, and the segmentation presented here is a tool to compute an initial partition of the image [23].

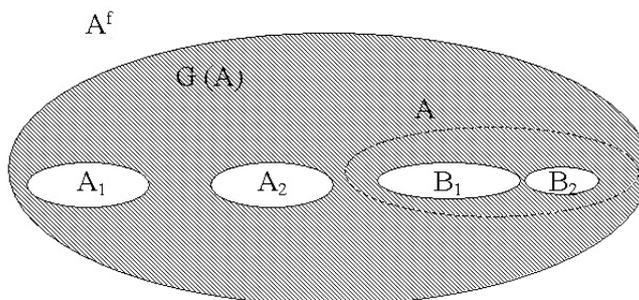


Figure 2. The domain $G(A)$ obtained after suppression of the shape A . It is the region determined by the father of A , denoted by A^f , the external shape in the Figure, the siblings of A , denoted by A_1, A_2 and the children of A , denoted by B_1, B_2 .

Observe that the computation of the optimum has an exponential complexity on the number of shapes if all possible combinations of them are taken into account. As explained in Section 3, this computation becomes feasible if we restrict our search space to a hierarchy of partitions of Ω , and we shall do this. Starting with the initial partition determined by $\partial\mathcal{S}(u)$ we construct a hierarchy using the mergings produced by a greedy algorithm applied to the energy (1) at several scales λ_k , $k \geq 1$. The greedy algorithm produces a local minimum of (4) and the hierarchy will contain all the merging steps to compute the local minima at several scales. Then the algorithm described in Section 3 will compute the global minimum of E_{MS}^λ on this hierarchy for any value

of λ . Notice that the global optimum corresponding to $\lambda = \lambda_k$ does not necessarily coincide with the local one obtained using the greedy algorithm.

The basic operation of the greedy algorithm is the merging of two neighboring regions which, in the present context is equivalent to the suppression of a shape. Given $\mathcal{T} \in \mathcal{ST}(u)$, the suppression of a shape A in \mathcal{T} gives $\mathcal{T} \setminus \{A\} \in \mathcal{ST}(u)$. Let us describe this operation as a merging of two regions of $\Omega \setminus \partial\mathcal{T}$. For that, let A^f be the father of A , let $\{B_1, \dots, B_p\}$ be the children of A , and let $\{A_1, \dots, A_k\}$ be the siblings of A . It is implicitly understood that, if A is a leaf of \mathcal{T} , the family of children of A is empty. Similarly, it may happen that the family of siblings of A is empty. The shape A determines two regions

$$A^u = \overline{A^f} \setminus \cup \left(A \cup \cup_{i=1}^k \overline{A_i} \right), \quad A^d = \overline{A} \setminus \cup \left(\cup_{i=1}^p \overline{B_i} \right).$$

and the merging of these two regions produces the region (see Figure 2)

$$G(A) = A^u \cup A^d = \overline{A^f} \setminus \cup \left(\cup_{i=1}^p \overline{B_i} \cup \cup_{i=1}^k \overline{A_i} \right)$$

Let us describe the greedy algorithm proposed in [24, 32] which finds a local minimum of (4). Since this algorithm could be applied to any energy, let us denote it by E instead of E_{MS}^λ . Let

$$\Delta E(\mathcal{T}, A) = E(\mathcal{T}) - E(\mathcal{T} \setminus \{A\}).$$

A greedy algorithm: Set $\mathcal{T}_0 = \mathcal{S}(u)$.

Step 1: For any $A \in \mathcal{T}_0$ compute $\Delta E(\mathcal{T}_0, A)$ and insert it in a queue Q with priority $\Delta E(\mathcal{T}_0, A)$, the highest priority corresponding to the highest value of $\Delta E(\mathcal{T}_0, A)$.

Step 2: Iterate the following procedure: Choose the shape $A^* \in \mathcal{T}_i$ which corresponds to the first element in the queue constructed in *Step 1* if $\Delta E(\mathcal{T}_i, A^*) > 0$, and define $\mathcal{T}_{i+1} = \mathcal{T}_i \setminus \{A^*\}$. Recompute the values of $\Delta E(\mathcal{T}_{i+1}, A')$ for all shapes A' which are adjacent to A^* (i.e., parent, children, or siblings of A^*) and reorder again the queue in decreasing order of the values $\Delta E(\mathcal{T}_{i+1}, A)$, $A \in \mathcal{T}_{i+1}$ (the highest priority corresponding to the highest value). We stop when no shape A^* exists with $\Delta E(\mathcal{T}_i, A^*) > 0$.

The last tree obtained \mathcal{T}^* determines the boundaries and the regions of the segmentation. It is a local optimal solution of (4), in the sense that any other merging of regions of the segmentation increases the energy [24, 32].

Construction of a hierarchy. Since E_{MS}^λ is a multiscale energy, we can compute its minimum on a hierarchy of partitions using Guigues algorithm [20] (see Section 3). To explain our construction of the hierarchy of partitions let us recall the definition of completion. Let $P_0 \in \mathcal{Part}(\Omega)$ and let \mathcal{H} be a pre-hierarchy over P_0 . The operation of adding to \mathcal{H} a node R constructed by merging two regions of \mathcal{H} without father is called a completion. Then we start with the initial partition P_0 determined by $\partial\mathcal{S}(u)$ and we take the pre-hierarchy $\mathcal{H}' := \{R : R \in P_0\}$. Then we choose $\lambda_1 > 0$ and we minimize $E_{MS}^{\lambda_1}$ using the greedy algorithm, adding to the hierarchy \mathcal{H}' the completions corresponding to the merging of neighboring regions performed during the execution of the algorithm. Let P_1 be the locally optimal solution obtained. We continue iteratively this process by minimizing the simplified Mumford-Shah energy $E_{MS}^{\lambda_{k+1}}$, $\lambda_{k+1} = 2\lambda_k$, $k \geq 1$, (one could also use $\lambda_{k+1} = \lambda_k + \Delta$, for some value $\Delta > 0$) on the initial partition P_k using the greedy algorithm and storing the successive mergings as nodes of the hierarchy. The construction may be stopped either when the value of λ_k attains a maximum scale value λ_{max} , or when we reach the set Ω . The value of λ^+ at each node is computed using Propositions 2 and 3. Then, using Proposition

1, we are able to compute the λ -cuts on the constructed hierarchy for any $\lambda > 0$. These λ -cuts are local minima of E_{MS}^λ ; they are also global minima when restricted to the hierarchy. The implementation of this algorithm is based on the results of Guigues [20]. It can be used for computing multiscale segmentations and to compute the solution of the rate/distortion problem.

4.1. EXPERIMENTS OBTAINED MINIMIZING FUNCTIONAL E_{MS}^λ

We display some results obtained by minimizing the simplified version of the Mumford-Shah functional E_{MS}^λ given by (4). The functional E_{MS}^λ is minimized on a hierarchy of partitions constructed with the algorithm described in Section 4. To construct it we started with the value $\lambda_1 = 2$ and updated it with $\lambda_{k+1} = 2\lambda_k$, $k \geq 1$, up to a maximal scale which gives the region Ω as segmentation. For each experiment, we shall display the original image, the boundaries of the segmentation B and the image \tilde{u} which takes the mean value of u on each region of the segmentation.

The energy functional is a multiscale one. The value of λ determines the minimal size of the regions of the segmentation [32]. If we do not know a priori this size, by taking (for instance) $\lambda = 2^k$ we can obtain a multiscale family of segmentations of the image which contain the information at several scales [24, 32]. For the reasons of display we selected some reasonable value of λ depending on the image. Using a different λ near to the one we used will not change much the results. One could filter the hierarchy of partitions so that all regions obtained have a minimal size, but this is not related with the optimization of the functional on the hierarchy.

Figure 3 displays the results obtained minimizing (4) (applied to Lena image in Figure 3.a) with $\lambda = 100$. Figure 3.b displays the set of curves B obtained, Figure 3.c displays the reconstruction \tilde{u} .

Figure 4 displays the results obtained minimizing (4) (applied to the image in Figure 4.a) with $\lambda = 60$. Figure 4.b displays the set of curves B obtained, Figure 4.c displays the reconstruction \tilde{u} .

Figure 5 displays the results obtained minimizing (4) (applied to the image in Figure 5.a) with $\lambda = 200$. Figure 5.b displays the set of curves B obtained, Figure 5.c displays the reconstruction \tilde{u} .

Figure 6 displays the results obtained minimizing (4) (applied to the image in Figure 6.a) with $\lambda = 50$. Figure 6.b displays the set of curves B obtained, Figure 6.c displays the reconstruction \tilde{u} .

5. Rate Distortion optimization on the tree of shapes

Rate distortion theory is a branch of information theory addressed to relate the *distortion* (reconstruction error) of a fixed-length coding scheme to the data rate (e.g., number of bits per pixel) used in the scheme. Under the assumption that the input image is continuous, the distortion will never be zero within a finite data rate, because of quantization error. While it does not specify the form of optimal coders, rate distortion theory does present guidelines about the conditions under which the best performance is achieved. If a lossy compression method is used, the recovered image $g(x)$ differs from the original $f(x)$. This difference (the distortion) is normally quantified by the mean square error of reconstruction:

$$D = E[(f(x) - g(x))^2]$$

If we ask for a maximum allowed distortion D^* then, the bit rate required in the coding scheme has the corresponding lower bound $R(D^*)$ which is a monotone decreasing function of D^* . Recall



Figure 3. Result obtained minimizing E_{MS}^λ with $\lambda = 100$ using the algorithm described in Section 4. a) Left: Original image. b) Middle: Segmentation boundaries. c) Right: the image \tilde{u} .



Figure 4. Result obtained minimizing E_{MS}^λ with $\lambda = 60$ using the algorithm described in Section 4. a) Left: Original image. b) Middle: Segmentation boundaries. c) Right: the image \tilde{u} .



Figure 5. Result obtained minimizing E_{MS}^λ with $\lambda = 200$ using the algorithm described in Section 4. a) Left: Original image. b) Middle: Segmentation boundaries. c) Right: the image \tilde{u} .



Figure 6. Result obtained minimizing E_{MS}^λ with $\lambda = 50$ using the algorithm described in Section 4. a) Left: Original image. b) Middle: Segmentation boundaries. c) Right: the image \tilde{u} .

that (for any memoryless source and single-letter distortion measure), the source coding theorem tells us that for any $\epsilon > 0$, there exists an r , and code of block length r and rate $R < R(D^*) + \epsilon$ such that the average per letter distortion denoted d , is such that $d \leq D^* + \epsilon$. $R(D^*)$ is called the rate distortion function. The inverse function will be denoted by $D(R)$. For a full account on the Rate Distortion Theory, we refer to [4, 40].

One of the fundamental problems is to minimize the rate (number of necessary bits to store an image), taking into account that the distortion between the given image and the compressed image does not exceed a given threshold D^* . This problem can be written as the following constrained optimization problem:

$$\min R(D), \quad \text{subject to: } D \leq D^*. \quad (5)$$

The associated dual problem is known as the optimal bit allocation problem, one tries to minimize the distortion under the constraint of a fixed data rate. To put us in the algorithmic framework that we shall describe below, let \mathcal{S}_b a finite set which represents the possible encoding structures and $B \in \mathcal{S}_b$ be a member of that set. Let $R(B)$ and $D(B)$ be the rate and distortion functions expressed in terms of the variable $B \in \mathcal{S}_b$. If R_{\max} represents the number of available bits, we may write the dual problem as

$$\min_{B \in \mathcal{S}_b} D(B), \quad \text{subject to} \quad R(B) \leq R_{\max}. \quad (6)$$

To fix ideas, given the image u , we take $\mathcal{S}_b = \mathcal{ST}(u)$, and for each $B \in \mathcal{S}_b$, let the reconstructed image \tilde{u} be given by (2). We measure the distortion in terms of the mean square error between u and \tilde{u} , i.e.,

$$D(B) = \int_{\Omega \setminus B} (u - \tilde{u})^2 dx. \quad (7)$$

The rate can be measured in terms of the cost in bits of encoding the curves B (which we may write as proportional to its length) plus the cost of encoding the gray level values in the regions of $\Omega \setminus B$. That is

$$R(B) = \beta \mathcal{H}^1(B) + \mathcal{G}(\tilde{u}|_{\Omega \setminus B}), \quad (8)$$

where β represents the cost in bits for each direction, usually $\beta = 2$ or 3 , and the second term, $\mathcal{G}(\tilde{u}|_{\Omega \setminus B})$, denotes the encoding cost of the gray values u_{O_i} , a constant value for each region of $\Omega \setminus B$. The number of bits needed to encode each value u_{O_i} is constant (for instance, 8 bits, if u is a gray level image with 256 possible gray level values).

Let us restrict to consider the optimal bit allocation problem (6). The complexity of using all possible partitions determined by the sub-trees of $\mathcal{ST}(u)$ is exponential. For that reason, we shall restrict our space of candidate solutions to a hierarchy of partitions obtained with the algorithm described in Section 4. For that, we observe that the energy $E^\lambda := D(B) + \lambda R(B)$, $B \in \mathcal{S}_b$, is a multiscale energy and the strategy that uses the greedy algorithm on a given set of scales (see Section 4) can be applied to construct a hierarchy \mathcal{H}_{rd} . Then, using Section 3, we compute the optima of E^λ on \mathcal{H}_{rd} for any given value of λ (observe that once the hierarchy is constructed, any of the algorithms described in [21, 37] could be used). Observe that the energy E^λ incorporates the budget constraint with the *Lagrange multiplier* λ , and we can use the *secant iterative method* to compute the value of λ which gives the specified rate. Let us explain this.

Redefine \mathcal{S}_b as the set of $B \in \mathcal{ST}(u)$ which can be obtained by pruning the hierarchy \mathcal{H}_{rd} . In the present context and following [40], the Lagrangian multiplier theorem states that, for any $\lambda \geq 0$, the optimal solution $B^*(\lambda)$ to the unconstrained problem,

$$\min_{B \in \mathcal{S}_b} (D(B) + \lambda R(B)) , \quad (9)$$

is also an optimal solution of the constrained problem,

$$\min_{B \in \mathcal{S}_b} D(B) , \quad \text{subject to} \quad R(B) \leq R(B^*(\lambda)). \quad (10)$$

Thus, for every non-negative λ , there exists a corresponding constrained problem whose solution is identical to that of the unconstrained problem. If $R(B^*(\lambda))$ happens to be equal to the upper bound R_{\max} , then $B^*(\lambda)$ is the desired solution of (6).

The algorithm to find the λ such that $R(B^*(\lambda)) = R_{\max}$ relies on the following result (see [40], pp. 57-58, and also [41]).

THEOREM 2. *If $R(B^*(\lambda_1)) > R(B^*(\lambda_2))$, then*

$$\lambda_2 \geq -\frac{D(B^*(\lambda_1)) - D(B^*(\lambda_2))}{R(B^*(\lambda_1)) - R(B^*(\lambda_2))} \geq \lambda_1. \quad (11)$$

As a consequence, $R(B^(\lambda))$ and $D(B^*(\lambda))$ are, respectively, non-increasing and non-decreasing functions of λ .*

Theorem 2 can be used to find the optimum value of λ . Indeed, since $R(B^*(\lambda))$ is a non-increasing function of λ , the bisection method can be used to find an optimal value of λ [40]. In practice, since \mathcal{S}_b is a finite set, the bisection method is stopped when a solution λ is found such that

$$R_{\max} - \text{error} \leq R(B^*(\lambda)) \leq R_{\max} + \text{error} \quad (12)$$

for some specified value of $\text{error} \geq 0$, where R_{\max} is the maximum number of bits available, i.e., the budget. As in [18], we have used the secant iterative method (a gradient search algorithm) to search for a value of λ satisfying (12). The secant method starts with two initial guesses for λ , λ_l and λ_h which have to be selected such that $R(B^*(\lambda_l)) \geq R_{\max} \geq R(B^*(\lambda_h))$. In practice, [18], the algorithm starts with a high value (we have used $\lambda_h = 10^6$) and a low value (we have used $\lambda_l = 2$) of λ . For each value of λ , the optimum of E^λ on \mathcal{H}_{rd} is computed, and also the values $D(B^*(\lambda))$ and $R(B^*(\lambda))$. For λ_h (respectively, λ_l), the optimum resulting segmentation should be a coarse partition associated to the root node (resp., a fine segmentation associated to leaf nodes). The partitions correspond to rates $R(B^*(\lambda_h))$ and $R(B^*(\lambda_l))$, and should be below

and above the budget R_{\max} , respectively. If none of these rates is closed enough to the budget, a new Lagrange parameter is defined as

$$\lambda = -\frac{D(B^*(\lambda_h)) - D(B^*(\lambda_l))}{R(B^*(\lambda_h)) - R(B^*(\lambda_l))}$$

The procedure is iterated until the rate satisfies (12).

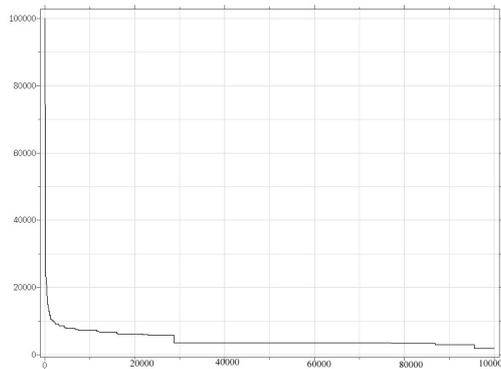


Figure 7. The rate $R(B^*(\lambda))$ for the bureau image of Figure 4.a for the of values of λ multiples of 10 up to $\lambda = 100000$.

We stress the fact that, as it is shown in [40], the above algorithm gives us the solution of problem (6) if we are able to compute the global minimum of (9). Observe that the algorithm of Section 4 is able to compute the global minimum of (9) on \mathcal{S}_b . In Figure 7 we display the rate $R(B^*(\lambda))$ for the bureau image of Figure 4.a for a set of values of λ (we have chosen the multiples of 10). We see that, as stated in Theorem 2, $R(B^*(\lambda))$ is a decreasing function of λ .

Figures 8 and 9 correspond to the minimization of (10) with a rate restriction of 40Kbits, 44Kbits, respectively, with an error = 1Kbits. These rate restrictions correspond to compression rates around 15.36, 11.91, respectively. The actual rates attained are 40348, 44218 bits, respectively. The corresponding PSNR are: 26.25, 28.32 dB. The left Figure corresponds to the family of selected level lines, the right Figure is the associated reconstruction, the value on each region being the average of the gray levels of the original image on that region. For these experiments, we have chosen $\beta = 2$, that is we encode the curves with two bits per direction. We have also checked that with an arithmetic encoding of B and the values of \tilde{u} on the regions of $\Omega \setminus B$, the actual compression rate is higher than the prescribed one.

We repeated the experiment using the method proposed in [18, 37]: the rate-distortion functional is minimized on a binary partition tree constructed by merging flat zones (which are connected components of iso-level sets). The experiments are displayed in Figures 10 and 11. We asked for the same rates as in the experiments displayed in Figures 8, 9, i.e., 40 and 44 Kbits, respectively. The actual rates attained are 40102 and 44086 bits, respectively. The corresponding PSNR are: 24.57 and 25.71 dB.

6. Minimum Cost Selection of Level Lines

Let $\Omega = \{1, \dots, N\} \times \{1, \dots, M\}$, $N, M \in \mathbb{N}$, and the image $u : \Omega \rightarrow \{1, \dots, L\}$, $L \in \mathbb{N}$.

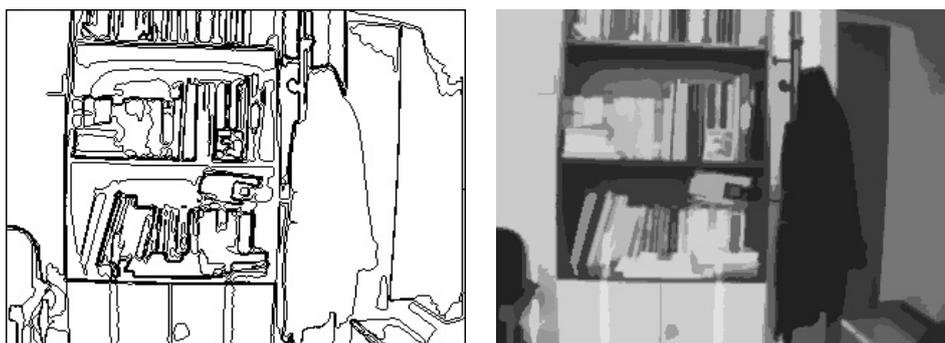


Figure 8. a) Left: Level lines selected by (10) asking for a rate of 40 Kbits (with an error of ± 1 Kbits). b) Right: Piecewise constant reconstructed image. The actual rate attained is 40348 bits and the PSNR is 26.25 dB.

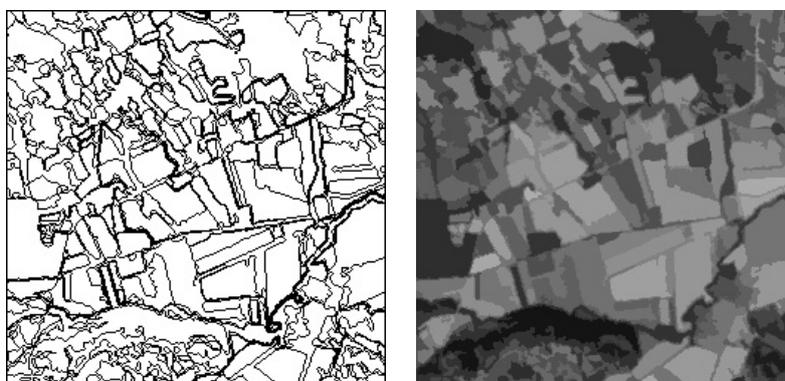


Figure 9. a) Left: Level lines selected by (10) asking for a rate of 44 Kbits (with an error of ± 1 Kbits). b) Right: Piecewise constant reconstructed image. The actual rate attained is 44218 bits and the PSNR is 28.32 dB.

Our goal is to obtain a minimal description of the image subordinated to the tree of shapes of u . For that, our algorithm will select the boundaries of a sub-family of shapes in the tree

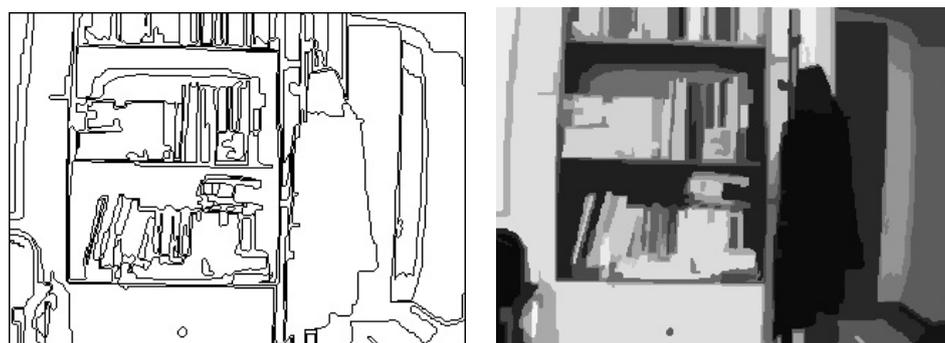


Figure 10. a) Left: Level lines selected using the approach in [18, 37] asking for a rate of 40 Kbits (with an error of ± 1 Kbits). The rate-distortion functional is minimized on a binary partition tree constructed by merging flat zones (which are connected components of iso-level sets). b) Right: Piecewise constant reconstructed image. The actual rate attained is 40102 bits and the PSNR is 24.57 dB.

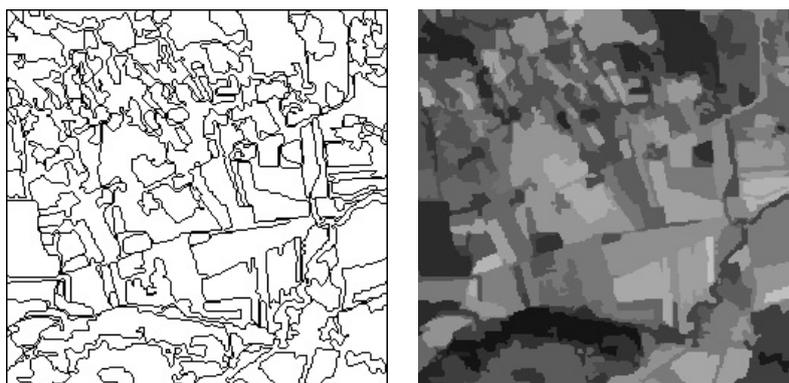


Figure 11. a) Left: Level lines selected using the approach in [18, 37] asking for a rate of 44 Kbits (with an error of ± 1 Kbits). b) Right: Piecewise constant reconstructed image. The actual rate attained is 44086 bits and the PSNR is 25.71 dB.

which minimize the number of bits necessary to encode a simplified version of the image. This simplified version is formed by (a) the selected family of curves, (b) some values of u which permit to reconstruct the image on the regions determined by the curves in (a), for instance, the gray levels on the curves, or a value on each region, and (c) the quantized errors between the original image and an estimation obtained by interpolating the data given by (a) and (b). The errors will be quantized so that the maximum error between the original and the reconstructed image is controlled. As we shall discuss below, this process depends obviously on the choice of the interpolation algorithm.

For each $\mathcal{T} \subseteq \mathcal{S}(u)$, let $\partial\mathcal{T} = \cup_{A \in \mathcal{T}} \partial A$, and let $u|_{\partial\mathcal{T}}$ be the values of u on \mathcal{T} . Either we use an interpolation operator which permits us to reconstruct an approximation to u in $\Omega \setminus \partial\mathcal{T}$ from the data $(\partial\mathcal{T}, u|_{\partial\mathcal{T}})$, or we directly store an approximation to u in each connected component (call it region) of $\Omega \setminus \partial\mathcal{T}$. Let O_i , $i = 1, \dots, r$, be the connected components of $\bar{\Omega} \setminus \partial\mathcal{T}$. Let $u_{\mathcal{T}}^{O_i}$ be an approximation to u in O_i and let

$$u_{\mathcal{T}}(x) = u_{\mathcal{T}}^{O_i}(x) \quad \text{if } x \in O_i, i \in \{1, \dots, r\}. \quad (13)$$

The most simple example is

$$u_{\mathcal{T}}^{O_i}(x) = u_{O_i} \quad \text{when } x \in O_i, \quad \text{where } u_{O_i} = \frac{1}{|O_i|} \int_{O_i} u \, dx. \quad (14)$$

Thus, $u_{\mathcal{T}}$ is given by the average of u on the connected components of $\Omega \setminus \partial\mathcal{T}$. More general cases can be considered if $u_{\mathcal{T}}$ is a polynomial of a given degree on each region.

Given the allowed maximal error $\delta \in \{1, \dots, L\}$, we shall denote by Q the quantization operator

$$Q(e) = \text{sign}(e) \lfloor |e|/\delta \rfloor \delta, \quad e \in \mathbb{R}. \quad (15)$$

We propose to obtain a simplified description of the image by minimizing the following minimal description length energy functional subordinated to the tree of shapes of u

$$E(\partial\mathcal{T}, u_{\mathcal{T}}) = \mathcal{C}(\partial\mathcal{T}) + \mathcal{G}(u_{\mathcal{T}}) + \mathcal{D}(Q(u - u_{\mathcal{T}})), \quad (16)$$

where $\mathcal{C}(\partial\mathcal{T})$ corresponds to the cost of encoding the set of curves $\partial\mathcal{T}$, $\mathcal{G}(u_{\mathcal{T}})$ denotes the cost of encoding $u_{\mathcal{T}}$, and the $\mathcal{D}(Q(u - u_{\mathcal{T}}))$ is the cost of encoding the quantized interpolation errors

$Q(u - u_{\mathcal{T}})$. To simplify, we shall assume that $u_{\mathcal{T}}$ is given by (13) and (14). Let us recall that the entropy gives a lower bound of the cost in bits/symbol of a lossless code for a sequence of symbols of an alphabet.

The curves are encoded using either a chain coding, or a differential chain coding scheme. In both cases, the cost of encoding the curves $\partial\mathcal{T}$ is given by the formula

$$\mathcal{C}(\partial\mathcal{T}) = \mathcal{E}(\partial\mathcal{T})\mathcal{H}^1(\partial\mathcal{T}), \quad (17)$$

where $\mathcal{H}^1(\partial\mathcal{T})$ denotes the length of the system of curves $\partial\mathcal{T}$, and $\mathcal{E}(\partial\mathcal{T})$ is the entropy of the statistical distribution of directions in $\partial\mathcal{T}$, in case of chain coding; or the entropy of the distribution of differences of directions in $\partial\mathcal{T}$ if we use a or differential chain coding strategy to encode it. We have tested both cases with slightly better results for the differential chain coding scheme; and we displayed the results corresponding to this case in the experiments in Section 6.2. An accurate analysis of the cost per contour point is given in [16] for a directional encoding of a 4-connected boundary. This cost is $\log_2 3 \approx 1.5849$ bits per contour point if we use an unconstrained 4-connected contour code, or a number between $\log_2 3$ and $\log(1 + \sqrt{2}) \approx 1.2716$ if we incorporate some first order constraints on the impossibility of certain consecutive configurations described in [16]. We have also made experiments using the cost $\mathcal{C}(\partial\mathcal{T}) = (\log_2 3) \mathcal{H}^1(\partial\mathcal{T})$ with results comparable to the differential chain coding scheme (see Section 6.2).

The term $\mathcal{G}(u_{\mathcal{T}})$ denotes the encoding cost of the gray values u_{O_i} , a constant value for each region of $\Omega \setminus \partial\mathcal{T}$. The number of bits needed to encode each value u_{O_i} is constant and equal to $\lceil \log_2 L \rceil + 1$ (for instance, 8 bits, if u is a gray level image with 256 possible gray level values).

Finally, let us describe the last term in (16). If $\delta \in \{0, \dots, L\}$ is the allowed maximum error, and Q denotes the quantization operator given by (15), then the quantized errors between $u_{\mathcal{T}}$ and u are

$$\epsilon(x) = Q(u - u_{\mathcal{T}})(x), \quad \text{where } x \in \Omega.$$

We denote by $\mathcal{D}(\epsilon)$ the cost in bits of the quantized error. Since the error is quantized with a quantization step equal to δ , we have $K := \text{range of the error}/\delta$ possible values for ϵ , and we can compute the error's probability distribution and entropy (which gives us the average codeword length). The cost in bits of encoding the error is given by the entropy of $\epsilon(x)$, denoted by $\mathcal{E}(\epsilon)$, multiplied by the number of pixels of the image, i.e., by

$$\mathcal{D}(\epsilon) = \mathcal{E}(\epsilon) \cdot N \cdot M. \quad (18)$$

Observe that, if we define the reconstructed image as $u_{rec} = u_{\mathcal{T}} + \epsilon$, then this construction guarantees that

$$\|u_{rec} - u\|_{\infty} \leq \delta.$$

The energy (16) measures the length of the image description, i.e., the length (or cost in bits) of the image model, plus the cost associated to the errors between the true image and its approximated version. In any case, this criterion of simplicity for determining the image description is a *Minimum-Description-Length* (MDL) criterion [34].

The energy (16) is simple enough to be able to address it numerically. We compute a local minimum of it using the greedy algorithm described in Section 4. We are unable to compute a global minimum, and this deserves further exploration. In spite of this, the result is probably near the optimal one at least when computing on a hierarchy of partitions. The only reason we have to say this is that the results obtained when minimizing the simplified Mumford-Shah

functional with the greedy algorithm are not far from the ones obtained when minimizing the functional on the hierarchy described in Section 4.

6.1. FURTHER REMARKS ON THE MODEL

We may also start with a given interpolation operator \mathcal{I} which permits to reconstruct an approximation to the image in Ω from the data $(\partial\mathcal{T}, u|_{\partial\mathcal{T}})$, and we denote this interpolated image as $u_{\mathcal{T}}^* = \mathcal{I}(\partial\mathcal{T}, u|_{\partial\mathcal{T}})$. That is, if $A \in \mathcal{T}$ and B_1, \dots, B_p are its children, we define

$$u_{\mathcal{T}}^*|_{A \setminus \cup_{i=1}^p B_i} = \mathcal{I}(\partial A \cup \cup_{i=1}^n \partial B_i, u|_{\partial A \cup \cup_{i=1}^n \partial B_i})$$

The function $u_{\mathcal{T}}^*$ is obtained by pasting together these functions.

Let $\tilde{\mathcal{G}}(u|_{\partial\mathcal{T}})$ denote the cost of encoding the gray values of u on $\partial\mathcal{T}$. Then we could consider the functional

$$\tilde{E}(\partial\mathcal{T}, u|_{\partial\mathcal{T}}) = \mathcal{C}(B) + \tilde{\mathcal{G}}(u|_{\partial\mathcal{T}}) + \mathcal{D}(Q(u - u_{\mathcal{T}}^*)). \quad (19)$$

As in (16), the first term in (19) corresponds to the cost of the curves. The encoding cost $\tilde{\mathcal{G}}(u|_{\partial\mathcal{T}})$ can be computed by

$$\tilde{\mathcal{G}}(u|_{\partial\mathcal{T}}) = \mathcal{E}(u|_{\partial\mathcal{T}}) \mathcal{H}^1(u|_{\partial\mathcal{T}}) \quad (20)$$

where $\mathcal{E}(u|_{\partial\mathcal{T}})$ denotes the entropy of the gray levels of u on $\partial\mathcal{T}$. Finally, the third term in (19) is the cost of the quantized interpolation error between u and $u_{\mathcal{T}}^*$. This cost is computed as in (18).

As it was proved in [10] in a continuous domain, interpolation operators satisfying the maximum principle and a set of structural and geometrical axioms can be given in terms of the solution of a degenerate elliptic partial differential equation. Between them, we would single out the Absolutely Minimizing Lipschitz Extension Model (the so-called AMLE) and the Laplacian [10]. Both are able to interpolate data given on a set of curves, and the AMLE is also able to interpolate data given on points. Moreover, as proved in [12], as a consequence of maximum principle, both methods are shape preserving in the sense that they do not create new critical levels (see [12] for a precise statement). But the computational complexity makes these operators unfeasible, if the corresponding partial differential equation is solved iteratively and we have not converged, the results are not good in terms of cost and errors with respect to the more simple method proposed here.

6.2. EXPERIMENTS USING $E(\partial\mathcal{T}, u_{\mathcal{T}})$

We display some results obtained by minimizing the functional E given by (16) which uses the mean value as interpolation operator. For the cost of the curves we tested three expressions: a) the differential chain coding cost, i.e., the expression (17) with $\mathcal{E}(\partial\mathcal{T})$ being the entropy of the distribution of differences of directions in $\partial\mathcal{T}$; b) the chain coding cost, i.e., (17) with $\mathcal{E}(\partial\mathcal{T})$ being the entropy of the distribution of directions in $\partial\mathcal{T}$; c) the cost $\mathcal{C}(\partial\mathcal{T}) = \log_2 3 \mathcal{H}^1(\partial\mathcal{T})$. The results obtained with a) and c) are quite similar and better than the ones obtained with b). The experiments displayed in Figures 12, 13, 14 and 15 correspond to the case a) but we shall comment on the results obtained with b), c). For each experiment, we shall display the selected family of level lines, denoted by $\partial\mathcal{T}$, and the image $u_{\mathcal{T}}$ which takes the mean value of u on each region of the segmentation.

Figure 12 displays the results obtained minimizing (16) with $\delta = 7$ for the image of Lena displayed in Figure 3.a. Figure 12.a displays the set of curves $\partial\mathcal{T}$ obtained, Figure 12.b displays



Figure 12. Experiment obtained minimizing (16) with $\delta = 7$. a) Left: the selected family of level lines. b) Right: the function $u_{\mathcal{T}}$.

the reconstruction $u_{\mathcal{T}}$. For this Figure and cases a), b), and c) described in the previous paragraph, the total cost $E(\partial\mathcal{T}, u_{\partial\mathcal{T}})$ in bits, the cost of curves $\mathcal{C}(\partial\mathcal{T})$ in bits, and the peak signal to noise ratio PSNR are respectively: case a) 141806.5, 54514.5, and 28.13; case b) 158570.7, 77450.6, and 29.17; case c) 138867.8, 57530.0, and 28.94.

Figure 13 displays the results obtained minimizing (16) with $\delta = 7$ for the original image displayed in Figure 4.a. Figure 13.a displays the set of curves $\partial\mathcal{T}$ obtained, Figure 13.b displays the reconstruction $u_{\mathcal{T}}$. For this Figure and cases a), b), and c) the corresponding values of the total cost, the cost of the curves and the PSNR are, respectively: case a) 157091.1, 42687.3, and 26.70; case b) 200348.8, 70656.4, and 26.29; case c) 180198.8, 61085.8, and 26.42.

Figure 14 displays the result obtained minimizing functional (16) applied to the original image displayed in Figure 5.a with $\delta = 7$. Figure 14.a displays the set of curves $\partial\mathcal{T}$ obtained. Figure 14.b displays the reconstruction $u_{\mathcal{T}}$. For this Figure and cases a), b), and c) the corresponding values of the total cost, the cost of the curves and the PSNR are, respectively: case a) 103663.9, 49370.8, and 29.74; case b) 121801.8, 67630.6, and 29.91; case c) 99326.9, 50030.5, and 30.18.

Figure 15 displays the results obtained minimizing (16) applied to the original image displayed in Figure 6.a with $\delta = 7$. Figure 15.a displays the set of curves $\partial\mathcal{T}$ obtained, Figure 15.b displays the reconstruction $u_{\mathcal{T}}$. For this Figure and cases a), b), and c) the corresponding values of the total cost, the cost of the curves and the PSNR are, respectively: case a) 67123.3, 28180.7, and 31.87; case b) 80563.8, 41459.8, and 32.72; case c) 67073.7, 28867.7, and 32.85.

REMARK 1. Let us comment on the result obtained using the AMLE interpolator (i.e. if we solve the PDE $\langle D^2u(Du), Du \rangle = 0$ on each region Γ of $\Omega \setminus \partial\mathcal{T}$ with Dirichlet boundary data u on $\partial\Gamma$). Using the AMLE as interpolation operator, minimizing (19) for Figure 13 and cases a), b), and c) described in the first paragraph of this section, the total cost $E(\partial\mathcal{T}, u|_{\partial\mathcal{T}})$ in bits, the cost of curves $\mathcal{C}(\partial\mathcal{T})$ in bits, and the peak signal to noise ratio PSNR are respectively: case a) 385689.8, 102314.7, and 21.66; case b) 390565.8, 115573.6, and 20.98; case c) 408788.3, 134369.1, and 21.70. Due to the computational complexity of the AMLE interpolation, if this algorithm has not converged, the results are not good.

The computational complexity of the minimization depends on the interpolator. In the case of functional (16) which uses the average as interpolator, the cost is the same as the cost of construction of the tree plus essentially the cost of the greedy algorithm for segmentation. This

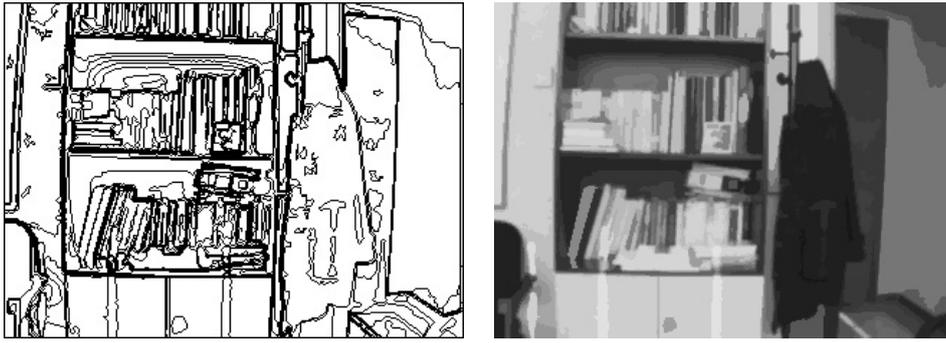


Figure 13. Experiment obtained minimizing (16) with $\delta = 7$. a) Left: the selected family of level lines. b) Right: the function u_T .

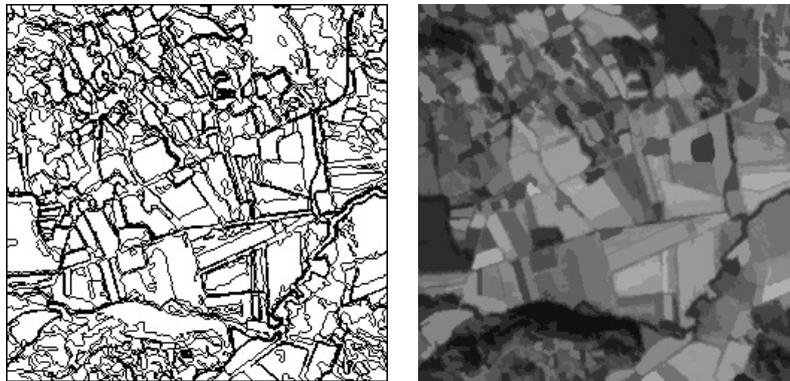


Figure 14. Experiment obtained minimizing (16) with $\delta = 7$. a) Left: the selected family of level lines. b) Right: the function u_T .



Figure 15. Experiment obtained minimizing (16) with $\delta = 7$. a) Left: the selected family of level lines. b) Right: the function u_T .

is of the order of a few seconds for a 256×256 image. Due to the computational cost, we do not recommend using a PDE based interpolation.

The subset of the level lines obtained by minimizing the energy functionals (16) is a partition of the image domain but it cannot be interpreted as a segmentation, since it is not based on a similarity criterion, or the contrast along the boundaries. On the other hand, the functional

gives us an algorithm for image encoding which permits to control the L^∞ error between the compressed image and the original one. We give some experiments in this direction.

6.3. IMAGE ENCODING AND COMPRESSION

Many algorithms exist for data compression. They can be classified into lossless and lossy. 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). In some applications, in particular compression of digital elevation models (DEM), the control of the maximum error is essential [12]. 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 coordinates in the image domain and the height z as the gray value of the image (see Fig. 16). 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 hundreds of these images, then storing them requires some compression.

The energy functional (16) permits us to select a sample $\partial\mathcal{T}$ of the level lines of the image, together with the approximation $u_{\mathcal{T}}$ given by the average values of u on the connected components of $\Omega \setminus \partial\mathcal{T}$. An example of the selected family of curves is displayed in Figure 16. The method permits a control of the error in the maximum norm. The only parameter to be given is the maximum error allowed δ . As in [12], in order to improve the performance of this algorithm we adopt a simple multiscale strategy. Given the original image v of size $N \times M$, we filter it to reduce its bandwidth and subsample it with a factor k (typically $k = 2$ and we average it in blocks of size 2×2). Let \tilde{v} be the filtered version of v , and let $u(i, j) = \tilde{v}(ki, kj)$ be its subsampled version of size $\frac{N}{k} \times \frac{M}{k}$. Then we minimize the energy functional (16) and we obtain a family of curves $\partial\mathcal{T}$, together with the approximation $u_{\mathcal{T}}$ given by the average values of u on the connected components of $\Omega \setminus \partial\mathcal{T}$. We store both $\partial\mathcal{T}$ and the values defining $u_{\mathcal{T}}$. We also store the quantized residuals $Q(u - u_{\mathcal{T}})$. We compute $u_{rec} = u_{\mathcal{T}} + Q(u - u_{\mathcal{T}})$. Then we zoom out u_{rec} to obtain an image U_{rec} , compute the residuals $Q(v - U_{rec})$, and define $v_{rec} = U_{rec} + Q(v - U_{rec})$. The image v_{rec} satisfies $\max_{(i,j)} |v(i, j) - v_{rec}(i, j)| \leq \delta$. We compress \mathcal{T} , the values $u_{\mathcal{T}}$, $Q(u - u_{\mathcal{T}})$, and $Q(v - U_{rec})$ with an arithmetic coder. As we shall explain below, we display in Table I some results obtained with this algorithm.

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. As in [12], 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 use the algorithm proposed in [22].

We shall compare the results obtained with our algorithm with the results obtained with JPEG-2000. For that, we have used the Jasper software [1] (available at <http://www.ece.ubc.ca/~mdadams/jasper>). Since, following strictly the JPEG-2000 specifications, this algorithm can not automatically control the maximum error, we will explain below in more detail how this comparison was performed. We also compare our results with ones obtained in [12]. From now on, the algorithm in [12] is denoted by ME, standing for morphological encoding, and our algorithm is denoted by MSLL, standing for minimal selection of level lines. We report results on a set of 10 DEM images of size 1200 by 1200 pixels with 16 bits per sample. The comparison

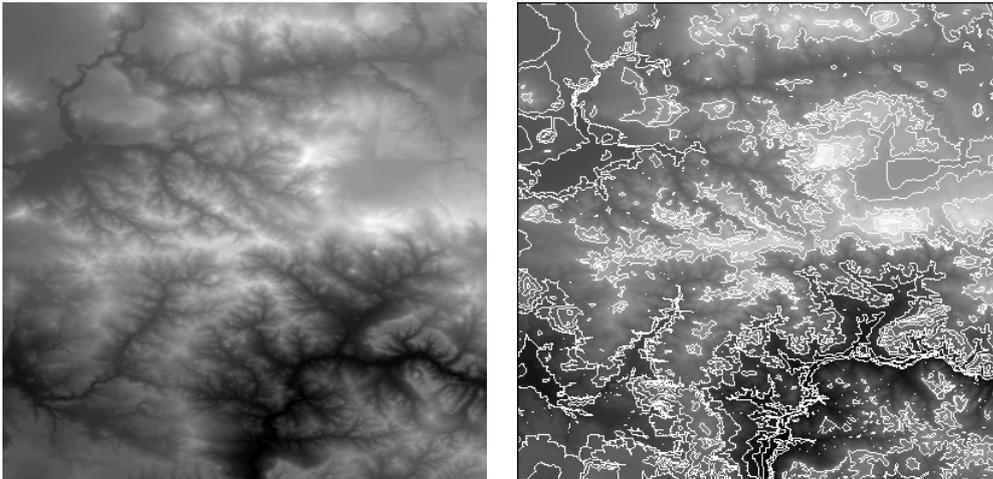


Figure 16. Left: Original DEM image example. Right: Selected level lines using our algorithm with maximum error 5.

will be done in terms of the compression ratio (abbreviated, CR) and the RMSE, the standard root mean square error. In all cases, the maximum error was fixed, and we compare the resulting RMS error. Other comparisons with JPEG-LS (which has a near lossless mode where one can impose the maximum allowed error) can be found in [45, 12].

Table I shows the compression ratio and RMSE in meters corresponding to a set of 10 images quantized with 16 bits/sample when compressed with ME and MSLL with a maximum error $e = 5$ meters. The compression ratio is given with respect to the size computed using 16 bits/sample. The maximum error being fixed, the RMS error gives another index to judge the algorithms.

Table II shows the results obtained using JPEG-2000 to compress the above set of images (column 0). For that, we have used the Jasper software [1] (available at <http://www.ece.ubc.ca/~mdadams/jasper>). The set of images is quantized with 16 bits/sample but their real range differs from image to image. We have computed it and expressed it in bits/sample (using the nearest integer by excess to cover the real range of the image). When running the Jasper software for a given image we introduced the real range in bits/sample, call it b (which for the set of images displayed is usually 11 or 12), together with the compression ratio desired. This compression ratio is equal to the one obtained with MSLL and given in the first column of Table I (since this CR is given with respect to 16 bits/sample we have to rescale it in the following way: if c represents the CR given in column 1 of Table I, we ask for the CR $c \frac{b}{16}$). The reason to specify the real range is that the quantizers used by the algorithm depend on it, and the results were much worse if we specified 16 bits/sample for each image. We may also choose to specify either the CR or the RMSE error and we have chosen to specify the CR (see below). Recall that following strictly the JPEG-2000 specifications, this algorithm can not automatically control the maximum error. Thus, if we want that the maximum error of the compressed images is 5, as in Table I, we compute the pixels of the JPEG-2000 compressed image where the error is higher than 5 and we correct their value by adding the quantized errors (with $\delta = 5$). We obtain a new compressed image with a maximum error of 5.

Table I. Compression Ratio (CR) and RMSE for MSLL and ME for a set of 10 DEM images quantized with 16 bits/sample, with a maximum error of 5 meters. The maximum error being fixed, the RMS error gives another index to judge the algorithms.

	CR (MSLL)	CR (ME)	RMSE (MSLL)	RMSE (ME)
baker-e	11.225	11.125	2.605	2.582
bend-e	11.643	11.599	2.614	2.552
bend-w	15.421	15.831	2.633	2.622
billings-e	18.557	19.284	2.437	2.541
sacramento-e	8.877	9.056	2.669	2.676
salina-e	13.296	13.713	2.613	2.593
salina-w	11.397	11.544	2.612	2.599
sandpoint-e	9.677	9.633	2.597	2.550
yakima-e	16.629	17.027	2.491	2.466
yakima-w	7.256	7.390	2.724	2.753
AVERAGE	12.398	12.620	2.599	2.593

Now, let us describe Table II. First we run Jasper software asking for the same compression ratio (as explained in the previous paragraph) as the one obtained with our algorithm MSLL which is displayed in column 1 of Table I. The compression ratio obtained may slightly differ from the one we asked for, due to the properties of the quantizer. The maximum error is larger than 5, and it is displayed in column 2. Then we correct the value of the pixels where the maximum error is greater than 5 (by adding the quantized errors) and we obtain an image whose maximum error is 5 (this is indicated in column 3). With this step we deviate from the standard JPEG-2000 specifications (that is, the complete compressed bit stream would not be in standard format due to the error control part). After these correction steps we obtain the final compression ratio displayed in column 1. Column 4 contains the final RMSE.

Figure 17 displays a selected region of a DEM image and its compressed versions using MSLL, JLS (JPEG-LS, a lossless mode of JPEG, [48]) 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 MSLL and JPEG-2000 compression while it is distorted in the case of JLS compression [48]. For other results concerning JLS compression we refer to [12].

7. Conclusion

We have discussed the interest of the Tree of Shapes of an image as a region oriented image representation. The Tree of Shapes offers a compact and structured representation of the family

Table II. Results obtained using *JPEG* – 2000 ([1]) to compress the images in column 0. For that we run the *JasPer* software giving to the code the number b of bits/sample according to the range of the image and asking for the same compression ratio as the one obtained with our algorithm (Table I) rescaled as explained in the text by the factor $b/16$. The compression ratio obtained may be different to the one we asked for, due to the properties of the quantizer. The maximum error is larger than 5 and is displayed in column 2. Then we correct the value of the pixels where the maximum error is greater than 5, by adding the quantized errors, and we obtain an image whose maximum error is 5 (this is indicated in column 3). Since the number of those pixels is small, the amount of bits required for this correction is insignificant and does not modify the CR. The final CR obtained after these corrections is displayed in Column 1 (and is referred to the size computed using 16 bits/sample). Column 4 contains the *RMSE* error after the correction. For more information, we refer to the text.

	CR (JPEG-2000)	L^∞ (no corrected)	L^∞ (corrected)	RMSE
baker-e	12.881	18	5	1.947
bend-e	13.453	20	5	1.967
bend-w	18.487	18	5	1.652
billings-e	23.128	16	5	1.559
sacramento-e	9.851	19	5	2.087
salina-e	15.974	20	5	1.787
salina-w	13.139	18	5	1.920
sandpoint-e	11.110	19	5	2.054
yakima-e	20.75	20	5	1.671
yakima-w	8.408	23	5	2.357
AVERAGE	14.719	19.1	5	1.900

of level lines of an image [29, 31]. We have shown how this representation can be used for segmentation, rate distortion optimization, and encoding. For segmentation purposes, we have minimized the simplified Mumford-Shah energy functional subordinated to the topographic map of the image, displaying some experiments. The functional was minimized on a hierarchy of partitions of the image domain constructed using a greedy algorithm to compute local minima of the simplified Mumford-Shah functional at several scales. The result corresponds to a minimum of the functional on the hierarchy of partitions. The rate distortion problem is also solved in this hierarchy of partitions.

We have formulated a version of the Minimum Description Length principle subordinated to the topographic map of the image which gives us a structured encoding of the image. The associated energy functional represents the cost in bits of encoding the selected level lines while

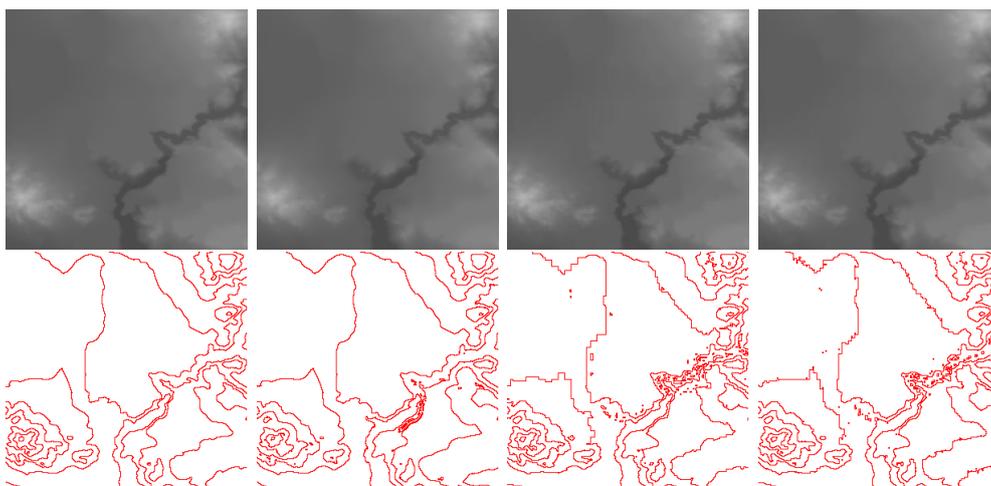


Figure 17. Top row: From left to right selected region of the original DEM image *bend-e*, the same region after compressing with MSL with sup error 5, after compressing with JLS with sup error 5 and after compressing with JPEG-2000 with sup error 5. Bottom row: Level lines of the above images.

controlling the maximum error of the reconstructed image. In this case, a greedy algorithm is used to minimize the corresponding functional and some experiments are displayed. In case of digital elevation models, we have tested the selected family of level lines as an encoding of the image, and we have displayed the corresponding compression results and compared them to JPEG-2000.

Acknowledgements

The first and second authors acknowledge partial support by the Departament d'Universitats, Recerca i Societat de la Informació de la Generalitat de Catalunya and by PNP GC project, reference BFM2003-02125. L. Igual acknowledges support by the French Space Agency (CNES) and the company THALES (France).

References

1. M. D. Adams, "The jpeg-2000 still image compression standard". Available at <http://www.ece.ubc.ca/~mdadams>, 2001.
2. C. Ballester, E. Cubero-Castan, M. Gonzalez and J.M. Morel, 'Image intersection and applications to satellite imaging'. Preprint, C.M.L.A., Ecole Normale Supérieure de Cachan, 1998
3. C. Ballester, V. Caselles, and P. Monasse, "The tree of shapes of an image". *ESAIM: Control, Opt. and Calc. of Variations* **9**, pp. 1–18, 2003.
4. T. Berger, *Rate distortion theory: A mathematical basis for data compression*. Prentice Hall, 1971.
5. L. Breiman, J.H. Friedman, R.A. Olsen, and C.J. Stone, *Classification and regression trees*. Wadsworth, Belmont, California, 1984.
6. F. Cao, P. Musé, and F. Sur, "Extracting meaningful curves in images". To appear in *Journal of Mathematical Imaging and Vision*, 2005.
7. F. Cao, Y. Gousseau, J.M. Morel, P. Musé, and F. Sur, "An a contrario decision method for shape element recognition". Preprint CMLA No. 16, 2004.

8. V. Caselles, B. Coll, and J.M. Morel, "Topographic maps and local contrast changes in natural images". *Int. J. Comp. Vision* **33**, pp. 5–27, 1999.
9. V. Caselles, J.L. Lisani, J.M. Morel and G. Sapiro, "Shape Preserving Histogram Modification". *IEEE Transactions on Image Processing* **8**(2), pp. 220–230, 1999.
10. V. Caselles, J.M. Morel, and C. Sbert, "An axiomatic approach to image interpolation". *IEEE Transactions on Image Processing* **7**, pp. 376–386, 1998.
11. V. Caselles, and P. Monasse, "Morse Theories of the Topographic Map". *In preparation*.
12. V. Caselles, G. Sapiro, A. Solé, and C. Ballester, "Morse Description and Morphological Encoding of Continuous Data". *SIAM Journal on Multiscale Modeling and Simulation* **2**, pp. 179–209, 2004.
13. R. Chiariglioni, "MPEG and Multimedia Communications". *IEEE Transactions on Circuits and Systems for Video Technology* **7**, pp. 5-18, 1997.
14. P.A. Chou, T. Lookabaugh, and R.M. Gray, "Optimal pruning with applications to tree-structured source coding and modeling". *IEEE Trans. Inform. Theory* **35**, pp. 299–315, 1989.
15. A. Desolneux, L. Moisan, and J.M. Morel, "Edge detection by Helmholtz principle". *Journal of Mathematical Imaging and Vision* **14**, pp. 271–284, 2001.
16. M. Eden, and M. Kocher, "On the performance of a contour coding algorithm in the context of image coding. Part I: Contour segment coding". *Signal Processing* **8**, pp. 381–386, 1985.
17. J. Froment, "A functional analysis model for natural images permitting structured compression". *ESAIM: Control, Optimization and Calculus of Variation* **4**, pp. 473–495, 1999.
18. L. Garrido, "Hierarchical Region Based Processing of Images and Video Sequences: Application to Filtering, Sementation and Information Retrieval". PhD thesis, Universitat Politècnica de Catalunya, 2002.
19. F. Guichard and J.M. Morel, *Image Iterative Smoothing and P.D.E.'s*. Book in preparation, 2000.
20. L. Guigues, "Modèles Multi-Échelles pour la Segmentation d'Images". PhD thesis, Université de Cergy-Pontoise, 2003.
21. A. Gersho and R.M. Gray, *Vector quantization and signal compression*. Kluwer Academic Publishers, 1992.
22. J. M. Hyman, "Accurate Monotonicity preserving cubic interpolation". *SIAM J. Sci. Stat. Comp.* **4**, pp. 645-654, 1983.
23. L. Igual, L. Garrido, and V. Caselles, "A Contrast Invariant Approach to Motion Estimation. Validation and Motion Segmentation".
24. G. Koepfler, C. Lopez and J.M. Morel, "A multiscale algorithm for image segmentation by variational method". *SIAM J. Numer. Anal.* **31**, pp. 282–299, 1994.
25. C. Kuratowski, *Topologie I, II*. Editions J. Gabay: Paris, 1992.
26. Y. G. Leclerc, "Constructing Simple Stable Descriptions for Image Partitioning". *International Journal of Computer Vision* **3**, pp. 73–102, 1989.
27. F. Meyer, "The dynamics of minima and contours". In: *Mathematical morphology and its applications to image processing*. Atlanta (GA), USA, pp. 329–336, 1996.
28. F. Meyer and S. Beucher, "Morphological Segmentation". *J. of Visual Communication and Image Representation* **1**, pp. 21–46, 1990.
29. P. Monasse, "Représentation morphologique d'images numériques et application au recalage". PhD thesis, Université de Paris-Dauphine, 2000.
30. P. Monasse, "Contrast invariant image registration". In: *Proc. of International Conference on Acoustics, Speech and Signal Processing* **6**, pp. 3221–3224, 1999.
31. P. Monasse and F. Guichard, "Fast computation of a contrast invariant image representation". *IEEE Transactions on Image Processing* **9**, pp. 860–872, 2000.
32. J.M. Morel and S. Solimini, *Variational methods in image processing*. Birkhäuser Verlag: Basel, 1994.
33. D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and variational problems". *Communications on Pure and Applied Mathematics* **42**(5), pp. 577–685, 1988.
34. J. Rissanen, "Minimum-Description-Length Principle". In: *Encyclopedia of Statistical Sciences* **5**, pp. 523–527, 1987. John Wiley: New York.
35. P. Salembier, "Morphological multiscale segmentation for image coding". *IEEE Transactions on Signal Processing* **38**(3), pp. 359–386, 1994.
36. P. Salembier, P. Brigger, J.R. Casas and M. Pardàs, "Morphological Operators for Image and Video Compression". *IEEE Transactions on Image Processing* **5**, pp. 881–897, 1996.
37. P. Salembier and L. Garrido, "Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval". *IEEE Transactions on Image Processing* **9**(4), pp. 561–576, 2000.

38. P. Salembier, F. Marqués, M. Pardás, R. Morros, I. Corset, S. Jeannin, L. Bouchard, F. Meyer, B. Marcotegui, “Segmentation-based video coding system allowing the manipulation of objects”. *IEEE Transactions on Circuits and Systems for Video Technology* **7**, pp. 60-73, 1997.
39. P. Salembier and J. Serra, “Flat zones filtering, connected operators and filters by reconstruction”. *IEEE Transactions Image Processing* **4**(8), pp. 1153-1160, 1995.
40. G.M. Schuster and A.K. Katsaggelos, *Rate-distortion based video compression*. Kluwer Academic Publishers, 1997.
41. Y. Shoham and A. Gersho, *Efficient Bit Allocation for an Arbitrary Set of Quantizers*. *IEEE Trans. on Acoust., Speech, Signal Processing* **36**(9), pp. 1445-1453, 1988.
42. J. Serra, *Image analysis and mathematical morphology*. Academic Press: New York, 1982.
43. T. Sikora, “The MPEG-7 visual standard for content description - an overview”. *IEEE Transactions on Circuits and Systems for Video Technology* **11**, pp. 696-702, 2001.
44. P. Soille, *Morphological Image Analysis*. Springer Verlag, 2003.
45. A. Solé, “Geometric Image Coding, Filtering and Restoration”. PhD thesis, Universitat Pompeu Fabra, 2002.
46. L. Vincent, “Morphological Area Openings and Closings for Grey-scale Images”. In: *Proceedings of the Workshop Shape in Picture: Mathematical Description of Shape in Gray-Level Images*. Driebergen, The Netherlands, pp. 197–208, 1994.
47. L. Vincent and P. Soille, “Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations”. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**(6), pp. 583–598, 1991.
48. M.J. Weinberger, G. Seroussi, and G. Sapiro, “The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS”. *IEEE Transactions on Image Processing* **9**(8), pp. 1309–1324, 2000.
49. M. Wertheimer, “Untersuchungen zur Lehre der Gestalt, II”. *Psychologische Forschung* **4**, pp. 301–350, 1923.