

Research Article

A Doubly Adaptive Algorithm for Edge Detection in 3D Images

Sagrario Lantarón,¹ M. Dolores López,¹ and Javier Rodrigo²

¹*Departamento de Matemática e Informática Aplicadas a las Ingenierías Civil y Naval, ETSI de Caminos, Universidad Politécnica de Madrid, Avda. Profesor Aranguren s/n, 28040 Madrid, Spain*

²*Departamento de Matemática Aplicada, Escuela Técnica Superior de Ingeniería (ICAI), Universidad Pontificia Comillas, C/Alberto Aguilera 25, 28015 Madrid, Spain*

Correspondence should be addressed to M. Dolores López; marilo.lopez@upm.es

Received 31 May 2016; Accepted 22 September 2016

Academic Editor: Ziran Wu

Copyright © 2016 Sagrario Lantarón et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a new algorithm (DA3DED) for edge detection in 3D images. DA3DED is doubly adaptive because it is based on the adaptive algorithm EDAS-1 for detecting edges in functions of one variable and a second adaptive procedure based on the concept of projective complexity of a 3D image. DA3DED has been tested on 3D images that modelize real problems (composites and fractures). It has been much faster than the 1D edge detection algorithm for 3D images derived from EDAS-1.

1. Introduction

Three-dimensional imaging is applied in many topics such as medicine and materials science among others. 3D medical imaging models structures of the human body. This is important in many fields as image guided surgery, assessment of the quality of bones, and so forth; see, for example, [1–3]. The complex geometric structure of some materials (composites, foams, etc.) can be treated with 3D images: 3D images of materials provide data such as the 3D connectivity of a structure, distribution of particles, etc. These data can be used in simulations to compute macroscopic material properties, what can be considered the basis of virtual material design [4].

The detection of edges is an essential objective in image processing. Besides the applications aforementioned, it is useful in the analysis and study of 3D Satellite images, among others. There are a lot of procedures for determining 3D edges. These procedures can be classified in Direct Methods and Indirect Methods [5–7]. Among the direct methods we can find 1D edge detection methods (1D3DED) [8, 9], spatial difference filters [10–12], polynomial fitting method [13], and deformable models [14–20]. We can add to the direct methods the adaptive splitting methods, which are based on the adaptive splitting of the domain of the function guided by

the value of an average integral. In [21], an algorithm (EDAS-3) to approximate the jump discontinuity set of functions defined on subsets of \mathbb{R}^3 based in these methods was proposed. This algorithm overcomes most inconveniences presented by deformable models. It exhibits effective edge detection in 3D models with different interface topologies. Also, EDAS-3 can obtain the edges with a prefixed accuracy.

The present paper shows an accurate algorithm (DA3DED) that can obtain a point based representation of the jump discontinuity set of a 3D image. The algorithm is a type of 1D edge detection method (1D3DED) and adaptive splitting method. It begins by considering a region in the plane x_1x_2 containing the projection of the image and a set of points in this region. Then, it considers straight lines perpendicular to the plane x_1x_2 and computes the edge points lying on these straight lines. This task is accomplished using the 1D edge detector EDAS-1 [22]. If the number of edges detected in the region is high or the distance between them is small, the algorithm concludes that the region under consideration is complex and performs a subdivision process; otherwise, the region is not divided and the algorithm studies a contiguous zone. The process is repeated for the x_1x_3 and x_2x_3 planes. The edge points obtained are stored in a file. In this way, complex regions are crossed by many lines and simple regions are crossed by few lines. The resulting

algorithm is doubly adaptive because the “straight line step” is performed in an adaptive way by EDAS-1 and the second step performs an adaptive process based on the complexity of the image. The algorithm uses less CPU time than previous algorithm [21]. We will give a C-like pseudocode description of the constituent algorithms of DA3DED.

Experimental results show a good behavior of DA3DED on practical 3D problems that model different types of composite materials and fractures.

The outline of the paper is as follows. Section 2 states some mathematical preliminary results and details the conceptual and algorithmic constituents of DA3DED: EDAS-1 algorithm, discrete and continuous representation of 3D images, projective complexity, subdivision procedures of discrete rectangles, and split criterion. It also describes the algorithm DA3DED. Section 3 shows computational experiments on models of complex materials. Section 4 provides some concluding remarks.

2. Materials and Methods

2.1. Mathematical Preliminaries. Let f be a general piecewise continuous function in $R \subset \mathbb{R}^d$, a compact d -interval.

We assume that $R = \bigcup_{i=1}^n C_i$, where $\{C_i\}_{i=1}^n$ is a series of connected sets with pairwise disjoint nonempty interiors and we call Γ_i be the boundary of C_i , $i = 1, \dots, n$.

Define $\Gamma \equiv \bigcup_{i=1}^n \Gamma_i$. We call Γ^J the set of points in Γ with jump discontinuities. Achieving a good approximation of Γ^J is our aim.

Let $\{\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_d\}$ be a set of $d + 1$ affinely independent vectors in \mathbb{R}^d and T the d -simplex generated by them. Consider a function $f : T \subset \mathbb{R}^d \rightarrow \mathbb{R}$. We call $L_T f$, the unique affine map such that

$$L_T f(\mathbf{v}_j) = f(\mathbf{v}_j), \quad j = 0, 1, \dots, d. \quad (1)$$

It is proved at [21, 22] that the behavior of the next average integral $AI_T(f)$ depends on the continuity or lack of continuity of f on T :

$$AI_T(f) \equiv \frac{\int_T |f(\mathbf{x}) - L_T f(\mathbf{x})| d\mathbf{x}}{\nu(T)}, \quad (2)$$

where $\nu(T)$ is the Lebesgue measure of T .

This result is the basis of the algorithm described in the next section.

For more details about the mathematical preliminaries of the problem, see [21, 22].

2.2. Constituents of the Algorithm DA3DED. DA3DED is based on several concepts and algorithms that are described in this subsection.

2.2.1. Algorithm EDAS-1. EDAS-1 is an algorithm to approximate the jump discontinuity set of functions of one variable. It is a particular case of the algorithm EDAS- d for functions defined on subsets of \mathbb{R}^d . In this section, we give an outline of

EDAS- d . Its validity for $d = 1, 2$ has been established in [22]. The case $d = 3$ has been studied in [21].

More details about the implementation and performance of EDAS-1 can be found in [22].

Next, the steps of the EDAS- d algorithm are summarized.

Step 1. Consider the initial partition ($j = 1$), $P_j \equiv \{T_i\}_{i=1}^n$ of $R = [\mathbf{a}, \mathbf{b}]$, where $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$.

Step 2. Put into the set \mathcal{G}_j the good simplices of P_j and put into the set \mathcal{B}_j the bad simplices of P_j .

T is a good simplex if $(AI_T(f) \leq E_1 \text{ or } |T| \leq E_2)$ and $|T| \leq E_4$, where $|T|$ is the diameter of T , E_1 is the maximum local error of the approximant L , E_2 is the approximation error of the points in Γ^J , and E_4 is a positive real parameter.

Step 3. Divide each bad simplex into two simplices, by splitting its largest edge. We have a new partition $j = j + 1$.

Step 4. Repeat Steps 2 and 3 until $\mathcal{B}_j = \emptyset$.

Step 5. $G = \mathcal{G}_j$ is the searched partition. Choose the following subset of G :

$$AI_{E_3}^J \equiv \left\{ T \in G : AI_T(f) > E_1, j^a \equiv \max_{\mathbf{v}_i \in \mathcal{V}(T)} \{f(\mathbf{v}_i)\} - \min_{\mathbf{v}_i \in \mathcal{V}(T)} \{f(\mathbf{v}_i)\} > E_3 \right\}, \quad (3)$$

where E_3 is the minimum magnitude of jump reported and E_2 is also a stopping criterion.

In the images studied in Section 3, a voxel is considered as an edge voxel if it contains some barycenter of the simplices in $AI_{E_3}^J$.

In practice, we have implemented the above algorithm using a binary tree whose leaves correspond to good simplices. In the case $d = 1$, the integrals $\int_T |f(x) - L_T f(x)| dx$ have been computed using the Gauss-Legendre integration formula [23]. More details about the implementation and performance of EDAS-1 can be found in [22].

2.2.2. Continuous and Discrete Representations of a 3D Image.

The algorithm DA3DED uses two different mathematical representations of a 3D image. It considers that the image is a function defined on a box in \mathbb{R}^3 when it computes edge points using EDAS-1. This continuous representation is also useful to define the concept of “projective complexity”; we will see this in the next subsection. In the remaining steps, DA3DED uses the usual discrete representation. Figure 1 shows the difference between both representations in the case of a 2D image. Below we define these concepts.

Let n and m be positive integers with $n < m$; define

$$[[n, m]] \equiv \{n, n + 1, \dots, m\} \quad (4)$$

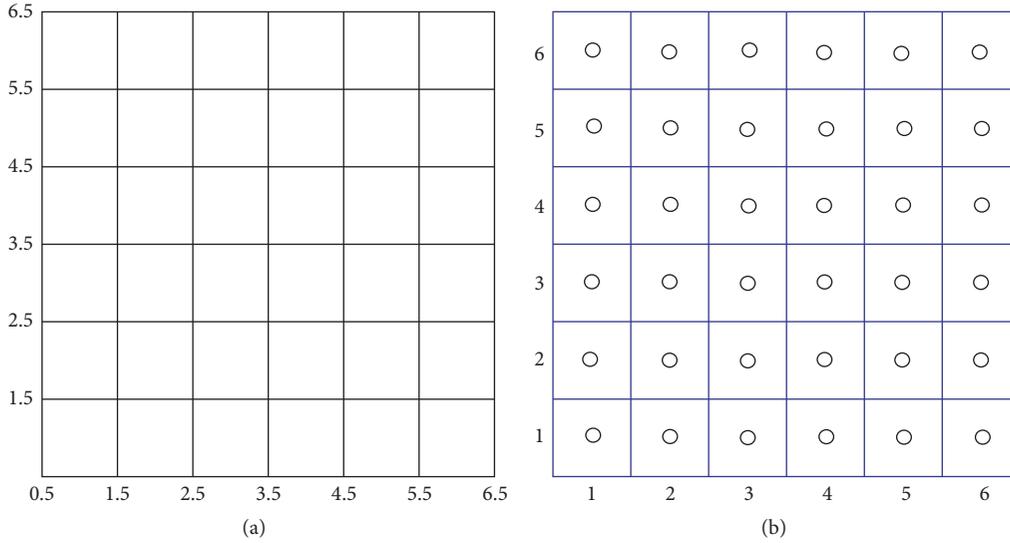


FIGURE 1: Continuous representation (a) and discrete representation (b) of a 2D image.

A 3D image is given by an $n_1 \times n_2 \times n_3$ array $I = \{I_{ijk}\}$ with indices $(i, j, k) \in [[1, n_1]] \times [[1, n_2]] \times [[1, n_3]]$. From a 3D image I , one can build a piecewise constant function \tilde{I} ,

defined on $[0.5, n_1 + 0.5] \times [0.5, n_2 + 0.5] \times [0.5, n_3 + 0.5]$ in the following way:

$$\tilde{I}(x_1, x_2, x_3) \equiv \begin{cases} I_{111}, & \text{if } (x_1, x_2, x_3) \in [0.5, 1.5] \times [0.5, 1.5] \times [0.5, 1.5], \\ I_{i11}, & \text{if } (x_1, x_2, x_3) \in [i - 0.5, i + 0.5] \times [0.5, 1.5] \times [0.5, 1.5] \quad (i > 1), \\ I_{1j1}, & \text{if } (x_1, x_2, x_3) \in [0.5, 1.5] \times [j - 0.5, j + 0.5] \times [0.5, 1.5] \quad (j > 1), \\ I_{ij1}, & \text{if } (x_1, x_2, x_3) \in [i - 0.5, i + 0.5] \times [j - 0.5, j + 0.5] \times [0.5, 1.5] \quad (i, j > 1), \\ I_{11k}, & \text{if } (x_1, x_2, x_3) \in [0.5, 1.5] \times [0.5, 1.5] \times [k - 0.5, k + 0.5] \quad (k > 1), \\ I_{i1k}, & \text{if } (x_1, x_2, x_3) \in [i - 0.5, i + 0.5] \times [0.5, 1.5] \times [k - 0.5, k + 0.5] \quad (i, k > 1), \\ I_{1jk}, & \text{if } (x_1, x_2, x_3) \in [0.5, 1.5] \times [j - 0.5, j + 0.5] \times [k - 0.5, k + 0.5] \quad (j, k > 1), \\ I_{ijk}, & \text{if } (x_1, x_2, x_3) \in [i - 0.5, i + 0.5] \times [j - 0.5, j + 0.5] \times [k - 0.5, k + 0.5] \quad (i, j, k > 1). \end{cases} \quad (5)$$

2.2.3. Projective Complexity of a 3D Image. Several complexity measures for 2D images have been proposed [24, 25]. Some definitions are based on statistical gray level features (global versus local histograms). Other definitions are based on spatial distributions of gray levels (edges, symmetry, and texture). Edges highlight image zones in which there are abrupt changes in luminosity level usually associated with surface discontinuities. The rationale is straightforward: images with more borders contain more objects (or objects with more facets), thus resulting in a greater perceived complexity. This has led to the following edge based measure of complexity. Complexity is measured by the number of edges per unit area in the image [26]. In the case of 3D images, we can extend this definition; for example, complexity can be measured by the number of edge voxels per unit volume.

This definition is not suitable for adaptive algorithms based on 1D edge detection methods. Therefore, we propose a new definition consistent with the above one.

Consider a continuously defined function $f : [a, b]^3 \rightarrow \mathbb{R}$. Let $[\alpha, \beta] \subset [a, b]$ and $S = [\alpha, \beta]^2$. Let $(x_1, x_2) \in S$ be a fixed point in S . Call $e(x_1, x_2)$ the number of edge points of the function of one variable $f(x_1, x_2, x)$ when $x \in [a, b]$. The projective complexity of f on the set $K = S \times [a, b]$ is defined as

$$\text{PC}(f, K) \equiv \frac{\int_S e(x_1, x_2) dx_1 dx_2}{v(S)}. \quad (6)$$

Let d be a positive integer. To approximate the above quantity, we subdivide S into d^2 identical squares. Denote the

center of the squares by (x_{1i}, x_{2j}) , $1 \leq i, j \leq d$. Then, (6) can be approximated by

$$\begin{aligned} PC^a(f, K) &= \frac{(\sum_{i,j=1}^d e(x_{1i}, x_{2j}))((\beta - \alpha)/d)^2}{(\beta - \alpha)^2} \\ &= \frac{\sum_{i,j=1}^d e(x_{1i}, x_{2j})}{d^2}. \end{aligned} \quad (7)$$

Observe that $PC^a(f, K)$ does not depend on the size of the square S . We can consider e as a vector with d^2 components and write (7) as

$$PC^a(f, K) = \frac{\|e\|_1}{d^2}. \quad (8)$$

In each experiment, the value of d has been almost constant. Then, using the norm equivalence, we can define the following measure of complexity:

$$PC^*(f, K) = \|e\|_\infty. \quad (9)$$

Although we have considered sets in the plane x_1x_2 , similar definitions can be stated for sets in the planes x_1x_3 or x_2x_3 . The above definitions are applicable to 3D images because they can be extended to continuously defined functions using the procedure detailed in Section 2.2.2.

2.2.4. Subdivision and Selection Procedures. Consider a 3D image given by an $n_1 \times n_2 \times n_3$ array $I = \{I_{ijk}\}$ with indices $(i, j, k) \in [[1, n_1]] \times [[1, n_2]] \times [[1, n_3]]$.

The doubly adaptive algorithm performs split and point selection operations. These procedures are defined on the set of indices; that is, they consider discrete rectangles. First, we describe how a discrete rectangle is divided into four almost equal discrete subrectangles.

Consider the discrete rectangle $\mathcal{R} = [[n, m]] \times [[p, q]]$ with $n < m$ and $p < q$. The subdivision is performed computing $r = n + \lfloor (m - n)/2 \rfloor$ and $s = p + \lfloor (q - p)/2 \rfloor$ ($\lfloor x \rfloor$ denotes the floor function that returns the greatest integer less than or equal to x). The resulting discrete rectangles are

$$\begin{aligned} \mathcal{R}_1 &= [[n, r]] \times [[p, s]], \\ \mathcal{R}_2 &= [[n, r]] \times [[s + 1, q]], \\ \mathcal{R}_3 &= [[r + 1, m]] \times [[p, s]], \\ \mathcal{R}_4 &= [[r + 1, m]] \times [[s + 1, q]]. \end{aligned} \quad (10)$$

Suppose now that we want to select $(d + 1)^2$ points from \mathcal{R} , distributed regularly. The selected points are obtained as the set

$$\begin{aligned} \mathcal{N} &= \{h[0], h[1], \dots, h[d]\} \\ &\quad \times \{v[0], v[1], \dots, v[d]\}, \end{aligned} \quad (11)$$

where the vectors h and v are given by the following algorithm:

$$\begin{aligned} &\text{left} = n - 0.5; \text{right} = m + 0.5; \text{up} = q + 0.5; \text{down} = \\ &p - 0.5; \\ &h[0] = n; h[d] = m; v[0] = p; v[d] = q; \\ &\text{for } (i = 1; i < d; i++) \\ &\{ \\ &\quad ht = \text{left} + i(\text{right} - \text{left})/d; \\ &\quad vt = \text{down} + i(\text{up} - \text{down})/d; \\ &\quad h[i] = \lfloor ht + 0.5 \rfloor; \text{if } (ht - \lfloor ht \rfloor == 0.5) h[i] = \\ &\quad h[i] - 1; \\ &\quad v[i] = \lfloor vt + 0.5 \rfloor; \text{if } (vt - \lfloor vt \rfloor == 0.5) v[i] = \\ &\quad v[i] - 1; \\ &\} \end{aligned}$$

Observe that $\lfloor x + 0.5 \rfloor$ returns the integer nearest to x . The above algorithm obtains a subset of $[[n, m]] \times [[p, q]]$ whose points are distanced approximately by $(n - m)/d((q - p)/d)$. If $d > n - m$ or $d > q - p$, the subdivision is not possible.

2.2.5. Split Criterion. The set of indices of a 3D image is given by

$$\mathcal{M} = [[1, n_1]] \times [[1, n_2]] \times [[1, n_3]]. \quad (12)$$

The projections of this set on the coordinate planes are

$$\begin{aligned} \mathcal{R}_a &= [[1, n_1]] \times [[1, n_2]], \\ \mathcal{R}_b &= [[1, n_1]] \times [[1, n_3]], \\ \mathcal{R}_c &= [[1, n_2]] \times [[1, n_3]]. \end{aligned} \quad (13)$$

Consider a set \mathcal{R}_α , ($\alpha = a, b, c$). Suppose that $\mathcal{R} = [[n, m]] \times [[p, q]]$ is a subset of \mathcal{R}_α . The split criterion for \mathcal{R} is defined by the following procedure.

Split Criterion

Step 1. If $(d > n - m \parallel d > q - p)$ return 0.

Step 2. Select $(d + 1)^2$ points from \mathcal{R} , distributed regularly (Section 2.2.4). The result is the set \mathcal{N} .

Step 3. Consider the straight lines perpendicular to the plane containing \mathcal{R}_α , passing through each point in \mathcal{N} .

Step 4. Apply EDAS-1 to each one of these lines to obtain the number of edge points corresponding to each point $\mathbf{k} \in \mathcal{N}(\text{NEP}(\mathbf{k}))$; see Figure 2. Compute the minimum distance between these edge points ($\text{MIND}(\mathbf{k})$). In this step, all the edge points found are stored in a file.

Step 5. Compute

$$\begin{aligned} \text{MAXNEP} &= \max_{\mathbf{k} \in \mathcal{N}} \text{NEP}(\mathbf{k}), \\ \text{MIND} &= \min_{\mathbf{k} \in \mathcal{N}} \text{MIND}(\mathbf{k}). \end{aligned} \quad (14)$$

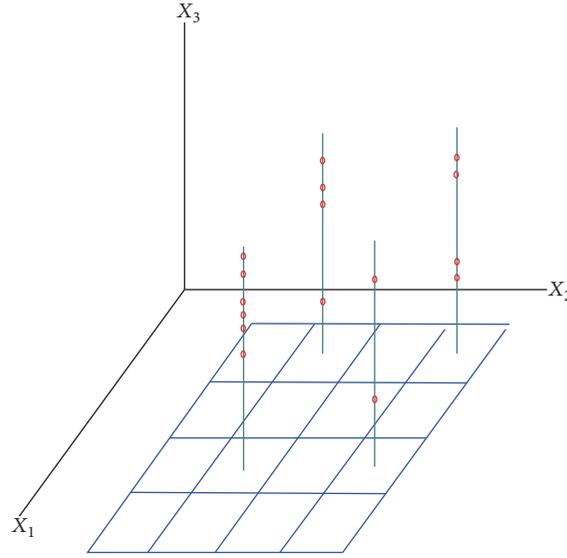


FIGURE 2: EDAS-1 obtains edge points (red) lying on the straight lines perpendicular to the considered plane.

Step 6. If we call TNEP the limit of the maximum number of edge points and TMIND the limit of the minimum distance between edge points, then

if ((MAXNEP \geq TNEP) || (MIND \leq TMIND)) return 1;
else return 0.

A discrete rectangle is subdivided when the measure of projective complexity corresponding to the discrete rectangle (PC*) is greater than a fixed beforehand limit or when the minimum distance between edge points is less than a fixed threshold. In this last case, it is possible that the discontinuity surface is nonsmooth and this makes it necessary to consider small discrete rectangles to obtain a detailed description of the jump discontinuity set. Given a discrete rectangle \mathcal{R} , the split criterion is implemented with the function SPC. SPC(\mathcal{R}) takes the value 1 if the split criterion is satisfied and 0 in the contrary case.

2.3. Doubly Adaptive Algorithm. Consider the discrete rectangle $\mathcal{R}_\alpha = [[1, n_1]] \times [[1, n_2]]$. Let \mathcal{R} be a discrete rectangle contained in \mathcal{R}_α . If SPC(\mathcal{R}) = 0, we call \mathcal{R} a good discrete rectangle; otherwise, \mathcal{R} is called bad.

Doubly Adaptive Algorithm for 3D Edge Detection (DA3DED)

Step 1. Consider $\mathcal{R}_\alpha = [[1, n_1]] \times [[1, n_2]]$. Divide \mathcal{R}_α into four discrete rectangles using the procedure described in Section 2.2.4. Apply the split criterion to the resulting rectangles. The good rectangles are put into the set \mathcal{G}_1 and the bad rectangles are put into the set \mathcal{B}_1 .

Step 2. At each step j we have a set of good discrete rectangles \mathcal{G}_j and a set of bad discrete rectangles \mathcal{B}_j . Divide each bad discrete rectangle into four discrete rectangles using

the procedure described in Section 2.2.4. Test whether these children are good or bad to obtain the sets \mathcal{G}_{j+1} and \mathcal{B}_{j+1} .

Step 3. The algorithm stops if $\mathcal{B}_j = \emptyset$. If the stopping criterion is not satisfied, go to Step 2.

Step 4. Repeat the above steps for $\mathcal{R}_b = [[1, n_1]] \times [[1, n_3]]$ and $\mathcal{R}_c = [[1, n_2]] \times [[1, n_3]]$.

DA3DED has been implemented using a quadtree. Below we detail this algorithm.

2.3.1. Algorithm to Generate the Quadtree. In this subsection, we provide a pseudocode of the algorithm used by DA3DED to generate the quadtree. We start by defining a structure of type node (each node is associated with a discrete rectangle \mathcal{R}):

```
struct node
{
    int n; int m; int p; int q;
    int spc;
    int pt;
    int ch[4];
    int d;
}Q[ ],
```

where, n , m , p , and q are the coordinates of the corners of the discrete rectangle:

spc: indicator variable that takes the value 1 if the algorithm has applied the *split criterion* to the rectangle and 0 otherwise.

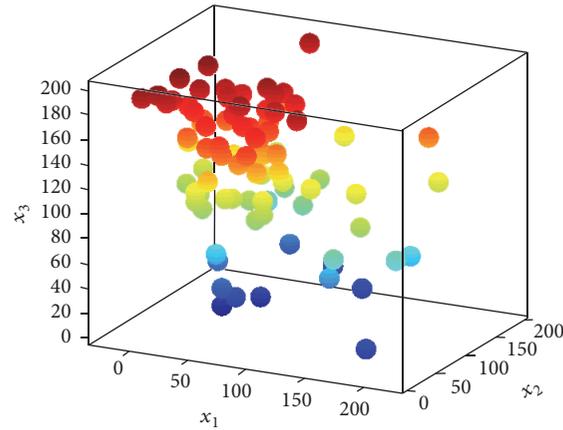
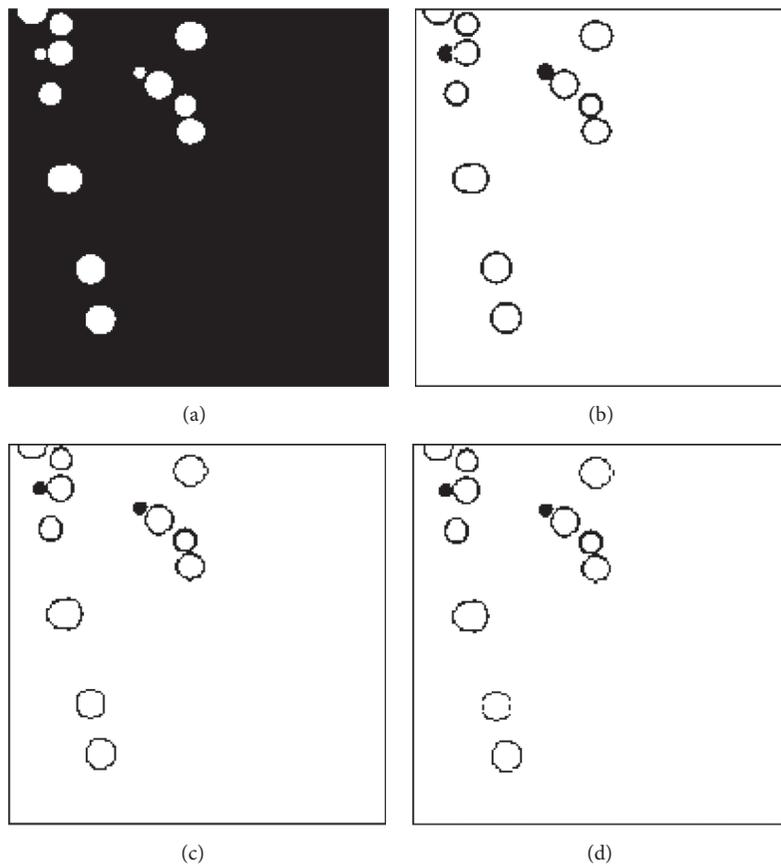


FIGURE 3: Spherical inclusions.

FIGURE 4: Spherical inclusions (intersection with the plane $x_1 = 50$). (a) Exact intersection, (b) 3D Sobel, (c) 3D EDAS-1, and (d) DA3DED.

pt: number of the node parent of the current one.

ch[j]: jth child of the current node (-1 in case of leaf nodes). ch[0] is the left child and ch[3] is the right child.

d: determines the subset of \mathcal{R} used by the split criterion.

The algorithm uses alternately two distinct values of d : d_1 and d_2 , in order to avoid the repetition of computations. In

practice, these values are almost equal. We denote by R_c the discrete rectangle associated with the current node c .

Algorithm to Generate the Quadtree. See Algorithm 1.

3. Results and Discussion

In this section, we have tested DA3DED with several synthetic material models. The results have been compared with

```

(i) Input of data:  $n_k, n_l, 1 \leq k < l \leq 3, E_1, E_2, TNEP, TDMIN, d_1, d_2, po = 0$ .
(ii) Initialize the node 0:
    Q[0].n = 1, Q[0].m =  $n_k$ ; Q[0].p = 1; Q[0].q =  $n_l$ ; Q[0].spc = 1; Q[0].pt = -1;
(iii) Split the node 0 into four discrete rectangles and initialize the nodes 1, 2, 3 and 4.
    Q[j].n, Q[j].m, Q[j].p, Q[j].q,  $j = 1, \dots, 4$ , are obtained according to the method described in Section 2.2.4.
    for ( $j = 1; j \leq 4; j++$ ) {Q[j].spc = 0; Q[j].pt = 0; Q[j].d =  $d_1$ ;}
(iv) Complete the members of the node 0,
    for ( $j = 0; j \leq 3; j++$ )Q[0].ch[j] =  $j + 1$ ;
(v) Initialize the current node and the counter of nodes  $c = 1; nm = 4$ ;
(vi) while ( $po < 4$ )
    {
    rch = Q[c].ch[3]; (right child of the current node)
    if (Q[c].spc == 0) (the split criterion has not been applied to the current node)
    {
        if (SPC( $\mathcal{R}_c$ ) == 1) (the current node satisfies the split criterion)
        {
            Split  $\mathcal{R}_c$  into four discrete rectangles,
            compute Q[nm + j].n, Q[nm + j].m, Q[nm + j].p, Q[nm + j].q,
             $j = 1, \dots, 4$ , by the method described in Section 2.2.4.
            if (Q[c].d ==  $d_1$ )DV =  $d_2$ ;
            else DV =  $d_1$ ;
            for ( $j = 1; j \leq 4; j++$ )
            {Q[nm + j].spc = 0; Q[nm + j].pt =  $c$ ;
            Q[nm + j].d = DV; Q[c].ch[j - 1] = nm + j;}
            Q[c].spc = 1;
             $c = nm + 1$ ; (new current node)
             $nm = nm + 4$ ;
        }
        else (the current node does not satisfy the split criterion)
        {
            Q[c].spc = 1;
            for ( $j = 0; j \leq 3; j++$ )Q[c].ch[j] = -1;
             $c = Q[c].pt$ ;
            if ( $c == 0$ ) $po = po + 1$ ;
        }
    }
    else (the split criterion has been applied to the current node)
    {
        if (Q[rch].spc == 1) (all the childs have been visited. Go to the parent)
        {
             $c = Q[c].pt$ ;
            if ( $c == 0$ ) $po = po + 1$ ;
        }
        else (there exist childs not visited)
        {
            for ( $j = 1; j \leq 3; j++$ ) if (Q[Q[c].ch[j]].spc == 0)break;
             $c = Q[c].ch[j]$ ;
            if ( $c == 0$ ) $po = po + 1$ ;
        }
    }
    }
}

```

ALGORITHM 1

those obtained with other algorithms: 3D Sobel, 3D Prewitt, and 3D EDAS-1. The test models have been the following:

- (i) Internal inclusions within a material
- (ii) Short fiber composites

(iii) Fracture models

Materials with internal inclusions are also called particle composites (filled materials). For example, internal pores in aluminium alloys can be obtained by introducing hydrogen in the melt metal. Since the solubility of the hydrogen decreases

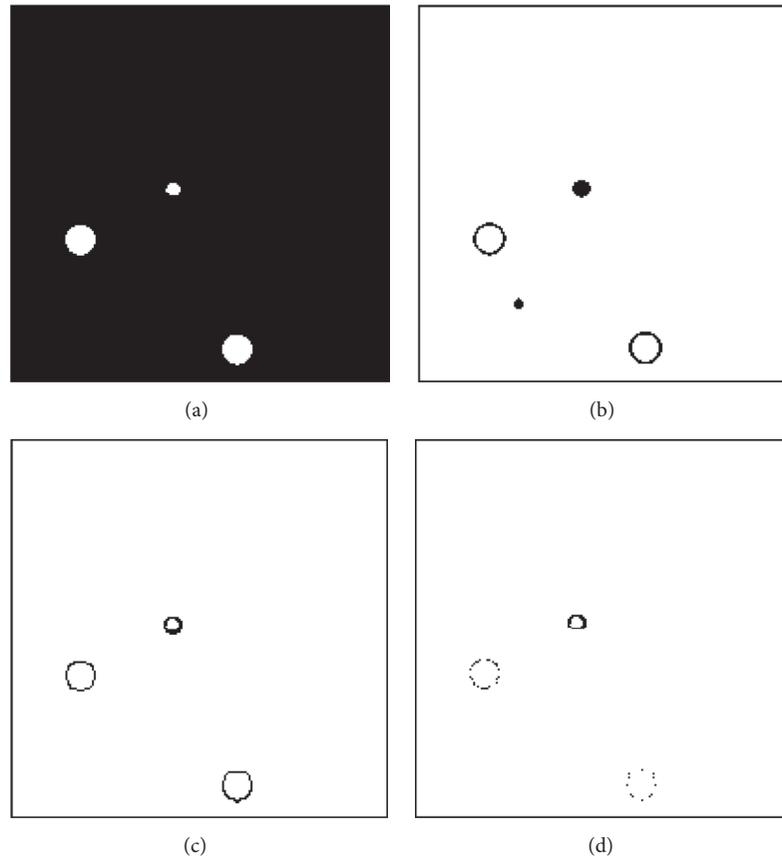


FIGURE 5: Spherical inclusions (intersection with the plane $x_1 = 150$). (a) Exact intersection, (b) 3D Sobel, (c) 3D EDAS-1, and (d) DA3DED.

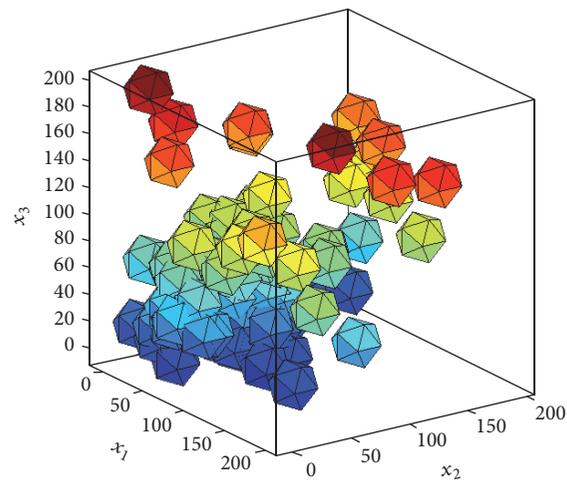


FIGURE 6: Icosahedral inclusions.

with the temperature, when the metal solidifies, the gas is rejected and forms gas pores which are easily visualized using X-ray tomography [27]. Icosahedral inclusions have been described in [28]. Short fiber composites are a kind of composite materials with short fibers of materials included in a matrix material. For example, glass or carbon fibers

are used to reinforce polymers. The habitual fibers present a circular cross section. Nowadays composite research is being conducted with more exotic fibers: hollow glass fibers and triangular glass fibers. In the case of fibers with triangular cross sections it is difficult to compute the fiber orientations from 2D images. This makes necessary 3D imaging [29]. Finally, we

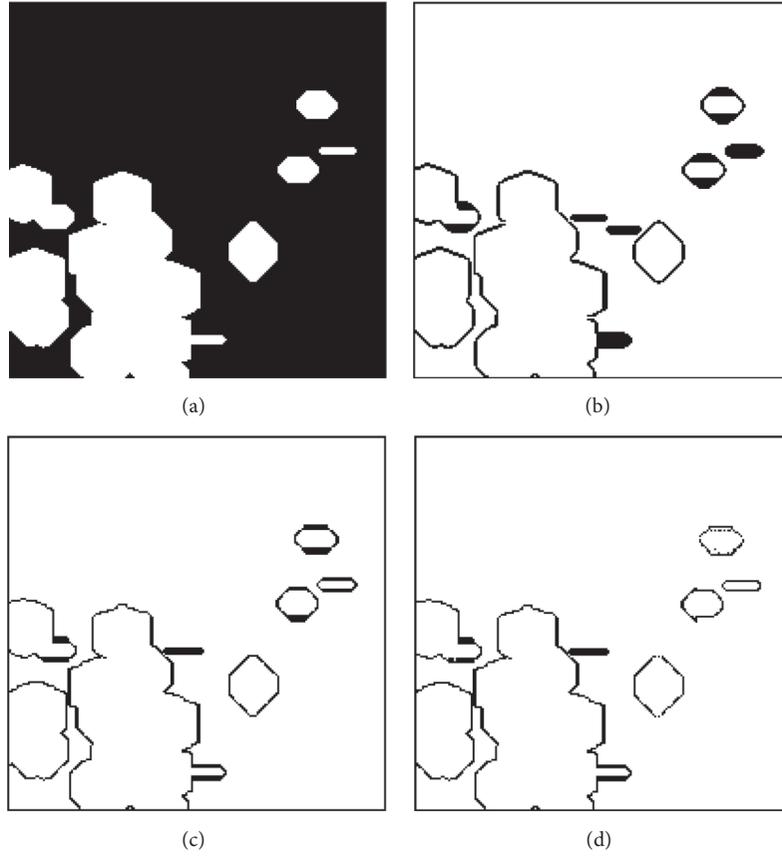


FIGURE 7: Icosahedral inclusions (intersection with the plane $x_1 = 75$). (a) Exact intersection, (b) 3D Sobel, (c) 3D EDAS-1, and (d) DA3DED.

have considered a model of fractured material. The interface of a fracture model is rather difficult to approximate because it presents acute dihedral angles. In fact, many procedures to detect 3D edges assume that the jump discontinuity set is smooth.

To test the algorithms, we have designed examples having different complexity in different zones. The box containing the composite has been divided into eight equal cubes. The inclusions and fibers have been randomly placed in each cube with the following distribution:

- (i) Spherical inclusions: 60 spheres (with radius 8) in one of the cubes, each one of the remaining parts contains 4 spheres
- (ii) Icosahedral inclusions: 60 icosahedra in one of the cubes, each one of the remaining parts contains 4 icosahedra
- (iii) Cylindrical fibers: 30 cylinders in one of the cubes, each one of the remaining parts contains 2 cylinders
- (iv) Prismatic fibers: 30 triangular prisms in one of the cubes, each one of the remaining parts contains 2 triangular prisms

The 3D images used in the experiments have been generated in two steps. First, we have considered a continuous function

with value 1 inside and 0 outside the bodies. In the case of icosahedral inclusions, prismatic fibers, and fracture model, we have used the algorithm proposed in [30] to find the distance from a point to a polyhedron. Then, the above continuous function has been discretized using a $200 \times 200 \times 200$ mesh and taking its value at the center of each cell.

The experimental environment has been the following:

- (i) CPU Intel(R) Core(TM) i7-4500U CPU 1.8 GHz.
- (ii) Running Software Microsoft Visual C++ 2012.

The test 3D images exhibit a complex structure containing several three-dimensional objects. In complex problems, it may be difficult to use deformable model algorithms. Therefore, we have compared DA3DED with 3D difference filters (Sobel, Prewitt) which are suitable for arbitrary (unstructured) images. In the experiments with the 3D Sobel method, the following mask to obtain the partial derivative of the image intensity with respect to the variable x_1 was adopted:

$$\begin{aligned}
 MX(:, :, 1) &= [-2 \ 0 \ 2; -3 \ 0 \ 3; -2 \ 0 \ 2], \\
 MX(:, :, 2) &= [-3 \ 0 \ 3; -6 \ 0 \ 6; -3 \ 0 \ 3], \\
 MX(:, :, 3) &= [-2 \ 0 \ 2; -3 \ 0 \ 3; -2 \ 0 \ 2].
 \end{aligned} \tag{15}$$

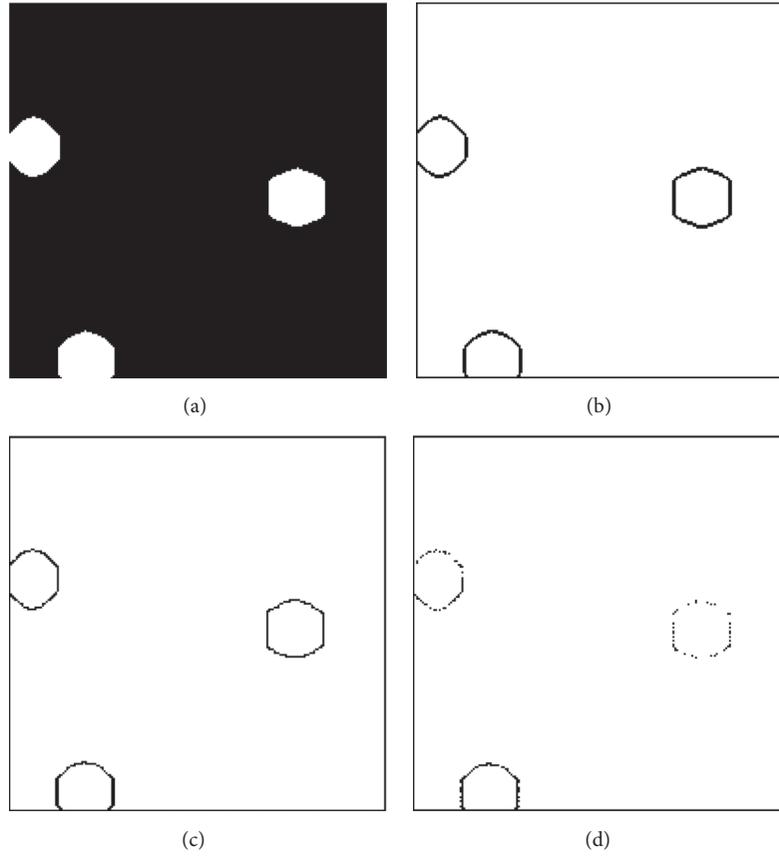


FIGURE 8: Icosahedral inclusions (intersection with the plane $x_1 = 150$). (a) Exact intersection, (b) 3D Sobel, (c) 3D EDAS-1, and (d) DA3DED.

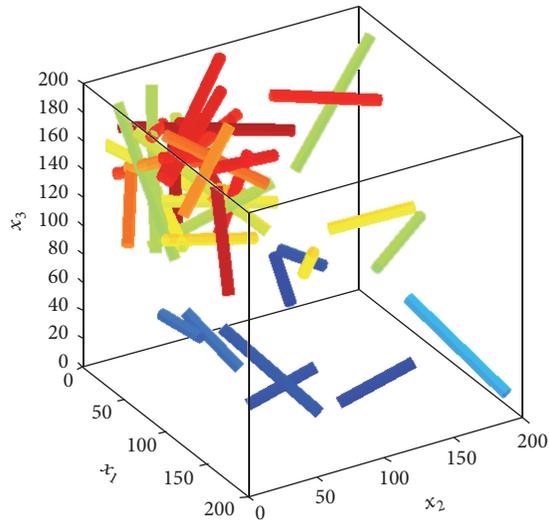


FIGURE 9: Cylindrical fibers (circular cross section).

The corresponding mask used for the 3D Prewitt method is

$$\begin{aligned} \text{MX}(:, :, 1) &= [-1 \ 0 \ 1; -1 \ 0 \ 1; -1 \ 0 \ 1], \\ \text{MX}(:, :, 2) &= [-1 \ 0 \ 1; -1 \ 0 \ 1; -1 \ 0 \ 1], \\ \text{MX}(:, :, 3) &= [-1 \ 0 \ 1; -1 \ 0 \ 1; -1 \ 0 \ 1]. \end{aligned} \quad (16)$$

We have used the MATLAB notation for matrices. The masks for the derivatives with respect to x_2 and x_3 can be obtained from the above ones [10, 31].

The results obtained with the Prewitt method are similar to those obtained with the Sobel method, we have not included them to save space.

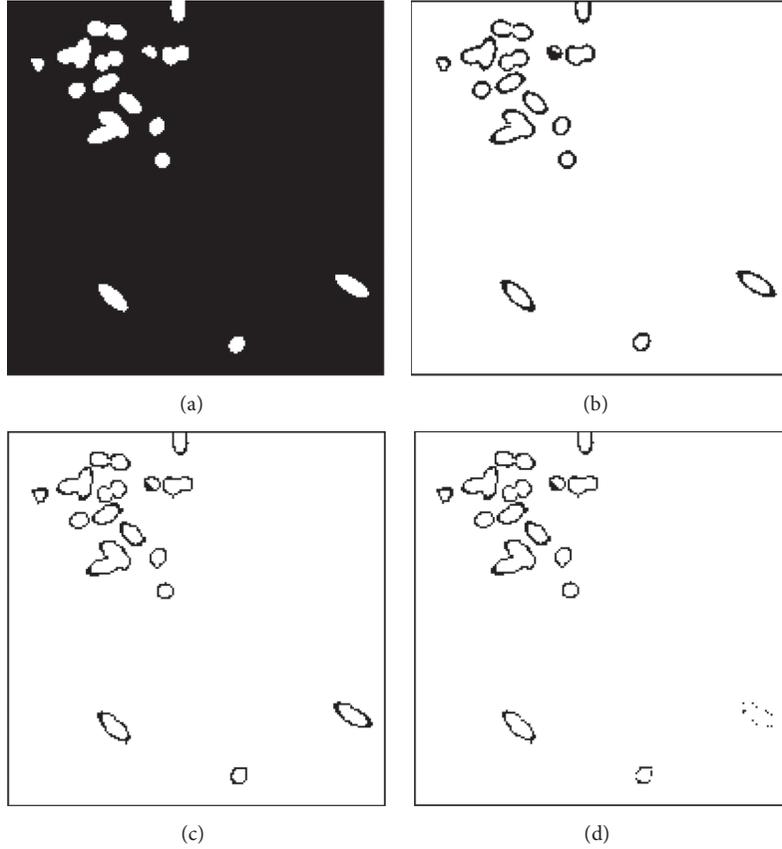


FIGURE 10: Cylindrical fibers (intersection with the plane $x_1 = 50$). (a) Exact intersection, (b) 3D Sobel, (c) 3D EDAS-1, and (d) DA3DED.

TABLE 1: Performance of 3D EDAS-1 on the test models.

3D image	E_1	E_2	E_3	E_4	E_5	TNI	TNJI	CPU (s)
Spherical inclusions	10^{-1}	10^{-2}	10^{-2}	10	10	4770465	93447	109.8
Icosahedral inclusions	10^{-1}	10^{-2}	10^{-2}	10	10	6123908	229703	156.7
Cylindrical fibers	10^{-1}	10^{-2}	10^{-2}	10	10	4854444	102482	111.8
Prismatic fibers	10^{-1}	10^{-2}	10^{-2}	10	10	4940281	112624	115.1
Fracture model	10^{-1}	10^{-2}	10^{-2}	10	10	8297400	452000	234.4

TABLE 2: Performance of DA3DED on the test models.

3D image	TMIND	TNEP	d_1	d_2	TNI	TNJI	CPU (s)
Spherical inclusions	3	25	10	9	2137515	71586	51.3
Icosahedral inclusions	3	25	10	9	4339122	192821	110.4
Cylindrical fibers	3	25	10	8	2774380	89438	65.8
Prismatic fibers	4	25	6	5	3369707	108608	79.5
Fracture model	4	17	8	7	3983960	338064	124.9

3D EDAS-1 consists of applying EDAS-1 to the straight lines, perpendicular to the plane $x_i x_j$, passing through each one of the points of $[[1, n_i]] \times [[1, n_j]]$. This process is performed for each coordinate plane.

The numerical results for 3D EDAS-1 and DA3DED are reported in Tables 1 and 2. We call TNI the total number of intervals and TNJI the total number of intervals containing jumps, generated in the applications of EDAS-1. E_5 is a

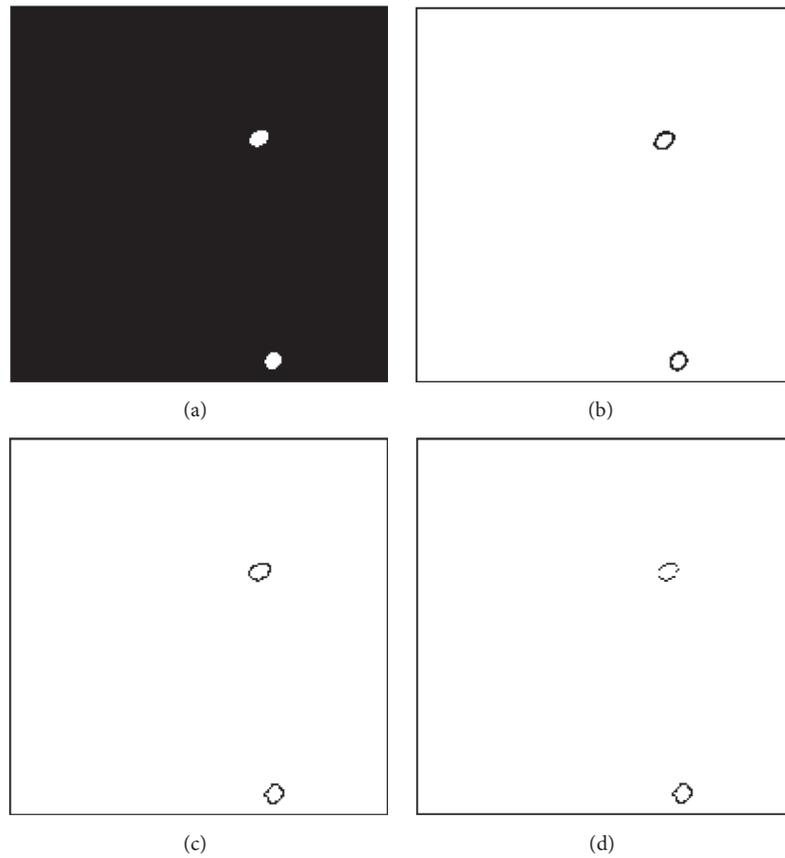


FIGURE 11: Cylindrical fibers (intersection with the plane $x_1 = 110$). (a) Exact intersection, (b) 3D Sobel, (c) 3D EDAS-1, and (d) DA3DED.

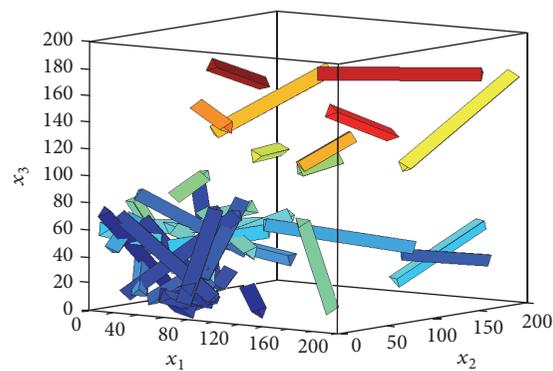


FIGURE 12: Prismatic fibers (triangular cross section).

parameter of adaptive cubature; see [22]. The results in Table 2 have been obtained with the same values of E_i , $i = 1, \dots, 5$, than those in Table 1.

The synthetic models are shown in Figures 3, 6, 9, 12, and 15. A graphic comparison of the results obtained with 3D Sobel, 3D EDAS-1, and DA3DED is shown in Figures 4, 5, 7, 8, 10, 11, 13, 14, 16, and 17.

4. Conclusions

Fast and accurate edge detection in 3D images is necessary in many applications. Some typical examples include image guided surgery, 3D assisted face recognition, safety and security applications [32], seismic imaging, and satellite images with natural color [33].

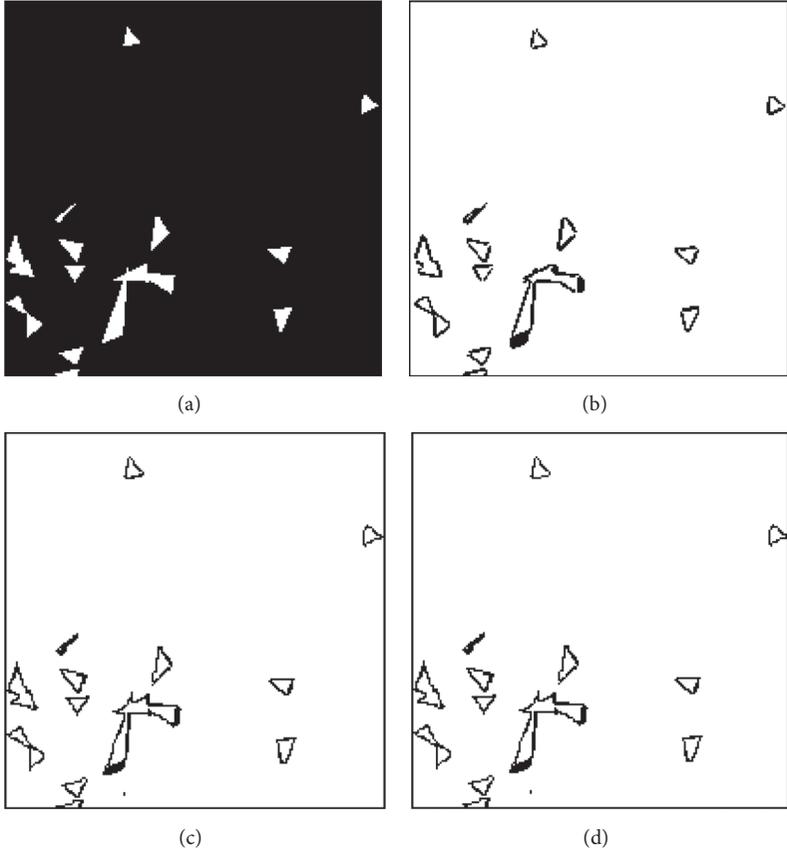


FIGURE 13: Prismatic fibers (intersection with the plane $x_1 = 50$). (a) Exact intersection, (b) 3D Sobel, (c) 3D EDAS-1, and (d) DA3DED.

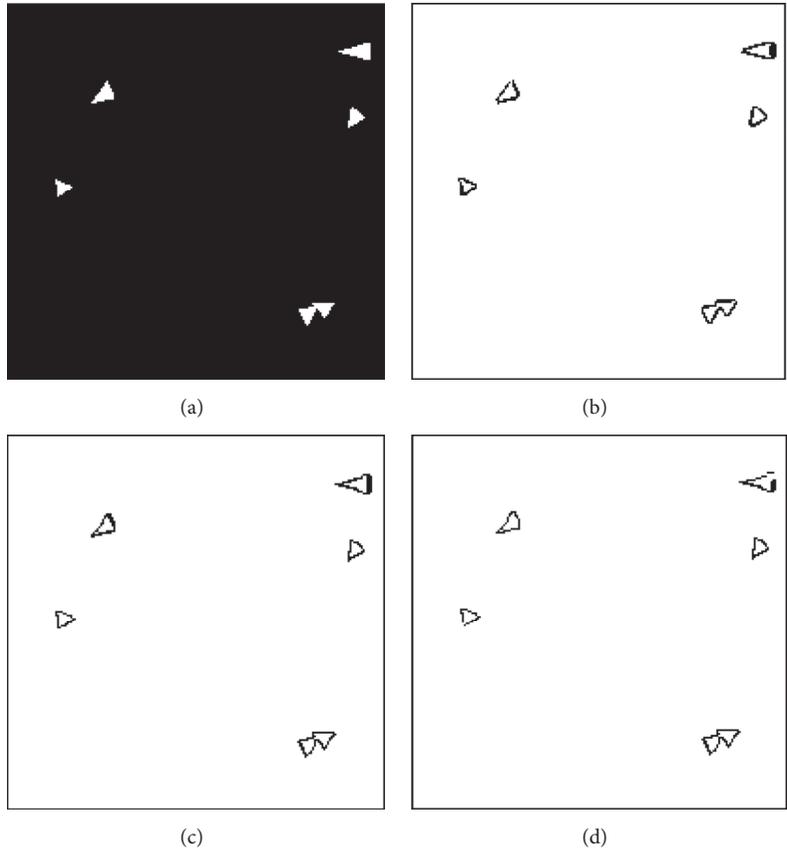


FIGURE 14: Prismatic fibers (intersection with the plane $x_1 = 150$). (a) Exact intersection, (b) 3D Sobel, (c) 3D EDAS-1, and (d) DA3DED.

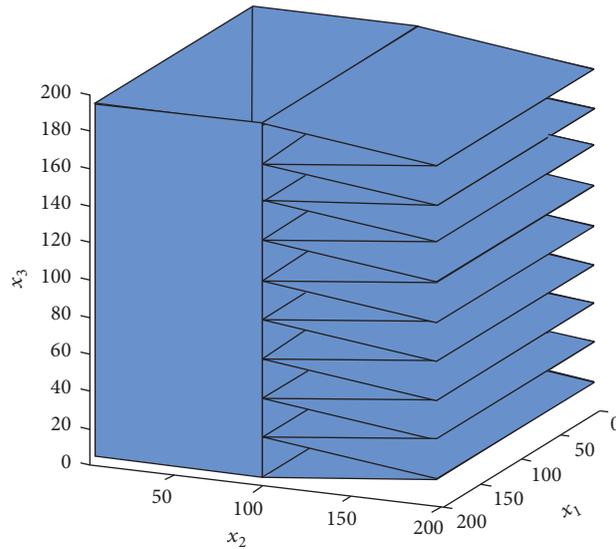
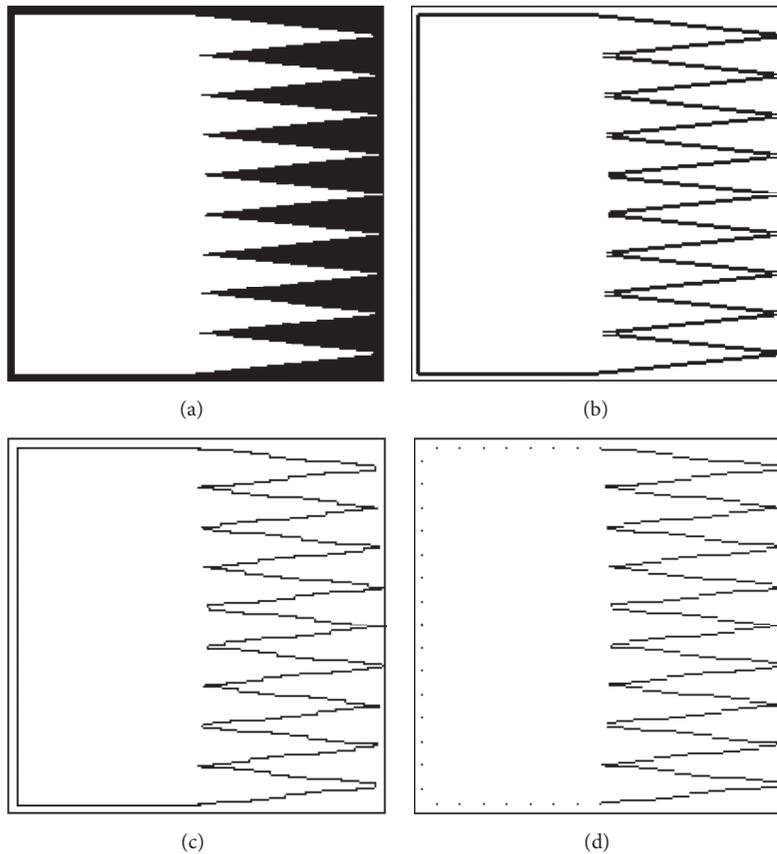


FIGURE 15: Fracture model.

FIGURE 16: Fracture model (section perpendicular to the x_1 -axis). (a) Exact intersection, (b) 3D Sobel, (c) 3D EDAS-1, and (d) DA3DED.

The problem is challenging because 3D images involve an enormous amount of data (voxels) that must be processed. Algorithms that provide a simple point based representation of the jump discontinuity set (3D filters) process all the voxels in an image.

However, in most images the complexity varies from a region to another. The proposed algorithm (DA3DED) is based on this fact. The depth of the treatment is proportional to the complexity of the region into consideration.

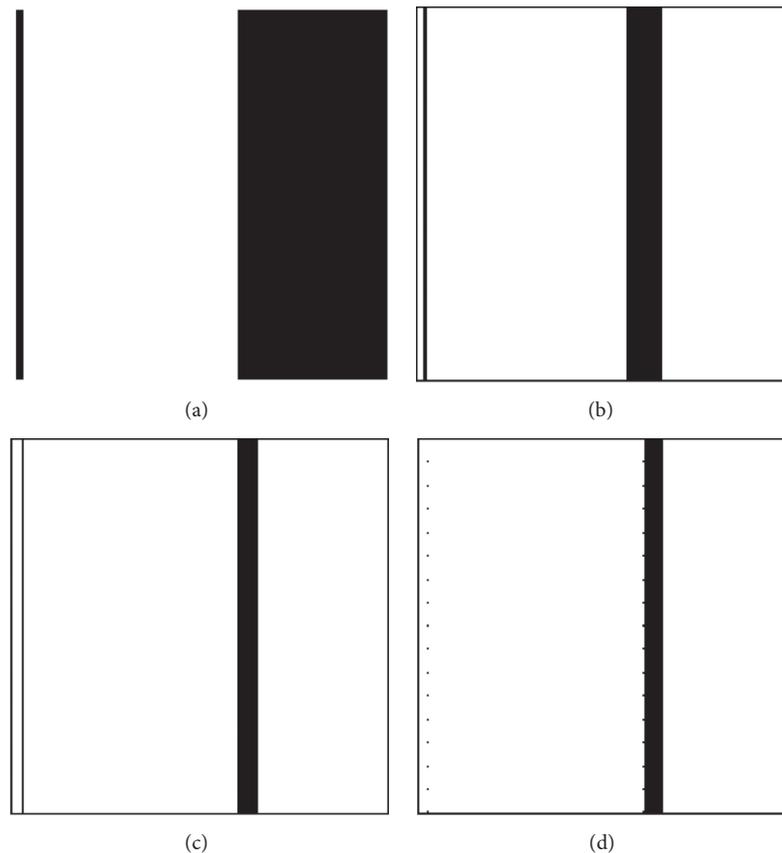


FIGURE 17: Fracture model (intersection with the plane $x_3 = 175$). (a) Exact intersection, (b) 3D Sobel, (c) 3D EDAS-1, and (d) DA3DED.

DA3DED has been tested on 3D images that modelize real problems (composites, fractures). It has been much faster than the 1D edge detection algorithm for 3D images derived from EDAS-1.

The doubly adaptive algorithm provides a thorough description of the edges in complex zones. Regions with low complexity are described in less detail, but in this case the discontinuity surface can be effectively reconstructed using techniques such as interpolation.

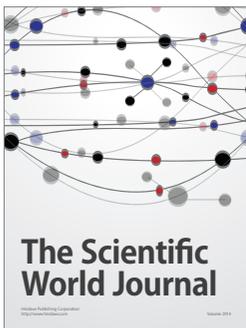
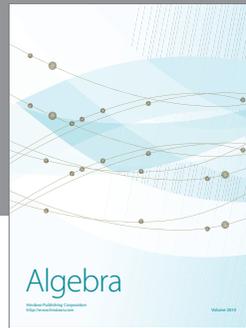
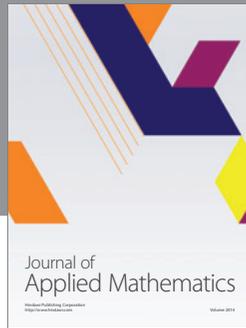
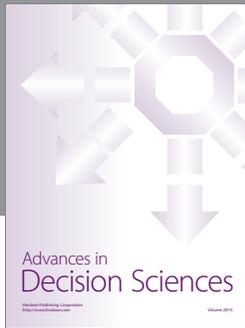
Competing Interests

The authors declare that they have no competing interests.

References

- [1] A. Bosnjak, G. Montilla, R. Villegas, and I. Jara, "3D segmentation with an application of level set-method using MRI volumes for image guided surgery," in *Proceedings of the 29th Annual International Conference of the IEEE EMBS*, pp. 5263–5266, Lyon, France, 2007.
- [2] C. Bauer, T. Pock, E. Sorantin, H. Bischof, and R. Beichel, "Segmentation of interwoven 3d tubular tree structures utilizing shape priors and graph cuts," *Medical Image Analysis*, vol. 14, no. 2, pp. 172–184, 2010.
- [3] L. Qin, H. K. Genant, J. F. Griffith, and K.-S. Leung, Eds., *Advanced Bioimaging Technologies in Assessment of the Quality of Bone and Scaffold Materials*, Springer, 2007.
- [4] J. Ohser and K. Schloditz, *3D Images of Materials Structures*, Wiley-VCH, Weinheim, Germany, 2009.
- [5] T. Kitasaka, K. Mori, J. Hasegawa, J. Toriwaki, and K. Katada, "Recognition of aorta and pulmonary artery in the mediastinum using medial-line models from 3D CT images without contrast material," *Medical Imaging Technology*, vol. 20, pp. 572–583, 2002.
- [6] K. Yokoyama, T. Kitasaka, K. Mori, Y. Mekada, J. Hasegawa, and J. Toriwaki, "Liver region extraction from 3D abdominal X-ray CT images using distribution features of abdominal organs," *Journal of Computer Aided Diagnosis of Medical Images*, vol. 7, pp. 1–11, 2003.
- [7] J. S. Suri, D. L. Wilson, and S. Laxminarayan, Eds., *Handbook of Biomedical Image Analysis, Volume III: Registration Models*, Kluwer Academic/Plenum Publishers, New York, NY, USA, 2005.
- [8] R. Archibald, K. Chen, A. Gelb, and R. Renaut, "Improving tissue segmentation of human brain MRI through preprocessing by the Gegenbauer reconstruction method," *NeuroImage*, vol. 20, no. 1, pp. 489–502, 2003.
- [9] R. Archibald, J. Hu, A. Gelb, and G. Farin, "Improving the accuracy of volumetric segmentation using pre-processing boundary detection and image reconstruction," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 459–466, 2004.
- [10] J. Toriwaki and H. Yoshida, *Fundamentals of Three-Dimensional Digital Image Processing*, Springer, Dordrecht, The Netherlands, 2009.
- [11] H. K. Liu, "Two- and three-dimensional boundary detection," *Comput Graphics Image Process*, vol. 6, no. 2, pp. 123–134, 1977.

- [12] Ch. Banisch, P. Stelldinger, and U. Kothe, "Fast and accurate 3D edge detection for surface reconstruction," in *Pattern Recognition: 31st DAGM Symposium, Jena, Germany, September 9–11, 2009. Proceedings*, J. Denzler, G. Notni, and H. Sube, Eds., vol. 5748 of *Lecture Notes in Computer Science*, pp. 111–120, Springer, Berlin, Germany, 2009.
- [13] R. Archibald, A. Gelb, and J. Yoon, "Polynomial fitting for edge detection in irregularly sampled signals and images," *SIAM Journal on Numerical Analysis*, vol. 43, no. 1, pp. 259–279, 2005.
- [14] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.
- [15] D. Terzopoulos, A. Witkin, and M. Kass, "Constraints on deformable models: recovering 3D shape and nonrigid motion," *Artificial Intelligence*, vol. 36, no. 1, pp. 91–123, 1988.
- [16] T. McInerney and D. Terzopoulos, "T-snakes: topology adaptive snakes," *Medical Image Analysis*, vol. 4, no. 2, pp. 73–91, 2000.
- [17] N. Barreira, M. G. Penedo, L. Cohen, and M. Ortega, "Topological active volumes: a topology-adaptive deformable model for volume segmentation," *Pattern Recognition*, vol. 43, no. 1, pp. 255–266, 2010.
- [18] T. Shen, H. Li, Z. Qian, and X. Huang, "Active volume models for 3D medical image segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 707–714, Miami, Fla, USA, June 2009.
- [19] J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, vol. 3, Cambridge University Press, Cambridge, UK, 2nd edition, 1999.
- [20] E. Meinhardt, E. Zacur, A. F. Frangi, and V. Caselles, "3D edge detection by selection of level surface patches," *Journal of Mathematical Imaging and Vision*, vol. 34, no. 1, pp. 1–16, 2009.
- [21] B. Llanas and S. Lantaron, "Edge detection by adaptive splitting II. The three dimensional case," *Journal of Scientific Computing*, vol. 55, pp. 474–503, 2012.
- [22] B. Llanas and S. Lantaron, "Edge detection by adaptive splitting," *Journal of Scientific Computing*, vol. 46, no. 3, pp. 485–518, 2011.
- [23] V. I. Krylov, *Approximate Calculation of Integrals*, Dover, Mineola, NY, USA, 2005.
- [24] R. A. Peters II and R. N. Strickland, "Image complexity metrics for automatic target recognizers," in *Proceedings of the Automatic Target Recognizer Systems and Technology Conference*, pp. 1–17, Naval Surface Warfare Center, 1990.
- [25] M. Cardaci, V. di Gesu, M. Petrou, and M. E. Tabacchi, "A fuzzy approach to the evaluation of image complexity," *Fuzzy Sets and Systems*, vol. 160, no. 10, pp. 1474–1484, 2009.
- [26] B. Bhanu, "Automatic target recognition: state of the art survey," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 22, no. 4, pp. 364–379, 1986.
- [27] J. Y. Buffiere, S. Savelli, and E. Maire, "Characterisation of MMCp and cast aluminium alloys," in *X-Ray Tomography in Material Science*, J. Baruchel, J.-Y. Buffiere, E. Maine, P. Merle, and G. Peix, Eds., pp. 103–113, Hermes Science Publications, Paris, France, 2000.
- [28] Z. Y. Kutikhina, L. N. Larikov, and O. A. Shamatko, "Influence of icosahedral inclusions on the thermal stability of Al–Fe–Mg system alloys," *Metal Science and Heat Treatment*, vol. 32, no. 3–4, pp. 208–210, 1990.
- [29] A. R. Clarke and C. N. Eberhardt, *Microscopy Techniques for Materials Science*, Woodhead, CRC Press, 2002.
- [30] B. Llanas, M. Fernandez de Sevilla, and V. Feliu, "An iterative algorithm for finding a nearest pair of points in two convex subsets of \mathbb{R}^n ," *Computers & Mathematics with Applications*, vol. 40, no. 8–9, pp. 971–983, 2000.
- [31] D. Wang, D. M. Doddrell, and G. Cowin, "A novel phantom and method for comprehensive 3-dimensional measurement and correction of geometric distortion in magnetic resonance imaging," *Magnetic Resonance Imaging*, vol. 22, no. 4, pp. 529–542, 2004.
- [32] A. Koschan, M. Pollefeys, and M. Abidi, Eds., *3D Imaging for Safety and Security*, vol. 35, Springer, Dordrecht, The Netherlands, 2007.
- [33] A. Gudipalli and R. Tirumala, "Comprehensive edge detection algorithm for satellite images," *World Applied Sciences Journal*, vol. 28, no. 8, pp. 1042–1047, 2013.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

