DOI 10.1007/s10851-008-0118-x

3D Edge Detection by Selection of Level Surface Patches

Enric Meinhardt · Ernesto Zacur · Alejandro F. Frangi · Vicent Caselles

Published online: 17 October 2008

Abstract We propose a new edge detector for 3D grayscale images, extending the 2D edge detector of Desolneux et al. (J. Math. Imaging Vis. 14(3):271–284, 2001). While the edges of a planar image are pieces of curve, the edges of a volumetric image are pieces of surface, which are more delicate to manage. The proposed edge detector works by selecting those pieces of level surface which are well-contrasted according to a statistical test, called Helmholtz principle. As it is infeasible to treat all the possible pieces of each level surface, we restrict the search to the regions that result of optimizing the Mumford-Shah functional of the gradient over the surface, throughout all scales. We assert that this selection device results in a good edge detector for a wide class of images, including several types of medical images from X-ray computed tomography and magnetic resonance.

E. Meinhardt (✉) · V. Caselles
Department of Information and Communication Technologies,
Universitat Pompeu Fabra, Passeig de Circumval·lació 8,
08003 Barcelona, Spain
e-mail: enric.meinhardt@upf.edu

E. Zacur
GTC, Aragon Institute for Engineering Research (I3A),
Universidad de Zaragoza, María de Luna 3, 50018 Zaragoza,
Spain

A.F. Frangi
Center for Computational Imaging & Simulation Technologies in
Biomedicine, Department of Information and Communication
Technologies, Universitat Pompeu Fabra, Barcelona, Spain

A.F. Frangi
Networking Research Center on Bioengineering, Biomaterials
and Nanomedicine (CIBER-BBN), Barcelona, Spain

Keywords Edge detection · Image segmentation · Level surfaces · Digital topology

1 Introduction

Edge detection is the task of finding the boundaries between the objects that appear in a digital image. Segmentation is a different, but closely related problem, which consists in finding the objects themselves. Both problems have different constraints and applications. From a mathematical standpoint, edge detection finds the discontinuities of a function and segmentation finds a partition of the domain. Edge detection, being of a lower-level nature than segmentation, is aimed at picking structures all over the image and usually needs no initialization. People have been using 2D edge detection for years (see [33]) for many tasks. For example, to obtain a visually appealing "primal sketch" [29, 42] of a picture, to reduce the amount of information present in an image, to get a manageable list of "features" to perform registration [3, 8, 31] of two images, or shape matching [38, 44]; and finally as a first step towards the segmentation of the image into regions. For an account of its applications to computer vision see [24].

While traditional edge detection was initially introduced for 2D images, most of the techniques can be extended to 3D images. However in three-dimensional images, the boundaries between objects are not curves in the plane but surfaces in space. In that setting, the summarizing property of edges is even more important because these images can not be easily visualized as a whole (without resorting to specialized rendering techniques). On the contrary, a set of surfaces in space is easy to visualize, specially if the user can rotate interactively the whole image domain. Even when the surfaces are nested it is useful, because the surfaces can be endowed with transparency. Thus, edge detectors are an invaluable tool in 3D visualization, for they provide an efficient way to glance through the content of whole images. Aside from visualization, 3D edges are used also for other tasks, e.g. registration [32] or landmarking [49].

Let us briefly review the main approaches to edge detection. A gray level image can be realistically modeled as a real-valued function $u(\mathbf{x})$ where $\mathbf{x}$ represents an arbitrary point of a rectangle Ω in $\mathbb{R}^N$ ($N = 2$ for usual pictures, 3 for medical images and movies for example) and $u(\mathbf{x})$ denotes the gray level at $\mathbf{x}$. In this continuous setting, when an image $u : \Omega \to \mathbb{R}$ is a smooth function, edges are usually defined in terms of a differential operator. Most, if not all, of them are based on one of the following three:

The *norm of the gradient* $|Du| = \sqrt{u_x^2 + u_y^2}$ produces an image which is interpreted as a measure of the "edgeness" at each point of the image domain. Many detectors (e.g., Sobel [18], Prewitt [50], Roberts [51], Kirsch [36] and the morphological gradient [6]) can be interpreted as numerical schemes to approximate this norm. The main advantage of these operators is that they are fast and easy to compute. Their main limitations are that their output is difficult to use and blurry edges are not well localized.

The *Laplacian*. According to Marr-Hildreth [42], edges can be defined as zero-crossings of the Laplacian $\Delta u = u_{xx} + u_{yy}$. This method has the advantage of being able to directly produce curves which are well-localized, but it may lead to false detections (e.g., at almost flat zones where noise dominates).

Canny's operator $D^2u(Du, Du) = u_x^2 u_{xx} + 2u_x u_y u_{xy} + u_y^2 u_{yy}$ is the second derivative of u along its gradient lines. Its zero-crossings are called Haralick's edges [27], and they are better localized than the zero-crossings of the Laplacian [28].

Since Canny's operator is the second derivative of u along its gradient lines, it vanishes where the first derivative of u is maximal in the direction of the gradient. Instead of computing second derivatives, these maxima can be found directly by looking at the values of the derivative at the neighboring pixels in the gradient direction and discarding those pixels that have higher contrasted neighbors. This process, called non-maximum suppression, forms the basis of an efficient implementation of Canny's filter. The explicit details of the method (see [19] and [9]) are somewhat intricate because the choice of previous filtering is critical to ensure the best localization. The result of this non-maximum suppression is then pruned using two threshold parameters in a process called *hysteresis*. Thus, Canny's edge detector, while being able to give very good results, uses three parameters which are usually set by hand, specifically for each image, by visual inspection: The first parameter is the width of the (necessary) initial linear filtering, whose optimal value depends on the overall amount of noise in the original image, and the other two parameters are thresholds, whose optimal value depends on the distribution of the contrast.

Edge detectors usually give as edges a set of pixels, and those have to be connected to produce a set of curves. Active contours or snakes were developed to obtain a boundary segmenting a region of the image (or a set of regions) [13, 16, 34, 35, 41]. They are interfaces (curves in 2D images, surfaces in 3D images) that evolve to minimize an energy functional. The minimization is usually performed using gradient descent starting from a given initialization. The choice of a good initialization is thus critical because the energy functional may have several local minima. Other particular approaches, based on segmentations, include finding a partition of the image domain that globally minimizes an energy, as in *graph cuts* [7]; or finding *watersheds* (i.e., connected components of lower level sets) of the gradient norm [53, 55].

Desolneux, Moisan and Morel proposed in [20] a new method for edge detection (named DMM, from now on), based on a general theory (see [21, 26, 40]) aimed at giving sensible values to perceptual thresholds. It is possible to apply that theory directly to set the hysteresis thresholds for Canny's filter, but DMM is more elaborate in that it finds a separate threshold for each edge, according to its size and its contrast. The main steps of this method are the following:

(i) The family of level lines of the image and the distribution of the modulus of the gradient are computed and stored.
(ii) Then, all arcs of level lines are subsequently tested, one by one, to verify that they are well contrasted. The arcs that pass the statistical test are the output of the algorithm (they are named *meaningful* edges).

This algorithm contains no tunable parameters because the minimum contrast required for a given curve to be meaningful is determined automatically by the statistics of the image contrast. With respect to Canny's detector, it has the advantage of not producing an unstructured set of edge points, but a set of continuous planar curves on the image domain (e.g., with sub-pixel precision). Let us mention that, in order to refine the computation of the boundary of an object, a meaningful level line may be used as an initialization for a classical active contour model [23]. It is worth noting that [20] introduced two variants of this algorithm, testing either whole level curves or arcs of them, and the respective outputs were named *meaningful boundaries* and *meaningful edges*. Our method is based on the second, more general, variant.

Our purpose is to follow the main steps of DMM algorithm in order to construct an edge detector for 3D images. Thus, the first step of our edge detector will be to compute the family of its level surfaces. For that, we require the use of an image representation (and its associated data structure) that enables us to compute and store them efficiently for later manipulation. Data structures are commonly used in image processing, each of them being adapted to a class of image operations (e.g. the Fourier transform is well adapted to the application of linear filters). Level surfaces are defined as the boundaries of the connected components of the (upper and lower) level sets of the image. They give a complete and non-redundant representation of the image which is contrast independent. The family of level surfaces of the image is organized in a data structure called the "tree of shapes" [14] of the image which extends to the 3D case the tree of shapes defined for 2D images in [46] and [45]. This structure merges into a single tree the information contained in the trees of connected components of upper and lower level sets and is well adapted to compute morphological operations on the image. 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. The theoretical foundation of this data structure is described in detail in [4] and [14], and its practical implementation for the 2D case in [45] and independently in [54], under the name of *monotonic tree*. Other closely related data structures are those developed by Cox-Karron-Ferdous [17], by Pascucci-Cole-McLaughlin [48], by Carr-Snoeyink-Axen [12], and by Sarioz-Kong-Herman [52]. The idea of using trees to encode the complexity of 3D structures dates back to the origins of image processing [15].

The proposed edge detector takes a three-dimensional image as input and produces a set of surfaces as output. The produced surfaces can be defined in the following way: first we build a list of all the possible connected patches of all the level surfaces of the gray scale image. Then we run a statistical test on each of these patches to decide whether it is well-contrasted or not. The output of the algorithm is the set of patches that pass the test. The method has a single parameter ε whose meaning is the sensitivity of the edge detector, defined as the number of bad edges that are expected to be produced. By "bad edges" we mean edges whose distribution of gradients can be explained by a model of noise, in a very precise sense described below. It is customary to set $\varepsilon = 1$, so that at most one output surface can be expected to be a result of noise. This paper explains how to implement such a selection process efficiently, and what kind of pre and post-processing can help to make the algorithm run much faster and produce better output. As most edge detectors, our method is based on the idea that edges occur at sharp intensity changes (or discontinuities) of the image intensity.

Our method relies on the following assumption: in an ideal case (e.g., a perfect acquisition method giving infinite resolution images without noise) the boundaries of the objects could be obtained by thresholding the image intensity.

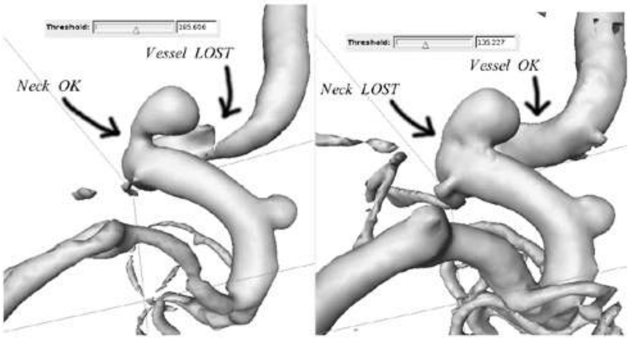


Fig. 1 Two different isosurfaces of the same medical image. Note that each choice of threshold segments well some part of the image, but no threshold gives a globally correct segmentation

This assumption holds for a wide class of real world images, like many medical images, namely, X-ray computed tomographies and magnetic resonances, where the acquisition apparatus measures the density of a physical or chemical property of objects in space. In practice, however, *a single threshold does not suffice*, because there are artifacts due to the reconstruction of the image, and the limitations inherent to its finite representation. For example, thin vessels having the same width as a voxel appear in the images much darker than the interior of large vessels, even if the contrast agent concentration or the measured property is the same in both places. See Fig. 1 for an illustration of this fact. Note that not all 3D images satisfy this assumption. For example, in ultra-sound images the objects are defined mostly by textures, and for these the proposed method will likely not give good results (nor the existing methods described above).

This paper is organized as follows. In Sect. 2 we recall the basic ideas related to the construction of the tree of shapes of an image. In Sect. 3 we describe a general statistical test which we use to decide which subsets (from within a given family) of an image are well-contrasted. In Sect. 4 we explain how to produce a family of subsets to run the previous statistical test on. Together, both sections contain the description of the proposed method. Section 5 describes a simple pre and post-processing of the data and results which improves the performance of the proposed method. In Sect. 6 we illustrate our method with a set of synthetic and real data. Finally, we summarize our conclusions in Sect. 7.

2 Preliminaries

The purpose of this section is to give a brief description of the family of level surfaces of a 3D image and its organization as a tree of shapes [4, 45]. Since we are considering 3D images, let us consider a gray scale image $u : \Omega \to \mathbb{R}$ where the image domain Ω is a closed rectangle in $\mathbb{R}^3$. The upper and lower level sets of the image u are defined respectively as

$$[u \geq \lambda] = \{\mathbf{x} \in \Omega \ : \ u(\mathbf{x}) \geq \lambda\}$$

$$[u < \lambda] = \{\mathbf{x} \in \Omega \ : \ u(\mathbf{x}) < \lambda\}$$

for $\lambda \in \mathbb{R}$. We assume that $u : \Omega \to \mathbb{R}$ 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. This allows to give a short description of the main objects contained in the tree of shapes [4, 45]. Moreover it covers the case of discrete supported images. Indeed, we may always transform a discrete supported image $u(i, j, k)$, $(i, j, k) \in \{1, \dots, N\}^3$ into an upper semi-continuous function in $\Omega = [0, N]^3$ by defining $u(x_1, x_2, x_3) = u(i, j, k)$ when $(x_1, x_2, x_3) \in (i-1, i) \times (j-1, j) \times (k-1, k)$, and taking at a common boundary element (be either, face, edge or vertex) the highest value of the neighboring points.

Given a point $\mathbf{x} \in \Omega$ and a set $A \subseteq \Omega$, we denote by $cc(A, \mathbf{x})$ the connected component of A that contains $\mathbf{x}$, or the empty set if $\mathbf{x} \notin A$.

Heuristically, the tree of shapes of a 3D image, u, is a data structure which encodes in a tree the family of its level surfaces. To be able to handle discontinuous functions, more specifically, upper semi-continuous ones, we define *level surfaces* as the external boundaries of the connected components of the level sets of the image. This leads us to the notion of shape, which consists in filling-in the holes of the connected components of the upper and lower level sets of u. The operation of hole filling was called saturation in [4, 45]. Thus, level surfaces are the boundaries of shapes and giving them is equivalent to give the family of shapes. Notice that it is easy to imagine them when the image is smooth.

Definition 1 Let $A \subset \Omega$. We call holes of A in Ω the components of $\Omega \backslash A$. Let $p_\infty \in \Omega \backslash A$ be a reference point, and let T be the hole of A in Ω containing p_∞. We define the saturation of A with respect to p_∞ as the set $\Omega \backslash T$ and we denote it by $\mathrm{Sat}(A, p_\infty)$. 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 $\mathrm{Sat}(A, p_\infty) = \Omega$. Note that $\mathrm{Sat}(A, p_\infty)$ is the union of A and its internal holes.

Definition 2 Given an image $u : \Omega \to \mathbb{R}$ and $p_\infty \in \Omega$, we call shapes of inferior (resp. superior) type the sets

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

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

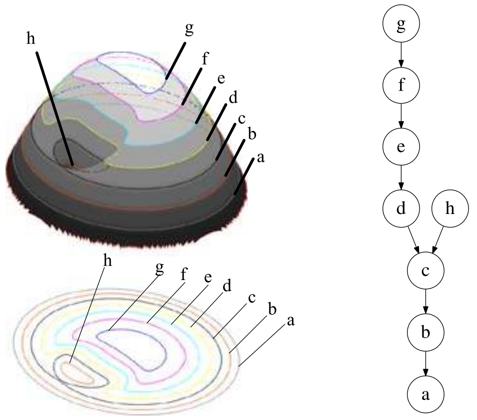


Fig. 2 An example of a tree of shapes. *Left*: the graph of a 2D function with its corresponding shapes. *Right*: the tree of shapes corresponding to the figure on the *left*, the nodes of the tree are numbered following a post-order traversal

The reference point p_∞ acts as a point at infinity and can be fixed anywhere without affecting the description of the tree of the level surfaces encoded in the tree of shapes [4, 45]. In practice, we have taken p_∞ as the origin of coordinates, which lies at corner of the image. Observe that since Ω is a rectangle the boundary of any shape of an image is connected. Moreover, as it is proved in [4, 45], if a shape S is closed, then $S = \mathrm{Sat}(\partial S, p_\infty)$. This is the mathematical translation of the fact that a shape is essentially equivalent to its boundary, the level surface.

The main result of this construction, proved in [4], says that *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. If the 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 root of the tree is Ω, 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 the sets B_1 and B_2 must be nested, since they are not disjoint. See Fig. 2 for the tree of shapes of a simple 2D image.

Let us note that, for discrete images, the number of shapes is at most equal to the number of voxels; thus it can be stored efficiently in a tree data structure using a space proportional to the original image.

3 Well-Contrasted Subsets of an Image: An A-contrario Approach

As we explained in the introduction, the proposed edge detector consists of two steps: first we produce a family of candidate surfaces and then we select the most contrasted ones (if any) among this family. In this section we give a general

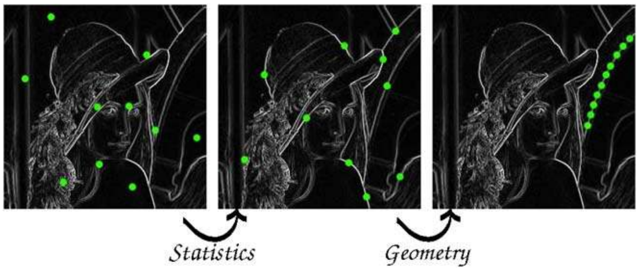


Fig. 3 *Well-contrasted level curves of an image*. These figures display three different sets of 10 points thrown in the domain of the Lena image. *Left*: points thrown randomly. *Middle*: points thrown randomly in places where the gradient is high. *Right*: points on a well-contrasted level curve. The first subset will fail the test described on Sect. 3, and the other two subsets will pass it. However, only the third subset will be presented to the test by the method described on Sect. 4

definition of well-contrasted subsets of an image according to a statistical test. In Sect. 4 we construct a family of surface patches to which we may apply the contrast test. The two sections together form the core of the method.

Let us define the concept of well-contrasted subset of an image. The definition is a slight generalization of the one given in [20, 21] based on an *a-contrario* model: Knowing the distribution function for the image contrast, we would sample the contrast values of the image at a randomly selected set of points, and we would look whether the sample had a distribution with exceptionally high contrast. In that case, this set of samples would be accepted as a well-contrasted set. Numerically, this reduces to selecting the sets which are large and whose minimum contrast is high. For a thorough discussion of the statistical foundation of this method, see [26], where it was described using the vivid name "conspiracy of random". For the intuitive idea in our case, see Fig. 3. Notice that this *a-contrario* model is not based on an image of noise, but on noisy curves over the original image.

The norm of the gradient defines a contrast for every point on the image domain. We regard the values of the contrast at each voxel as *independent and identically distributed random variables, X_i, whose distribution is given by the histogram of the contrast*. This notation will be used throughout this section. This is a good model when a few voxels are chosen randomly over the image domain, but it fails when the voxels are not chosen independently (for instance, if they are specially chosen along the boundary of an object). This failure is precisely what we look for, as the method can be regarded as an hypothesis testing of the independence assumption.

In the following paragraphs we describe a general setting to detect whether a sample from a distribution has abnormally large values. This device can be used to detect exceptionally well-contrasted subsets of an image. *The proposed edge detector is a particular case of this when the subsets are the level surfaces of the image* (*or their connected parts*).

To measure the contrast of sets of points we use an arbitrary statistic f, which for now is a parameter of the method:

$$\mathrm{contrast}(\{X_i\}) := f(X_1, \ldots, X_n).$$

This statistic serves to summarize the whole contrast distribution of the set of points into a single real number. It may help to think that f is increasing in each of its components, but this is not logically needed for the following propositions to hold. Possible choices of f are the minimum $X_{(1)}$, the mean value $n^{-1}\sum X_i$, the median $X_{(n/2)}$, or some other quantile $X_{(\frac{q}{100}n)}$. From the statistic f we need its distribution functions F_n:

$$F_n(\mu) := \mathbb{P}(f(X_1, \ldots, X_n) \geq \mu)$$

which are decreasing functions of a real variable with values in the $[0, 1]$ interval. Notice that these are the complement of the usual cumulative distribution function of $f(X_i)$. We also define for every positive integer n a meaningfulness function $\mathcal{M}_n$ as

$$\mathcal{M}_n(x_1, \ldots, x_n; \varepsilon, N) := \log \varepsilon - \log N - \log F_n(f(x_1, \ldots, x_n))$$

which is a real-valued function of n real variables (and two real parameters ε, N). In the following, when the subindex n of both F and $\mathcal{M}$ can be deduced from context, it will be omitted.

Now we define our statistical test. Suppose that we are going to deal with N sets of samples of the contrast:

$$S_i = \{X_1^i, \ldots, X_{n_i}^i\} \quad i = 1, \ldots, N$$

Definition 3 We define the *meaningfulness* of the set S_i as $\mathcal{M}(X_1^i, \ldots, X_{n_i}^i; \varepsilon, N)$. We say that the set S_i is *meaningful* when its meaningfulness is positive. If we want to emphasize the parameters ε and f we will talk about ε-meaningfulness of S_i in the f-sense, or of whether the set S_i is ε-meaningful in the f-sense.

Notice that the set S_i is meaningful when

$$N \cdot F_n(f(X_1^i, \ldots, X_{n_i}^i)) < \varepsilon$$

and this is the usual definition of "meaningfulness" given in [20, 21]. The quantity on the left hand side of the previous inequality is then called the Number of False Alarms of the set S_i.

Definition 3 is justified by the following proposition.

Proposition 1 *Under the same statistical model as Definition 3, the expectation of the number of ε-meaningful sets is smaller than ε.*

The proof of the proposition is an easy consequence of this elementary result.

Lemma 1 *Let Y be a random variable and let G be its distribution function $G(y) = \mathbb{P}(Y \geq y)$. Then, for $t \in [0, 1]$*

$$\mathbb{P}(G(Y) < t) \leq t$$

Proof of the Proposition 1 Let V be the random variable that counts the number of ε-meaningful sets. Notice that V is a function of the variables X^i_j. We want to prove that $\mathbb{E}(V) \leq \varepsilon$. Let V_i be the random variable that equals 1 if the set S_i is ε-meaningful and 0 otherwise, thus

$$\mathbb{E}(V) = \mathbb{E}(V_1) + \cdots + \mathbb{E}(V_N)$$

Now we have

$$\begin{aligned} \mathbb{E}(V_i) &= \mathbb{P}(V_i = 1) = \mathbb{P}\big(N \cdot F(f(X^i_1, \ldots, X^i_{n_i})) < \varepsilon\big) \\ &= \mathbb{P}\left(F(f(X^i_1, \ldots, X^i_{n_i})) < \frac{\varepsilon}{N}\right) \leq \frac{\varepsilon}{N} \end{aligned}$$

where the last step is the application of the lemma to the random variable $Y = f(X^i_1, \ldots, X^i_{n_i})$, whose distribution function if $G(y) = F(y)$. Substituting this result in the previous formula we get

$$\mathbb{E}(V) = \frac{\varepsilon}{N} + \cdots + \frac{\varepsilon}{N} = \varepsilon$$

The above proof is adapted from the proof given in [10]. □

The original definition of meaningfulness given in [20] used the statistic $f = \min$. There is a reason to allow for different choices of f, that can give more robust detectors; see for example Sect. 5.1 where it is used with a statistic other than $f = \min$. In the case of the minimum, the function F can be obtained directly from the distribution of the contrast, which is approximated using its histogram:

$$H(\mu) = \mathbb{P}(X \geq \mu) := \frac{\text{number of voxels with } |Du| \geq \mu}{\text{total number of voxels}}$$

Then we have

$$F_n(\mu) = H(\mu)^n$$

where n is the number of points in the subset (the number of arguments of the function f). This minimum is a special case of the quantiles:

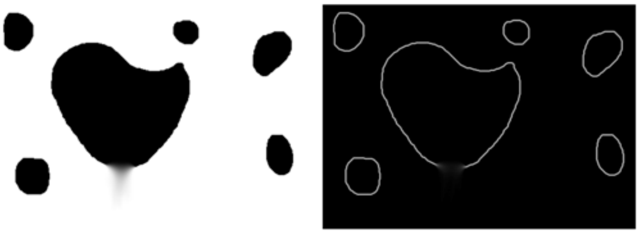


Fig. 4 An image where level lines selection using $f = \min$ fails to detect a boundary which is visually obvious. *Left*: the image. *Right*: its contrast

Proposition 2 (Distribution of quantiles) *Let $X_1, \ldots, X_n$ be independent and identically distributed random variables with distribution function $H(\mu) = P(X_1 \geq \mu)$ and let $X_{(1)}, \ldots, X_{(n)}$ be the outcomes of these variables ordered increasingly. Then $X_{(k)}$ is a random variable whose distribution function is given by a binomial tail of parameter $H(\mu)$:*

$$P(X_{(k)} \geq \mu) = \sum_{i=0}^{k-1} \binom{n}{i} (1 - H(\mu))^i H(\mu)^{n-i} \tag{1}$$

in particular, we have the distribution of the minimum computed before: $P(X_{(1)} \geq \mu) = H(\mu)^n$.

The right hand side of (1) can be written in terms of the incomplete beta function (see [1]) as $I_{H(\mu)}(n - k + 1, k)$. We used the GSL library [25] which provides a function call for it.

Remark 1 In allowing for a choice of statistic f we are motivated by the fact that the minimum contrast is not a robust descriptor: if one single point of the set has very low contrast, the whole set is discarded regardless of the contrast of all other points. We refer to Fig. 4 for a synthetic image illustrating this phenomenon. In that figure we have a well-contrasted object, surrounded by level curves whose gradient is maximum at almost the totality of their points. However, all the curves cross a blurred region of the image, where the gradient can be made to be arbitrarily low. Then, none of these curves will be detected as meaningful. There are two approaches to deal with this kind of problem: either we work with parts of level curves instead of whole level curves, or we choose a different statistic such $f = 10$th quantile. The first approach is the one chosen in this paper, but the use of a robust statistic f is much faster and, in some cases, gives similar results.

4 Computing Meaningful Patches of Level Surfaces

This section treats the main difference between the 2D and 3D versions of the edge detector.

4.1 Hierarchies of Partitions

The objects of our study are parts of level surfaces. This is the place where our method differs from the 2D case: a connected subset of a surface can be much more convoluted than a connected subset of a curve (which is determined by its two endpoints). This means that we can not treat all the connected subsets of each level surface, as is done in 2D: the search space would be too large. The first aid in the reduction of this search space comes from the observation that we are not really interested in *all* the subsets of a surface that pass the ε-meaningfulness test, but only in those that are "maximal" in the following sense:

Definition 4 ([20]) Let $\mathcal{S}$ be a level surface of the image. A connected subset $S \subseteq \mathcal{S}$ is *maximal meaningful* when it is meaningful and

- it does not contain a strictly more meaningful connected subset,
- it is not contained in a more meaningful connected subset.

Proposition 3 ([20]) *Maximal meaningful subsets in the* min-*sense are disjoint inside its level surface.*

Definition 4 was given in [20] when the set $\mathcal{S}$ is a level curve and S an edge curve, that is, a connected subset of S_i. In that case the NFA is given by the function

$$F(\mu, l) = N \cdot H(\mu)^l \tag{2}$$

where N is the number of edge curves of the image. Since edge curves are connected subsets of level curves, N can be computed for a given image. Definition 4 is analogous to the one given in [20]. For the time being, we assume that N is a constant that can be computed. Then, the proof of Proposition 3 is the same as in [20] and is based on the observation that, for a fixed value of μ, the function $F(\mu, l)$ is nondecreasing in l.

Proposition 4 *Let $\mathcal{S}$ be a level surface of the digital image u. Maximal meaningful subsets of $\mathcal{S}$ are connected components of upper level sets of the modulus of the gradient restricted to $\mathcal{S}$.*

Observe that there may not be any meaningful connected subset of $\mathcal{S}$, in which case the proposition is vacuously true. In case there is one, then the connected subset with the smallest NFA is maximal meaningful.

Proof Let S be a maximal meaningful subset of $\mathcal{S}$, and let $\mu = \min_{x \in S} |\nabla u(x)|$. If y is a neighboring point of S with $|\nabla u(y)| \geq \mu$, we could add the point y to S without increasing the NFA, contradicting the fact that S is maximal meaningful. Thus, all neighboring points of S have a modulus of the gradient lower than μ, and the statement of the proposition holds. □

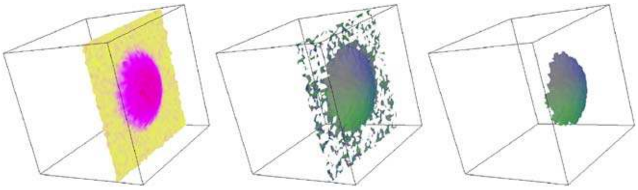


Fig. 5 The maximal meaningful subsets of a level surface may not correspond to physical features. *Left*: a level surface of an image colored by the contrast. *Middle*: the (connected) maximal meaningful subset of this surface. *Right*: the node of the Mumford-Shah hierarchy with maximal significativity

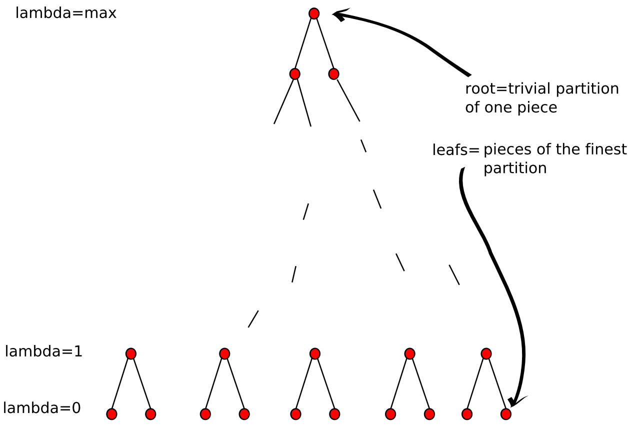

lambda=max
root=trivial partition of one piece
leafs= pieces of the finest partition
lambda=1
lambda=0

Fig. 6 A hierarchy of partitions whose depth is indexed by a scale parameter λ. The leaves of the tree represent the points of the discrete surface, and the root of the tree represents the whole surface. Notice that the total number of nodes, being a binary tree, is proportional to the number of leaves (exactly the double minus one)

Proposition 4 suggests an strategy for computing the maximal meaningful subsets of $\mathcal{S}$. We can find one of them on the collection of upper level sets of $|\nabla u|$ restricted to $\mathcal{S}$, and then searching recursively into its complementary.

But it turns out that the maximal meaningful subset of a level surface tends to be topologically very complex, with many holes and a complicated boundary. This happens because there is no restriction on the form of a maximal meaningful subset, besides being connected. See Fig. 5.

The proposed approximation, suggested by the observation above, is to restrict ourselves to a reduced class of well-behaved subsets. A reasonable way to produce such a class of connected subsets of all sizes is a *hierarchy* of partitions (Fig. 6).

Definition 5 A *hierarchy of partitions* over a set M is a family H of subsets of M such that

- $M \in H$.
- There is a subclass $L \subset H$, whose elements are disjoint and cover M. They are called the *leaves* of the hierarchy.
- Any element of H which is not a leaf can be represented as a disjoint union of leaves.
- Any pair of elements of H are either disjoint or nested.

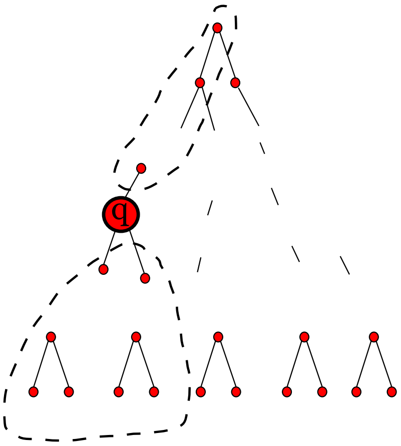


Fig. 7 Each node of the tree in Fig. 6 represents a connected patch of surface. Once we have selected the most meaningful node (in this figure, the enlarged one), we can remove all the nodes that are not disjoint with this one. They are all the ancestors and descendants, marked by the *dotted line* in this figure. Then we are left with the rest of the nodes in the tree

Restricting the family of connected subsets of level surfaces $\mathcal{S}$ to a hierarchy of partitions we define $F(\mu, l)$ as in 2, with N equal to the sum of all nodes of the hierarchies associated to all $\mathcal{S}$.

Once we have a hierarchy of partitions for a given level surface, it is easy to select the maximal meaningful objects of this partition in a greedy way (see Fig. 7). We first compute the meaningfulness of each object, which can be done in linear time in the case $f = \min$. Then we pick the object which is most meaningful. This clears from the search all the descendants and ancestors of this object within the tree of subsets, because we want a set of disjoint patches. Then we pick the most meaningful object in the remaining part of the tree, and we keep doing that iteratively until no more patches can be picked.

4.2 Mumford-Shah Surface Partition Hierarchy

In the previous paragraph, we have proposed to restrict the search of well-contrasted subsets of a level surface to a hierarchy of partitions over it. Our purpose now is to construct a particular hierarchy based on a family of nested segmentations computed using the simplified Mumford-Shah energy functional [37, 47] applied to the contrast function $g = |\nabla u|$ on a level surface.

Let M be a surface and suppose we want to approximate a real-valued function $g : M \to \mathbb{R}$ using piecewise constant functions. If a partition of M into regions is given, the best piecewise constant approximation to g is obtained by selecting the mean value of g as the value over each region. The piecewise constant Mumford-Shah functional assigns an energy to such a partition of M, depending on a non-negative parameter λ:

$$E(\text{Partition } \Omega_1, \dots, \Omega_n) = \sum_i \int_{\Omega_i} |g - m_i|^2 + \lambda \sum_{i,j} l_{ij},$$

where m_i is the mean value of g on the region Ω_i, and l_{ij} is the length of the common boundary of the regions Ω_i and Ω_j. Notice that the first term is the variance of the approximation (that decreases for finer partitions) and the second therm is the length of all the boundaries (that decreases for coarser partitions). The Mumford-Shah segmentation of g at scale λ is defined as the partition of M that minimizes the energy above. The parameter λ acts as a scale of the approximation. For $\lambda = 0$, the optimal partition is given by the connected regions of M where g is constant. As λ grows, we get coarser and coarser partitions until we find the trivial partition that has M as a single patch. These partitions do not necessarily form a hierarchy, that is, the finer can not necessarily be obtained as refinements of the coarser. In [37], Koepfler, López and Morel proposed to act *as if* these partitions were, in fact, hierarchical. Then, it is possible to start from the finest possible partition, compute its Mumford-Shah energy, and coarsen the partition by merging the pair of regions that make the energy decrease as much as possible.

From the previous expression of the functional, we can compute how does the value of E change when we merge two contiguous regions of the partition. After some algebraic manipulation, we find that when we merge the regions Ω_i and Ω_j to obtain a new region Ω_{ij} the increment of E is

$$E(\text{after}) - E(\text{before}) = \frac{|\Omega_i||\Omega_j|}{|\Omega_i| + |\Omega_j|} |m_i - m_j|^2 - \lambda l_{ij}$$

When this increment is negative it means that the approximation has improved after the merging, so that it is worth to merge. Therefore, to justify merging regions i and j they must fulfill the condition

$$\frac{|\Omega_i||\Omega_j|}{|\Omega_i| + |\Omega_j|} \frac{|m_i - m_j|^2}{l_{ij}} \le \lambda \qquad (3)$$

The left-hand side of this inequality is a number independent from λ, which can be stored for each edge of the region-adjacency graph. Doing so iteratively, starting from the discrete partition, one can store the whole history of the mergings in a tree structure. This results in a hierarchy (see Fig. 6) which is a good approximation of the set of all Mumford-Shah optimal segmentations varying λ.

We have used the setting above to segment the contrast over each level surface M of the image. The level surfaces are constructed, to sub-pixel accuracy, using a slightly modified version of the Marching Cubes [39] algorithm, designed

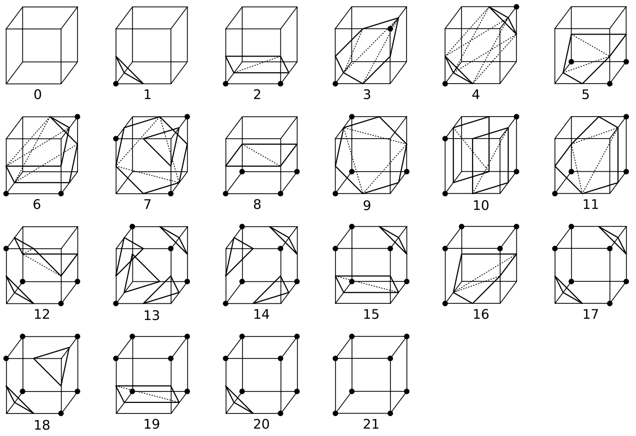


Fig. 8 Adaptation of the Marching Cubes table to the topology of upper semi-continuous interpolation. This table is used to compute the boundaries of shapes at a high resolution. To obtain sub-voxel precision, as customary, the vertices are not located at the midpoints of the cube edges, as shown in the figure, but at a linearly interpolated places between the corresponding gray-values. The triangulated surfaces obtained by this method have the same topology as the Shapes of the tree defined on Sect. 2, and this is not true for other versions of the table

to be consistent with our upper-semi-continuous interpretation of the discrete image. See Fig. 8 for the details. The lengths of curves over M are defined (as in [5, 43]) using the edges of the dual graph of the triangulation, given by barycentric subdivision. The function g is the contrast interpolated trilinearly at the vertices of the triangulation from its values on the image rectangular grid. We could also have used a vector-valued image g, for example involving second derivatives, but its usefulness is yet a subject of further study. See [30] for a related work on the selection of good vector-valued image descriptors in the context of surface evolution.

As far as we know, this is the first time that the Mumford-Shah functional is used to segment data defined on surfaces.

4.3 3D Edge Detection Algorithm

On Sect. 3 we have explained how to detect when a set of points within a given family is significantly well-contrasted. On Sect. 4 we have explained how to produce a large but manageable family of subsets to apply this test to. The sets of this family are patches of level surfaces, of all sizes, where the image contrast is as homogeneous as possible. Putting these two ingredients together we obtain the proposed edge detector:

Input:

- Original gray scale image, u
- Sensitivity parameter, $\varepsilon > 0$, ($\varepsilon = 1$ by default)

Output:

- A set of patches of surface, Γ

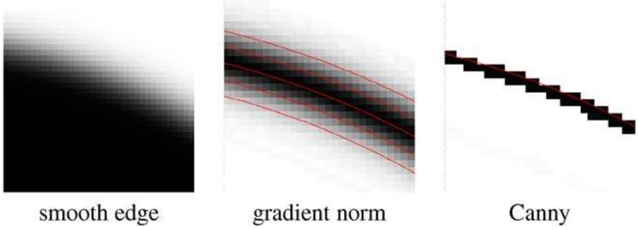


Fig. 9 Combination of our method with Canny's. We can use the output of Canny detector as the contrast for our method, thus enhancing the localization of the detected features. In this figure we show the detected curves of a synthetic 2D smooth edge in both cases

Algorithm:

1. Compute the image of contrast $g = |\nabla u|$.
2. Let N be twice the sum of the surface areas of all the level surfaces. This will be the total number of tests to be done.
3. For each connected component S of each level surface of the gray scale image:
   (a) Generate a mesh of triangles to represent S
   (b) Interpolate the contrast at the vertices of the triangulated surface S
   (c) Compute the Mumford-Shah tree, T, of the contrast function on S
   (d) Perform the statistical test with $f = \min$ to all the nodes of T
   (e) While there are still nodes in T:
      (i) Pick the node q of T that passes the statistical test with highest score
      (ii) Output the patch of surface corresponding to q
      (iii) Remove from T the node q and all its ancestors and descendants.

Notice that this modular design allows us to try some variations of the algorithm. For instance, we can use a different statistical test (as described on Sect. 5), or a different set of surface patches (for example, the shapes of the tree). Another variation that gives specially good results consists in setting the contrast g equal to the output of Canny operator, instead of the norm of the gradient. See Fig. 9 for an example where this makes a difference.

5 Pre and Post-processing Steps

The proposed method can be readily implemented as described. But in that case we observe that it is fairly slow and the output it gives is redundant (the detected edges are replicated in slightly different positions). To improve these two practical aspects we suggest filtering out the level surfaces of the tree that will not produce edges, and applying an exclusion criterion that forces that any voxel of the image only "belongs" to one output edge.

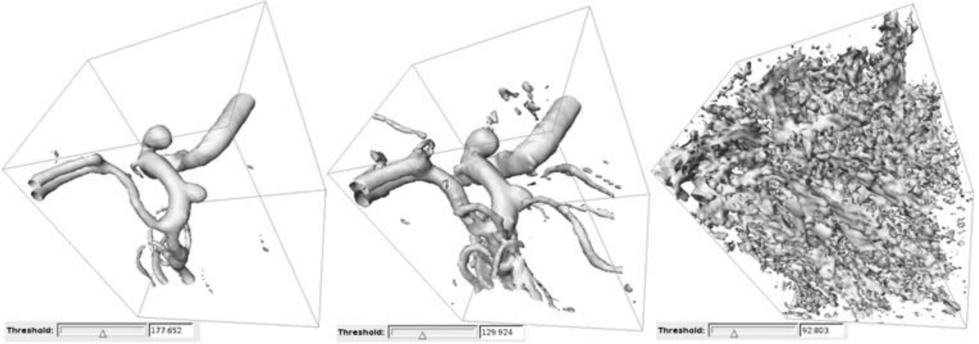


Fig. 10 Three thresholds of the same image. The first two thresholds show some image content. The third one shows mainly background noise. The level surface on the third image has one large and very convoluted component and many small and almost spherical components

5.1 Pruning the Original Tree of Level Surfaces

Let us study the cost of our algorithm. An image of m voxels has $O(m)$ level surfaces (in fact, it has exactly m when all the values are different). A typical level surface has $O(m^{\frac{2}{3}})$ points (this is the area of one side of a cube of volume m). This is a very rough estimate, which happens to underestimate the complexity of the algorithm for real images, for further empirical analysis on this topic see [11] and for a mathematical justification see [2]. Thus, the cost of traversing all the points of all level surfaces is about $O(m^{\frac{5}{3}})$, which is very large, and therefore too slow to scan the surfaces. Recall that a typical size for medical images is $m = 128^3$. We can make the algorithm much faster by discarding from the beginning those level surfaces which we know beforehand that will not produce any useful patches.

Here we discuss three ways to prune the input tree to reduce the number of processed surfaces: pruning very small surfaces, filtering the tree using robust statistics, and pruning the tree using the gray-level values. These are independent steps. The first one does not require any a priori information, but can incorporate it. The other two steps may be applied or not whether we have the required a priori information.

The first pre-processing step we propose is *pruning the smallest shapes of the tree*. When working with the tree of shapes of real images, one notices that usually most of the shapes in the tree have a small volume and belong to the noise that appears inside homogeneous regions (see Fig. 10, right). We can realize this behavior by plotting the number of shapes of each volume, as done in Fig. 11. A good model for the number of small shapes in a textured region of volume M is a power law of the form $p(v) = \frac{M}{6v^{3/2}}$, meaning that there are about $p(v)$ shapes of volume v (see also [2]). This model seems to be independent of the kind of noise and is quite accurate (for the purpose of realizing that most shapes in the tree are rather small) for $v < 20$. On Fig. 11 the number of small shapes for some images is plotted as dots, and the estimated power-law is plotted as a continuous line. The only differences that we observe are due to images with large saturated regions, where there is no texture.

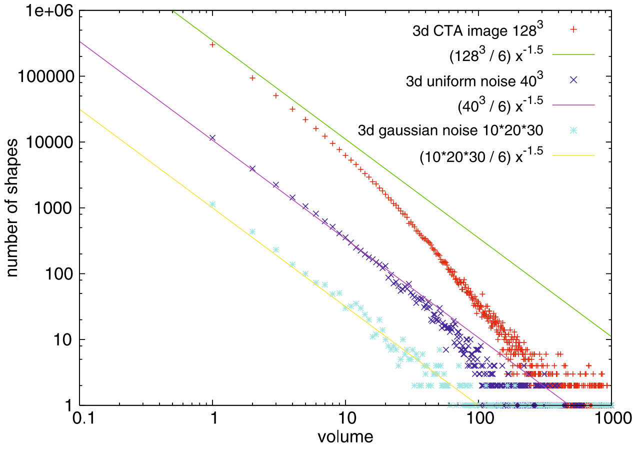


Fig. 11 Number of shapes of each size, for some images. The interesting pattern is that all the curves start decreasing more or less linearly with slope $-\frac{3}{2}$ (on the log–log scale shown here). This means that the number of shapes of each size decreases very fast, as there are $\propto v^{-1.5}$ shapes of volume v, for small v, and these small shapes belong to the texture of noise. This also means that most of the shapes on the tree are small: in a typical tree of shapes, about half of the shapes enclose a single voxel

All of these surfaces (say, of volume less than 10 voxels) are too small to pass the statistical test, so they can be discarded from the beginning. This will effectively discard most level surfaces of the image. While there is no study of the computational cost after this optimization, in practice this pruning helps to make the algorithm more tractable: for relatively small images of size 60^3 our implementation on a PC takes between one and five minutes, and we could process images of up to 128^3 voxels in less than one hour.

The number "10" in the previous paragraph is only an example. An appropriate bound can be computed from the image data. For instance, when $f = \min$ and $\varepsilon = 1$, a set of size l and minimum contrast μ is meaningful when $NH(\mu)^l < 1$, or equivalently when $l > -\log(N)/\log(H(\mu))$. Thus, if μ_1 is the last-to minimum contrast of the whole discrete image, then $-\log(N)/\log(H(\mu_1))$ is a lower bound for the size of a meaningful subset. This lower bound is not trivial when $NH(\mu_1) > 1$, which is usually the case. When f is another robust statistic, this bound is not easy to compute analytically, but it can nevertheless be obtained by a pre-computed table lookup to find the inverse of $F_l(\mu)$. These bounds on the area of meaningful level surfaces provide equivalent bounds for the minimal *volume* of a meaningful level surface, because for a discrete image the perimeter of a surface is bounded by its volume (each boundarying voxel is also part of the interior).

A second pre-processing step that can be applied consists in *pruning out low contrasted shapes*. As in [20], the statistical tests of Sect. 3 can be applied to the set of all level surfaces (without breaking those into pieces). The purpose of this pre-processing is to reduce the number of level surfaces that will be analyzed by the Mumford-Shah hierarchy, hence reducing the computational time. This may be a risk when the discarded surfaces have well-contrasted parts. But it could be justified if we have the a priori information that this is not the case.

The third pre-processing step, which is only useful in some common special cases, is *pruning the tree of shapes using* a-priori *information*. This data structure allows to use easily some *a-priori* information about the intensity ranges of the desired objects that may drastically reduce computational burden. For example, if we know that all gray values in some interval belong to noise, or to structures we are not interested in, we can immediately discard the level surfaces of those values. In X-ray computed tomography images, where the gray values have physical meaning, this means that we can discard most bones and background structures from the beginning and significantly reduce computation time and increase the quality of the output.

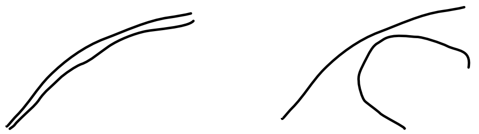
5.2 Filtering the Output Using the Exclusion Principle

The output of our edge detector (and that of DMM) is usually highly redundant in the following sense: edges appear represented as bundles of surfaces (or curves). Here we introduce an exclusion principle to reduce the redundancy of the output, by picking the best representative of each bundle. It is based on a similar principle used in a segment detector [22] to reduce output redundancy. As this method works exactly in the same way in 2D and 3D, we only describe here the 2D case where we can support the explanation with figures. For the 3D case, it suffices to replace "curve" by "surface" and "square" by "cube".

The proposed "exclusion principle" works by dividing the image domain into small square regions (e.g. of pixel size, but not necessarily so), and imposing these two requirements on the final set of curves:

(i) Each square belongs to at most one curve
(ii) Each curve passes the statistical test

Note that we say that a square P *belongs* to a curve C when C crosses through P. Of course, the first requirement is not usually fulfilled by the original set of curves. The exclusion method works by removing parts of curves until the first requirement is fulfilled. Then, it removes the remaining pieces of curves that do not pass the test. See Figs. 12, 13, 14 and 15 for a graphical explanation.



(a) Two curves "covering the same object"

(b) Two curves "covering different objects"

Fig. 12 The two synthetic cases that we are going to consider below

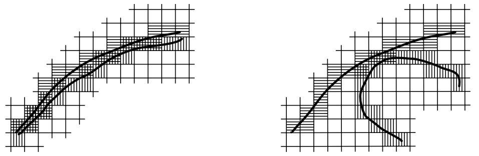


Fig. 13 Marking the squares according to which curves cross each one

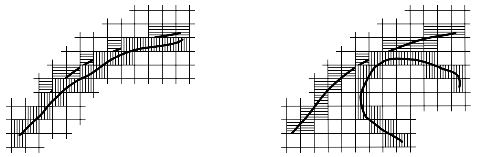
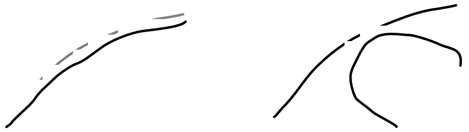


Fig. 14 Assignment of at most one curve to each square, thus fulfilling the first requirement



(a) Only the lower curve passes the statistical test

(b) Both parts pass the statistical test

Fig. 15 Performing the statistical test for the remaining pieces of curve, thus fulfilling the second requirement

There are in general non-unique ways to reduce the original set of curves so that the first requirement is true. We propose the following greedy strategy to force uniqueness:

1. Start with the set of all curves
2. While there are still curves that pass the test:
   (a) Pick the curve C that passes the test with highest score
   (b) The curve C owns all the squares that it crosses
   (c) Delete the parts of all the other curves that cross through squares owned by C
   (d) Output C and remove it from the set
3. Delete the remaining curves

Remark 2 In the previous algorithm, the “curves” we speak about are not necessarily connected. For example, when we remove a piece in the middle of a curve, the remaining two pieces are still considered “one curve”. This can be seen on the upper curve at Fig. 15(b).

Remark 3 The proposed exclusion principle has a scale parameter, namely the size of the grid. We make the natural proposal to set it to the same size as the voxels of the original image.

5.3 Merging the Patches (or *Edge Linking*)

Our edge detector does not produce a segmentation of the image domain into parts, but a set of boundaries. This may be enough for some applications like visualization or detection of structures, but it does not correspond to a segmentation. Let us discuss here a procedure to paste together a set of patches of level surfaces to produce a closed output surface (or a set of them). The method is based on a reconstruction algorithm introduced in [56].

Let $S \subseteq \Omega$ be a set of edges and let $d_S : \Omega \to \mathbb{R}$ be the distance function to S. Suppose that S covers part of the boundary of an object. A common approach to recover the whole boundary of the object is to search for closed surfaces Γ that are local minima of the following functional

$$E_0(\Gamma) = \int_\Gamma d_S(x) dA$$

where dA denotes the area element. These minima can be found by starting from an initial guess, a user-supplied closed surface which approximately contours the object, and then letting it evolve by gradient descent of E_0. The Gâteaux derivative of E_0 is

$$\nabla d_S(x) \cdot N + d_S(x)\kappa$$

where N is the outwards unit normal to Γ and κ its mean curvature. In an implicit formulation, where Γ is the zero level set of a function ϕ, the gradient descent of E_0 can be described by the evolution of the following PDE:

$$\frac{\partial \phi}{\partial t} = \left(\nabla d_S \cdot \frac{\nabla \phi}{|\nabla \phi|} + d_S \mathrm{div}\left(\frac{\nabla \phi}{|\nabla \phi|} \right) \right) |\nabla \phi|$$

Further regularization (mainly for display purposes) can be added to the model if we replace the functional above by adding the term $\lambda \int_S dA$ to the above functional, $\lambda > 0$. For large values of λ the functional approaches the area times λ, and its optima approach minimal surfaces. For small values of λ, only sharp edges are smoothed out.

Remark 4 This reconstruction method is not definitive (it will fail at junctions where more than three regions meet), but it usually improves the quality of the visualization by smoothing out the ragged appearance of the surface patches. It has two important problems to be used in full generality for this application. The first problem is the choice of the initial surface. Examples of reasonable initial surfaces are the boundary of the image or the most meaningful shape of the tree, but either method can fail when there are multiple objects to be detected, specially when they are nested. The second problem is that the functional E_0 has a global minimum of zero (attained at the empty surface). In practice, this means that the minimization can collapse if the surface “misses” the objects that we want to reconstruct. We are currently working on a modification of the functional to ammend this problem, in order to achieve an almost-automatic reconstructor.

6 Experiments and Discussion

We display and comment the results of our method when applied to four sample images, two synthetic images, a magnetic resonance and an X-ray computed tomography. In the synthetic images the task is to find the boundaries that generated them. In both medical images the task is to find the border of a vessel that contains a cerebral aneurysm. We compare the results with those obtained by simple thresholding and by Canny’s filter.

The first example is a synthetic image built in the following way. A sphere of radius 15 has been drawn at the center of a 40^3 black image, and the interior of the sphere has been colored in three different homogeneous regions. Then, a Gaussian noise of variance 10 has been added to the image, to add some texture. Figure 16 shows a slice of this image, and the output of the proposed edge detector. Notice that the output is a set of three smooth level surfaces, corresponding to the boundaries of the four large homogeneous regions on the image. This first example is a best case for our method, and serves as a check that the algorithm is working well. Notice that no global threshold can produce all the boundaries of the image, and that Canny’s detector misses the junctions.

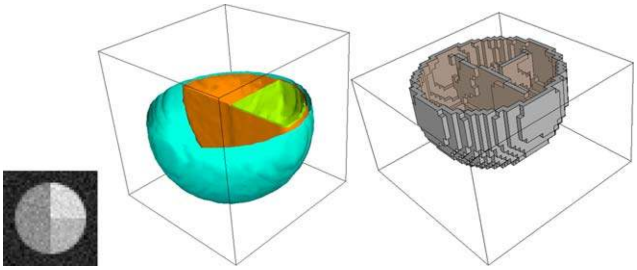


Fig. 16 Best case for our algorithm. Segmentation of a piecewise constant image with added texture. From *left* to *right*: slice of the image, output of the proposed edge detector, output of Canny’s edge detector. The 3D images are clipped to show the interior of the object

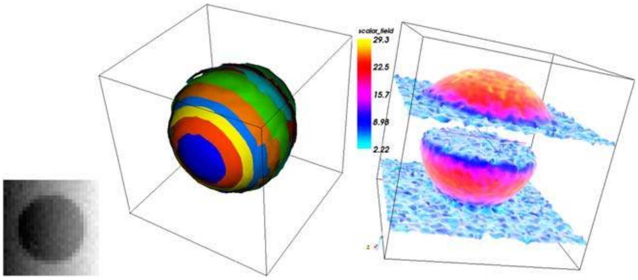


Fig. 17 Worst case for our algorithm. Segmentation of a piecewise constant image with an added ramp function. From *left* to *right*: slice of the image, output of the proposed edge detector, two isosurfaces of the image

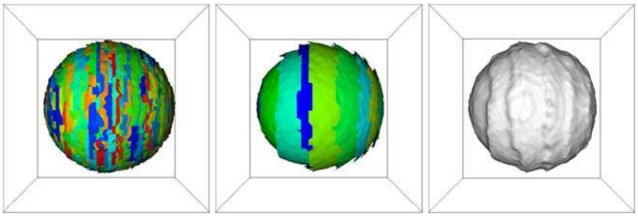


Fig. 18 Effect of the post-processing pipeline on the synthetic worst-case image (Fig. 17). *Left*: before exclusion principle, 207 patches. *Middle*: after exclusion principle, 9 patches. *Right*: after edge linking, one single surface patch. In this figure, the edge linking is performed using a higher resolution than the input image and without any smoothing. When using the same resolution as the input image, an almost perfectly spherical surface is obtained

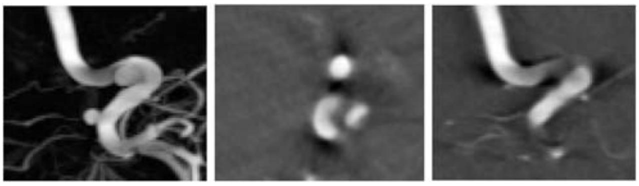


Fig. 19 A real CT image. From *left* to *right*: Maximum intensity projection, vertical slice, horizontal slice. The slices show several artifacts, and the anisotropy of noise

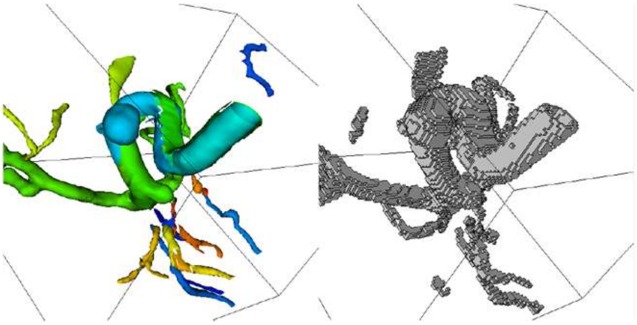


Fig. 20 Edge detection on the CT image. *Left*: Output of the proposed edge detector. *Right*: Output of Canny's edge detector, using hand-tuned parameters to obtain the best result for this image. The main advantage of the proposed method, besides the lack of tunable parameters, is the format of the output: instead of a large set of voxels we have a small set of triangulated surfaces, sampled at sub-voxel precision

The second example is a synthetic image built in the following way. A black sphere of radius 19 has been drawn at the center of a 50^3 white image, and then we added to it a ramp function of slope 1. This means that the contrast of both the background and the inside of the sphere are constant (equal to 1) and the borders have a much higher contrast. Then the image has been made more textured by adding a Gaussian noise of variance 5 and blurring it with a Gaussian of width 3. Two level surfaces of this image are shown in the right part of Fig. 17. Notice that no level surface can surround the whole sphere, but that many surfaces contain a band touching the sphere. This image is an example of a worst case for our algorithm (and a best case for Canny's). However, the algorithm manages to find the good parts of all level surfaces, as shown on Figs. 17 and 18.

Remark 5 The two synthetic images above are intentionally low-resolution to emphasize the sub-pixel accuracy of the output.

The first real example we show is the computed tomography image discussed in the introduction (see Fig. 1). Its size is 180 × 84 × 72. It is a noisy image with several artifacts (e.g. dark shadows, radial anisotropic noise), due to the reconstruction algorithm. The proposed edge detector finds the correct boundaries at several difficult places, but still misses some small arteries. See Fig. 19 for a discussion on the image, and Fig. 20 for the output of the proposed edge detector, and a comparison with Canny's. After linking the patches via the functional described on Sect. 5, we find a single surface which is better than the best manually-set global threshold, see Fig. 21.

The second real image is an anatomic MRI image of size 111 × 65 × 57. While this image is not as noisy as the previous one, it has an artifact which produces a problem like that of the second synthetic example. Namely, the image domain is partitioned into three bands, where the gray levels have a different starting point (so that the histograms are displaced). Even if this is an artifact easily tractable in a preprocessing stage, the proposed edge detector produces good results when applied directly on the raw data. See Fig. 22 for slices and projections, and Fig. 23 for the result of the proposed edge detection, compared to a manually selected isosurface.

7 Conclusion and Future Work

We have proposed a new edge detector that selects the well-contrasted patches of all level surfaces of a 3D image. The

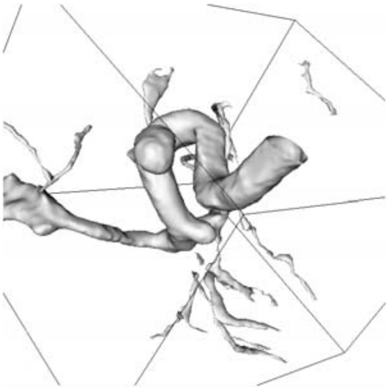


Fig. 21 Output of the experiment on Fig. 20, after joining the edges via Osher-Zhao functional. The result is a clean set of five disjoint surface patches

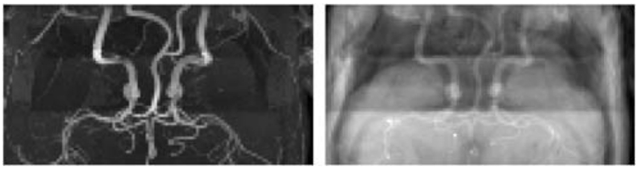


Fig. 22 A real MRI image. *Left*: Maximum intensity projection. *Right*: Average projection (notice the different averages over three bands). The results of our edge detector on this image appear on Fig. 23

proposed method is free of parameters and is based on the *a-contrario* methods developed by Desolneux et al. [20]. The edges obtained are robust and can be used for later purposes such as segmentation, registration, or visualization of the main structures of the image. We have also discussed an algorithm for edge linking to produce a closed surface passing nearby a given set of edges. We have illustrated the results with a series of experiments on synthetic and real data.

The proposed method raises a series of questions that need to be further studied. First, we are currently working on an improvement of the interpolation method proposed by [56], in order to avoid its dependence on the initialization and its local minima. Second, a systematic testing on sets of real and synthetic images, evaluating the results using different quantitative criteria, is also the object of further research.

A third line of future research concerns the efficiency of the presented algorithms. Medical images tend to be very large, e.g., a floating-point image of size 256^3 needs 64 MB to be stored. Even if our algorithms have been implemented to be as fast as possible, they are too slow to be used on regular computers for processing images of that size. We believe that any significant speed improvement needs to rely on a multi-scale representation of the data. For that, we would like to interleave the tree of shapes with a scale-space representation. This would lead to an efficient scheme to obtain finer and finer results starting from an initial coarse guess of the edges. Even if the finest scale was to be achieved at the end, this order of computation would be much faster than the current method, because large volumes of low contrast would be discarded at the coarser scales. In the current implementation, most of the running time is spent in discarding the extremely complicated high-resolution level surfaces that lie in the large and low contrasted parts of the image. To achieve interactive running times this is the problem that needs to be solved.

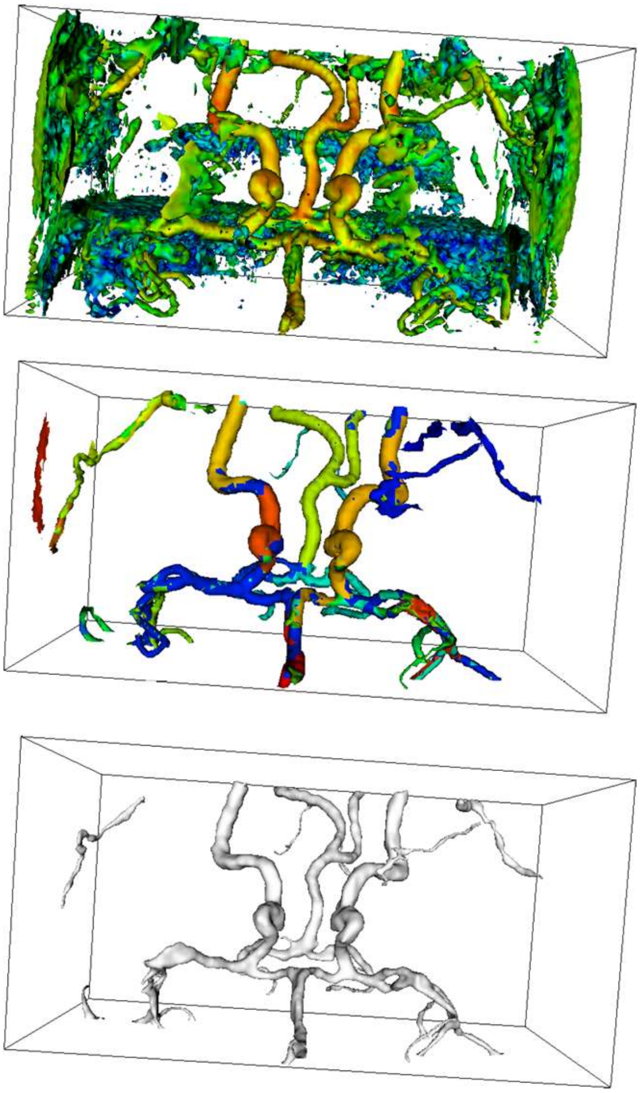


Fig. 23 Processing of the real MRI image from Fig. 22. *Top*: manually selected isosurface that segments well the upper part of the image. *Middle*: output of the proposed edge detector, 14 surface patches. Notice that the two structures which are not vessels (they are parts of bones) can be easily removed manually. *Bottom*: result of edge linking via Osher's functional

Acknowledgements V. Caselles acknowledges partial support by the PNPGC project, reference MTM2006-14836. E. Meinhardt acknowledges support by the FPI grant BES-2004-3723 and ALFA project CVFA II-0366-FA. This work was partially funded by grants TEC2006-03617/TCM from the Spanish MEC, the CENIT-CDTEAM Project funded by the Spanish CDTI-MITYC, and the @neurIST

Project (IST-2007-027703) supported by the European Commission. We would like to thank Juan Cardelino and Gregory Randall for enlightening discussions on the topic of this paper.

References

1. Abramowitz, M., Stegun, I.A.: Handbook of Mathematical Functions. Dover, New York (1965)
2. Alvarez, L., Gousseau, Y., Morel, J.-M.: The size of objects in natural and artificial images. Adv. Imaging Electron. Phys. **111**, 167–242 (1999)
3. Ayache, N., Faverjon, B.: Efficient registration of stereo images by matching graph descriptions of edge segments. Int. J. Comput. Vis. **1**(2), 107–131 (1987)
4. Ballester, C., Caselles, V., Monasse, P.: The tree of shapes of an image. ESAIM: Control, Optim. Calc. Var. **9**, 1–18 (2003)
5. Banchoff, T.: Critical points and curvature for embedded polyhedra. J. Differ. Geom. **1**, 245–256 (1967)
6. Beucher, S., Lantuejoul, C.: Use of watersheds in contour detection. In: Proc. Int. Workshop Image Processing (1979)
7. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: Proc. 9th Int. Conf. Comput. Vis., pp. 26–33 (2003)
8. Brown, L.G.: A survey of image registration techniques. ACM Comput. Surv. **24**, 325–376 (1992)
9. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **8**(6), 679–698 (1986)
10. Cao, F., Musé, P., Sur, F.: Extracting meaningful curves from images. J. Math. Imaging Vis. **22**(2), 159–181 (2005)
11. Carr, H., Duffy, B., Denby, B.: On histograms and isosurface statistics. IEEE Trans. Vis. Comput. Graph. **12**(5), 1259–1266 (2006)
12. Carr, H., Snoeyink, J., Axen, U.: Computing contour trees in all dimensions. Comput. Geom.: Theory Appl. **24**(2), 75–94 (2003)
13. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. Int. J. Comput. Vis. **22**(1), 61–79 (1997)
14. Caselles, V., Meinhardt, E., Monasse, P.: Constructing the tree of shapes of an image by fusion of the trees of connected components of upper and lower level sets. Positivity **12**(1), 55–73 (2008)
15. Clark, J.H.: Hierarchical geometric models for visible surface algorithms. Commun. ACM **19**(10), 547–554 (1976)
16. Cohen, I., Cohen, L.D., Ayache, N.: Using deformable surfaces to segment 3-D images and infer differential structures. CVGIP: Image Underst. **56**(2), 242–263 (1992)
17. Cox, J., Karron, D.B., Ferdous, N.: Topological zone organization of scalar volume data. J. Math. Imaging Vis. **18**(2), 95–117 (2003)
18. Davis, L.S.: A survey of edge detection techniques. Comput. Graph. Image Process. **4**, 248–270 (1975)
19. Deriche, R.: Using Canny's criteria to derive a recursively implemented optimal edge detector. Int. J. Comput. Vis. **1**(2), 167–187 (1987)
20. Desolneux, A., Moisan, L., Morel, J.-M.: Edge detection by Helmholtz principle. J. Math. Imaging Vis. **14**(3), 271–284 (2001)
21. Desolneux, A., Moisan, L., Morel, J.-M.: Gestalt theory and computer vision. In: Seeing, Thinking and Knowing, pp. 71–101. Springer, Berlin (2004)
22. Desolneux, A., Moisan, L., Morel, J.M.: Meaningful alignments. Int. J. Comput. Vis. **40**(1), 7–23 (2000)
23. Desolneux, A., Moisan, L., Morel, J.M.: Variational Snake Theory. Geometric Level Set Methods in Imaging, Vision, and Graphics. Springer, Berlin (2003)
24. Forsyth, D.A., Ponce, J.: Computer Vision: A Modern Approach. Prentice Hall, New York (2003)
25. Galassi, M. et al.: GNU Scientific Library. Network Theory Ltd. (2002)
26. Grimson, W.E.L., Huttenlocher, D.P.: On the verification of hypothesized matches in model-based recognition. IEEE Trans. Pattern Anal. Mach. Intell. **13**(12), 1201–1213 (1991)
27. Haralick, R.M.: Digital step edges from zero crossing of second directional derivatives. IEEE Trans. Pattern Anal. Mach. Intell. **6**(1), 58–68 (1984)
28. Haralick, R.M., Shapiro, L.G.: Image segmentation techniques. Comput. Vis. Graph. Image Process. **29**, 100–132 (1985)
29. Haralick, R.M., Watson, L.T., Laffey, T.J.: The topographic primal sketch. Int. J. Robot. Res. **2**(1), 50–72 (1983)
30. Hernandez, M., Frangi, A.F.: Non-parametric geodesic active regions: Method and evaluation for cerebral aneurysms segmentation in 3DRA and CTA. Med. Image Anal. **11**(3), 224–241 (2007)
31. Hsieh, J.W., Liao, H.Y.M., Fan, K.C., Ko, M.T., Hung, Y.P.: Image registration using a new edge-based approach. Comput. Vis. Image Underst. **67**(2), 112–130 (1997)
32. Hsu, L.Y., Loew, M.H., Ostuni, J.: Automated registration of brain images using edge and surface features. IEEE Eng. Med. Biol. Mag. **18**(6), 40–47 (1999)
33. Julesz, B.: A method of coding TV signals based on edge detection. Bell Syst. Technol. **38**(4), 1001–1020 (1959)
34. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. Int. J. Comput. Vis. **1**(4), 321–331 (1988)
35. Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A., Yezzi, A.: Conformal curvature flows: From phase transitions to active vision. Arch. Ration. Mech. Anal. **134**(3), 275–301 (1996)
36. Kirsch, R.A.: Computer determination of the constituent structure of biological images. Comput. Biomed. Res. **4**(3), 315–328 (1971)
37. Koepfler, G., Lopez, C., Morel, J.-M.: A multiscale algorithm for image segmentation by variational method. SIAM J. Numer. Anal. **31**(1), 282–299 (1994)
38. Lisani, J.L., Moisan, L., Monasse, P., Morel, J.-M.: On the theory of planar shape. Multiscale Model. Simul. **1**, 1 (2003)
39. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3D surface construction algorithm. In: *Proc. 14th Annu. Conf. Comput. Graph. Interact. Tech.*, pp. 163–169 (1987)
40. Lowe, D.G.: Perceptual Organization and Visual Recognition. Kluwer Academic, Dordrecht (1985)
41. Malladi, R., Sethian, J.A., Vemuri, B.C.: Shape modeling with front propagation: a level set approach. IEEE Trans. Pattern Anal. Mach. Intell. **17**(2), 158–175 (1995)
42. Marr, D., Hildreth, E.: Theory of edge detection. Proc. R. Soc. Lond. **207**(1167), 187–217 (1980)
43. Meyer, M., Desbrun, M., Schroder, P., Barr, A.H.: Discrete differential-geometry operators for triangulated 2-manifolds. Vis. Math. **3**, 35–57 (2002)
44. Mokhtarian, F., Abbasi, S., Kittler, J.: Robust and efficient shape indexing through curvature scale space. In: Proc. Br. Mach. Vis. Conf., pp. 53–62 (1996)
45. Monasse, P.: Contrast invariant representation of digital images and application to registration. PhD thesis, Université Paris IX-Dauphine, June 2000
46. Monasse, P., Guichard, F.: Fast computation of a contrast-invariant image representation. IEEE Trans. Image Process. **9**, 860–872 (2000)
47. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Commun. Pure Appl. Math. **42**(5), 577–685 (1988)
48. Pascucci, V., Cole-McLaughlin, K.: Parallel computation of the topology of level sets. Algorithmica **38**(2), 249–268 (2003)
49. Pielot, R., Scholz, M., Obermayer, K., Gundelfinger, E.D., Hess, A.: 3D edge detection to define landmarks for point-based warping in brain imaging. In: Proc. Int. Workshop Image Process., vol. 2 (2001)

50. Prewitt, J.: Object enhancement and extraction. In: Picture Processing and Psychopictorics, pp. 75–149. Academic Press, New York (1970)
51. Roberts, L.: Machine perception of 3D solids. In: Opt. Electro-Opt. Inf. Process., pp. 159–197 (1965)
52. Sarioz, D., Kong, T.Y., Herman, G.T.: History trees as descriptors of macromolecular structures. In: Lect. Notes Comput. Sci., vol. 4291, p. 263. Springer, Berlin (2006)
53. Sijbers, J., Scheunders, P., Verhoye, M., Van der Linden, A., Van Dyck, D., Raman, E.: Watershed-based segmentation of 3D MR data for volume quantization. Magn. Reson. Imaging **15**, 679–688 (1997)
54. Song, Y., Zhang, A.: Monotonic tree. In: Proc. 10th Int. Conf. Discrete Geom. Comput. Imagery (2002)
55. Vincent, L., Soille, P.: Watersheds in digital spaces: an efficient algorithm based onimmersion simulations. IEEE Trans. Pattern Anal. Mach. Intell. **13**(6), 583–598 (1991)
56. Zhao, H.K., Osher, S., Fedkiw, R.: Fast surface reconstruction using the level set method. In: 1st IEEE Workshop Var. Lev. Set Methods, vol. 80(3), pp. 194–202 (2001)

Enric Meinhardt received the Llicenciatura degree in mathematics from the Technical University of Catalonia in 2003, the MSc degree in applied mathematics, ENS-Cachan in 2006 and the DEA degree in computer science at the Universitat Pompeu Fabra. Currently, he is a PhD student at the Universitat Pompeu Fabra. His research is focused on algorithms and data structures for 3D image processing and visualization.

Ernesto Zacur was born in 1978, in Santa Fe, Argentina, a city located 500 Km north Buenos Aires. He studied Physics at Universidad Nacional de Rosario until 2000. There, he obtained a scholarship to continue his studies in Bariloche, Argentina, at Instituto Balseiro. In December 2003, when he finished the courses and obtained the Master in Physics degree. He came to Spain for a Computer Vision PhD, under the supervision of Alejandro Frangi, at Department of Technology in the University Pompeu Fabra, in Barcelona. Currently, he is investigating Facial Biometrics with 3D Models and Learning Models optimization.

Alejandro F. Frangi received the MSc degree in telecommunications engineering from the Technical University of Catalonia (Barcelona) in 1996. He subsequently carried out research on electrical impedance tomography for image reconstruction and noise characterization at the same institution under a CIRIT grant. In 1997, he received a grant from the Dutch Ministry of Economic Affairs to pursue the PhD degree at the Image Sciences Institute of the University Medical Center Utrecht on model-based cardiovascular image analysis under the sponsorship of Philips Medical Systems, Nederland BV. After graduation in 2001, Dr. Frangi moved to Zaragoza, Spain. He was an assistant professor at the University of Zaragoza from 2001 to 2003. Subsequently, he was awarded a Ramon y Cajal Research Fellowship from 2003 to 2008, a national program of the Spanish Ministry of Science and Technology for promoting outstanding young investigators. From 2001 to 2004, he was affiliated with the Division of Biomedical Engineering at the Aragon Institute of Engineering Research, a multidisciplinary research institute of the University of Zaragoza. As of 1 September 2004, he was invited to join the Department of Technology at the Pompeu Fabra University in Barcelona, Spain. He moved to Barcelona with his research team. He currently leads the Computational Imaging Lab at the Pompeu Fabra University (www.cilab.upf.edu). His main research interests are in computer vision and medical image analysis with particular emphasis in model and registration-based techniques, and statistical methods. He has more than 20 journal papers in key international journals of his research field and more than 60 book chapters and international conference papers. He was a guest editor two times for two special issues of the IEEE Transactions on Medical Imaging and one time for Medical Image Analysis. He is a senior member of the IEEE and SPIE, an associate editor of the IEEE Transactions on Medical Imaging, and an associate editor of Medical Image Analysis.

Vicent Caselles received the Licenciatura and PhD degrees in mathematics from Valencia University, Spain, in 1982 and 1985, respectively. Currently, he is a Professor at the Universitat Pompeu Fabra, Barcelona, Spain. His research interests include image processing, computer vision, and the applications of geometry and partial differential equations to both fields.