Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Full length article

Accurate subvoxel location and characterization of edges in 3D images based on the Partial Volume Effect[☆]

Agustín Trujillo-Pino [*], Miguel Alemán-Flores, Daniel Santana-Cedrés, Nelson Monzón

Centro de Tecnologías de la Imagen (CTIM), Instituto Universitario de Cibernética, Empresa y Sociedad (IUCES), Universidad de Las Palmas de Gran Canaria (ULPGC), Campus Universitario de Tafira, Las Palmas de Gran Canaria, 35017, Las Palmas, Spain

ARTICLE INFO

MSC:
62H35
68U10

Keywords:
3D images
Edge detection
Subvoxel accuracy
Partial volume effect
Feature extraction

ABSTRACT

An accurate estimation of the position, orientation, principal curvatures, and change in intensity of the edges in a 3D image provides highly useful information for many applications. The use of derivative operators to compute the gradient vector and the Hessian matrix in each voxel usually generates inaccurate results. This paper presents a new edge detector which is derived from the Partial Volume Effect (PVE). Instead of assuming continuity in the image values, edge features are extracted from the distribution of intensities within a neighborhood of each edge voxel. First, the influence of the intensities of the voxels in first- and second-order edges is analyzed to demonstrate that these types of edges can be precisely characterized from the intensity distribution. Afterward, this approach is extended to especially demanding situations by considering how adverse conditions can be tackled. This extension includes filtering noisy images, characterizing edges in blurred regions, and using windows with floating limits for close edges. The proposed technique has been tested on synthetic and real images, including some particularly difficult objects, and achieving a highly accurate subvoxel characterization of the edges. An open source implementation of our method is provided.

1. Introduction

Edge detection is a common process in a wide variety of image processing applications. In the case of 2D images, edges represent the boundaries of the objects present in the scene. In many cases, pixel-level edge detection is sufficient to determine the contours of the objects. However, in certain applications, it is preferable to find the edge curve at a subpixel level. When we deal with 3D images, such as those generated by computed tomography (CT) or magnetic resonance imaging (MRI), edges represent the surfaces of the objects. As with 2D images, an accurate identification of edge features at a subvoxel level is often necessary. These situations include medical imaging or metrology applications, where a precise measurement of certain dimensions is crucial. The main contribution of this work consists in introducing a method which can obtain edge parameters with subvoxel precision and the highest possible accuracy, both in ideal situations and in cases with severe adverse conditions.

Most methods in the literature assume that an image is the sampled version of a function $f(x, y, z)$ over a voxel grid. In order to apply differential calculus techniques on the image, it is also assumed that f is continuous and differentiable within its domain. In this way, derivative operators can be used to estimate the gradient vector and the Hessian matrix in those voxels where the intensity changes sharply. From these estimations, some parameters of the surface that crosses the voxels, such as orientation or principal curvatures, can be derived. However, the results provided by this kind of approaches are not accurate enough for demanding applications, and they cannot cope with difficult situations.

To overcome these limitations, in this paper we consider that an edge is an actual discontinuity in the function f, and the intensity of an edge voxel, where two 3D regions of different intensities coincide, has an intermediate value that depends on the proportion of the voxel covered by each region. This effect is known in the literature as the Partial Volume Effect (PVE) (see [1–3]). We propose a method to revert the PVE from the intensities within the neighborhood of an edge voxel, and obtain the surface parameters with a high accuracy.

The proposed method relies on our 2D edge detector, explained in [4], which has been extended to 3D images. Initially, we present a method to accurately detect edges in ideal images of simple geometric figures. Then, we adapt it to tackle different challenging scenarios. In order to deal with noisy images, the equations and the neighborhood are adjusted to use a previously smoothed version of the input image. Although the smoothed image no longer complies strictly with the PVE,

[☆] This paper has been recommended for acceptance by Zicheng Liu.

[*] Corresponding author.
E-mail address: agustin.trujillo@ulpgc.es (A. Trujillo-Pino).

https://doi.org/10.1016/j.jvcir.2023.103928
Received 13 January 2023; Received in revised form 28 July 2023; Accepted 12 August 2023
Available online 18 August 2023

we estimate edge features in the original image from the values in the smoothed one. In addition, a different type of neighborhood with floating limits is introduced to avoid interference between extremely close edges, so that they can be identified and treated separately.

The rest of the paper is organized as follows: Related works are outlined in Section 2. In Section 3, the error obtained with the traditional derivative operators applied to 3D images is explained. The basic version of our method for ideal images is presented in Section 4, and an improved variant for noisy images is described in Section 5. In Section 6, we deal with the problem of nearby edges, which alter the estimation of each other. Section 7 presents some experiments on synthetic images, real images of a phantom, and medical images, together with a brief discussion about the method. Finally, Section 8 summarizes the main conclusions.

2. Related work

The first edge detectors developed for 3D images relied on the computation of the gradient vector in every voxel in order to estimate the magnitude of the contrast and the orientation of the surface. Within this type of detectors, the Zucker–Hummel edge operator [5] is commonly used in medical imaging applications. In [6], Monga and Deriche propose an extension of their optimal 2D edge detector to the 3D case. In [7], Luo et al. use the 3D spatial moments to estimate the location, contrast, and orientation of the surface. A method which generalizes the Sobel operator to 3D is presented in [8]. In [9], Wang et al. propose a template-matching approach, based on the Kirsch operator for 2D images. These methods are able to obtain some edge features, but cannot provide a function that satisfactorily describes the edge.

A different kind of detectors are based on the location of zero-crossings and iso-density surfaces to obtain the edges as boundary surfaces. In [10], Zhan and Mehrotra propose a rotationally invariant 3D step edge detector by optimizing a penalty function that combines some criteria, such as signal-to-noise ratio, localization, and spurious responses. In [11], a moment-based operator which uses several edge profile models is applied to obtain the iso-density surface.

To cope with the anisotropic sampling in 3D Volume data, the authors of [12] propose a surface detection operator that uses signal processing to extract a continuous anisotropic model, instead of traditional interpolation. In [13], Brejl and Sonka propose a directional 3D edge detector for anisotropic data, interpolating the image intensity function in a small neighborhood of each voxel by a tricubic polynomial.

In order to obtain an estimation of the subvoxel position of the surface, Wu and Wee [14] compute 3D directional derivatives using facet models, and then locate the zeros of the 3D second directional derivative along the estimated gradient. In [15], another subvoxel surface detector is presented, based on Sonka's tricubic facet gradient operator, but using an adaptive threshold to improve the performance when processing images with non-uniform contrast. In [16], Bahnisch et al. present a method based on the Canny edge detector, with a subsequent phase of subvoxel refinement using B-spline-based interpolators.

Computing the curvatures of the surface is a very difficult task due to the high sensitivity when measuring local second-order derivatives in volume data. In [17], Bouma et al. propose a method to locate edges with locally constant curvature in a very accurate way by using the locally measured isophote curvature to correct the dislocation of the curved surface due to Gaussian blurring. In [18], Hiranoy and Fels propose an optimum surface fitting method to obtain the curvatures on the boundary.

To tackle the effect of noise on the images, a deep study about multiscale edge extraction approaches as well as nonlinear diffusion and wavelet approaches in 3 and higher dimensional images is presented in [19]. In [20], Schug uses shearlets and wavelets algorithms to identify edges of complex objects, especially under high noise conditions.

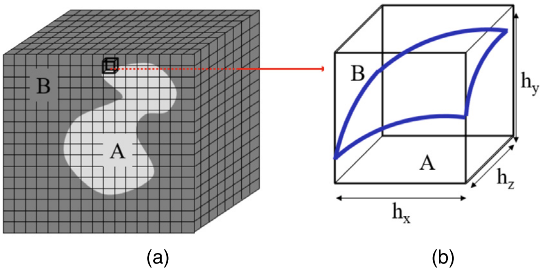


Fig. 1. Image acquisition based on the PVE: **(a)** 3D image containing an object of intensity A inside a volume of intensity B. **(b)** Detail of a contour voxel.

In metrology applications, the accuracy of the measurements is determining. In [21], the authors propose a method based on level set and non-local means methods, oriented to detect corner and edge features from noisy computed tomography (CT) images. Yagüe and Ontiveros first proposed a method to obtain subvoxel precision based on the 3D Canny detector in [22], and later a second method based on Deriche's 3D algorithm in [23]. Some recent works in metrology focus on extracting the surface with subvoxel accuracy for X-ray CT images [24,25].

Regarding medical imaging, recent works include a graph-based multiple surface segmentation method in tomography images of the eye [26], and a convolutional neural network to measure vessel wall thickness of the intracranial arteries in magnetic resonance imaging (MRI) [27].

The reconstruction of the whole surface of the objects in an image is another important area [28–33], which requires obtaining accurate data at a subvoxel level. This way, it is possible to reduce the error in the computation of the surface, particularly with those methods based on graph techniques [34,35]. Further approaches intend to locate any kind of geometric structures, which are not necessarily aligned with Cartesian axes. This task is critical within surface extraction methods (see [36]), and can clearly benefit from an estimation of edge parameters with subvoxel precision like that proposed in this paper.

3. Image acquisition based on the PVE

In this paper we consider that an edge is a discontinuity in the intensity values of a function f, and the intensity of an edge voxel between two regions depends on the proportion of the voxel covered by each region, strictly following the PVE assumption. Therefore, if an edge between two volumes of homogeneous intensities A and B crosses voxel (i, j, k), the intensity of that voxel is:

$$F_{i,j,k} = \frac{AV_A + BV_B}{h_x h_y h_z},$$

where V_A and V_B are the volumes of the regions inside that voxel with intensities A and B, respectively, and $(h_x,\ h_y,\ h_z)$ denotes the voxel size, so that $h_x h_y h_z = V_A + V_B$ (see Fig. 1). Note that we consider the possibility of dealing with non-isotropic voxels (when length, width and height are not identical).

In the following sections, we develop an algorithm to accurately extract edge features (position, orientation, curvatures, and change in intensity) in each edge voxel, assuming the acquisition model described above applies. In our approach, ideal images are those which completely comply with this assumption. This assumption is not taken into account by traditional derivative operators to compute the gradient vector and obtain the orientation of the surfaces, as described below.

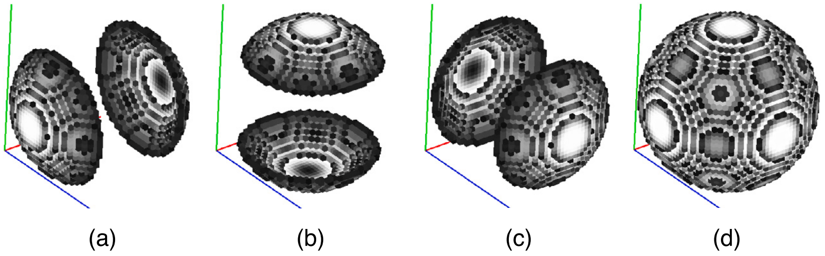


Fig. 2. Partial derivatives of the image of a solid sphere using traditional masks. In each case, the gray level of a voxel indicates: **(a)** Partial derivative in x. **(b)** Partial derivative in y. **(c)** Partial derivative in z. **(d)** Gradient norm.

3.1. *Error analysis with traditional derivative operators*

Classical edge detectors usually compute the gradient vector in each edge voxel to estimate edge parameters. This gradient vector is generally estimated by means of convolutional masks for partial derivatives, like classical Sobel or Prewitt operators for 2D images, that can be extended to 3D using 3 × 3 × 3 masks, as proposed in [37]. Nevertheless, neither the direction of the edge nor the change in intensity can be accurately obtained with these gradient estimations.

Let us consider this generic mask for the partial derivative in the z direction:

$$M_z = \frac{1}{2h_z} \times \left[\begin{bmatrix} -\beta/4 & -\alpha/4 & -\beta/4 \\ -\alpha/4 & -(1-\alpha-\beta) & -\alpha/4 \\ -\beta/4 & -\alpha/4 & -\beta/4 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} \beta/4 & \alpha/4 & \beta/4 \\ \alpha/4 & (1-\alpha-\beta) & \alpha/4 \\ \beta/4 & \alpha/4 & \beta/4 \end{bmatrix} \right].$$

The masks for the other two main directions are similar. They are an extension of the basic 2D masks to 3D, and the parameters α and β are used to adjust the contribution of the lateral rows and columns, and the eight corners. In particular, Prewitt operator uses $\alpha = \beta = 4/9$, whereas Sobel operator uses $\alpha = 8/15$ and $\beta = 4/15$. Let us consider an ideal image of a solid sphere. If the sphere and the surrounding area have homogeneous intensities, the magnitude of the gradient should be the same all across the surface of the sphere. Fig. 2 shows, in grayscale, the estimation of the partial derivatives and the norm of the gradient in each edge voxel when this type of masks are used. As observed, the result is not accurate, since the norm of the gradient varies significantly depending on the orientation of the edge, and should be constant.

Apart from computing the first order derivatives with these masks, some methods compute the second derivatives using the Laplacian operator to obtain the Hessian matrix (see [38]). Then, eigenvalues and eigenvectors of this matrix are computed in each voxel, and finally a quadratic surface is interpolated in order to estimate the subvoxel position. The problem with this technique is that continuity throughout the image is assumed, and PVE is therefore completely ignored, producing errors in the position values.

With the aim of overcoming these limitations, we present a new approach for 3D subvoxel edge characterization in the following sections. This approach provides an accurate estimation of all edge parameters, including subvoxel position, orientation, curvatures, and change in intensity.

4. Edge characterization in ideal images

In this section, we will extract those equations which allow us to characterize an edge which crosses a voxel from the values in a neighborhood of that voxel. In order to explain how these equations have been obtained, we first exemplify our approach with a first-order edge with slopes between 0 and 1. Then, a second-order edge is analyzed to extend the previous description to curved edges. Afterward, these expressions are generalized to all possible configurations depending on the slopes. Finally, some synthetically generated images are used to test and illustrate the applicability of the described method.

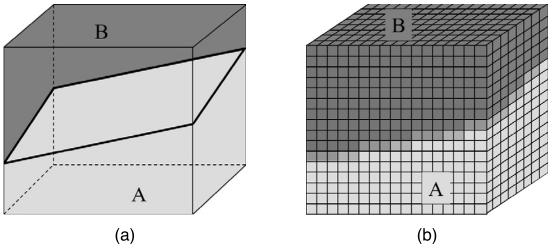


Fig. 3. Ideal image with a single edge: **(a)** 3D scene to be acquired. **(b)** Intensities in the acquired 3D image.

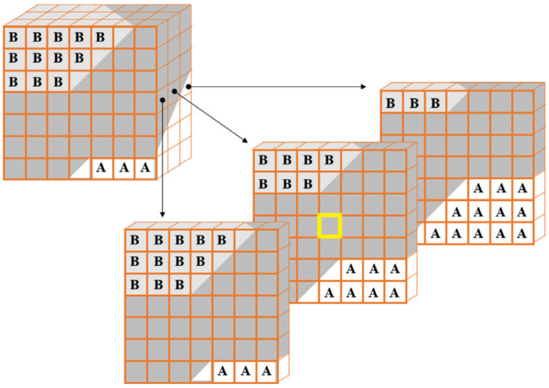


Fig. 4. First-order edge with positive slopes between 0 and 1: Voxels in the intermediate region could have values between A and B depending on the orientation of the edge. Voxel (0,0,0) is highlighted in yellow.

Let us consider an ideal 3D image F with a single edge that divides the image into two regions of intensities A and B (see Fig. 3). In this scenario, our method should be able to obtain the subvoxel position of the edge, its orientation and principal curvatures, and the change in intensity between both sides.

4.1. *First-order subvoxel edge detection for slopes between* 0 *and* 1

Let us start by considering that the edge is a plane with slopes between 0 and 1 that intersects the central voxel of the 3D image, as in Fig. 3. For simplicity, let us consider that the coordinates of this voxel are $(0, 0, 0)$. The edge is described by the function $y = a + bx + cz$, where the coordinate system is centered at the middle point of the voxel $(0, 0, 0)$. This plane divides the volume into two regions of intensities A and B. If a, b, c, A and B are unknown, the distribution of intensities in the neighboring voxels is illustrated in Fig. 4, where voxels in the intermediate region can have values between A and B.

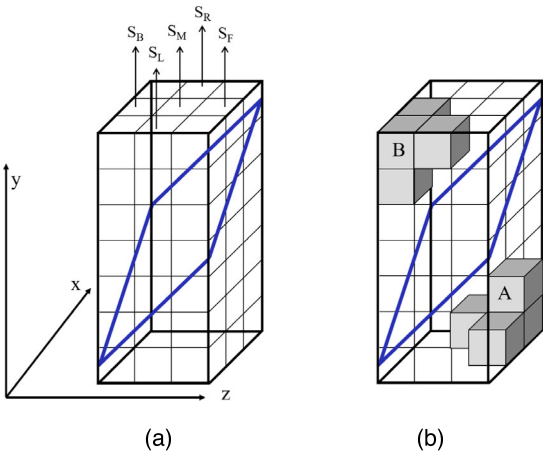


Fig. 5. Addition of the intensities within certain columns at both sides of the edge: **(a)** A 7 × 3 × 3 vertical window centered at an edge voxel ensures that the plane only intersects side walls. **(b)** 4 voxels in each corner are used to estimate intensities A and B.

Assuming the PVE applies, the intensities of the voxels follow this expression:

$$F(i,j,k) = B + \frac{A-B}{h_x h_y h_z} P_{i,j,k}, \tag{1}$$

where $P_{i,j,k}$ is the volume inside the voxel (i,j,k) under the edge plane $(0 \le P_{i,j,k} \le h_x h_y h_z)$, and h_x, h_y, h_z are the dimensions of the voxel, which are not necessarily equal.

Let us consider a $3 \times 7 \times 3$ vertical window centered at voxel $(0,0,0)$. Seven rows are needed to ensure that the edge crosses the window from left to right and from back to front, even in the worst case (slopes close to 1, as shown in Fig. 5). Let S_M, S_L, S_R, S_B and S_F be the sums of the left, middle, right, back, and front columns of the window, respectively. These sums are:

$$\begin{aligned}
S_L &= \sum_{n=-3}^{3} F_{-1,n,0} = 7B + \frac{A-B}{h_x h_y h_z} V_L; \\
S_M &= \sum_{n=-3}^{3} F_{0,n,0} = 7B + \frac{A-B}{h_x h_y h_z} V_M; \\
S_R &= \sum_{n=-3}^{3} F_{1,n,0} = 7B + \frac{A-B}{h_x h_y h_z} V_R; \\
S_B &= \sum_{n=-3}^{3} F_{0,n,-1} = 7B + \frac{A-B}{h_x h_y h_z} V_B; \\
S_F &= \sum_{n=-3}^{3} F_{0,n,1} = 7B + \frac{A-B}{h_x h_y h_z} V_F;
\end{aligned} \tag{2}$$

where V_M, V_L, V_R, V_B and V_F represent the volumes inside each column under the plane of the edge. These volumes, in turn, correspond to the following expressions:

$$\begin{aligned}
V_M &= \int_{-h_z/2}^{h_z/2} \int_{-h_x/2}^{h_x/2} \left(a + bx + cz + \frac{7}{2} h_y\right) dx\,dz = \frac{h_x h_z}{2} \left(2a + 7h_y\right) \\
V_L &= \int_{-h_z/2}^{h_z/2} \int_{-3h_x/2}^{-h_x/2} \left(a + bx + cz + \frac{7}{2} h_y\right) dx\,dz = \frac{h_x h_z}{2} \left(2a - 2bh_x + 7h_y\right) \\
V_R &= \int_{-h_z/2}^{h_z/2} \int_{h_x/2}^{3h_x/2} \left(a + bx + cz + \frac{7}{2} h_y\right) dx\,dz = \frac{h_x h_z}{2} \left(2a + 2bh_x + 7h_y\right) \\
V_B &= \int_{-3h_z/2}^{-h_z/2} \int_{-h_x/2}^{h_x/2} \left(a + bx + cz + \frac{7}{2} h_y\right) dx\,dz = \frac{h_x h_z}{2} \left(2a - 2ch_z + 7h_y\right) \\
V_F &= \int_{h_z/2}^{3h_z/2} \int_{-h_x/2}^{h_x/2} \left(a + bx + cz + \frac{7}{2} h_y\right) dx\,dz = \frac{h_x h_z}{2} \left(2a + 2ch_z + 7h_y\right)
\end{aligned} \tag{3}$$

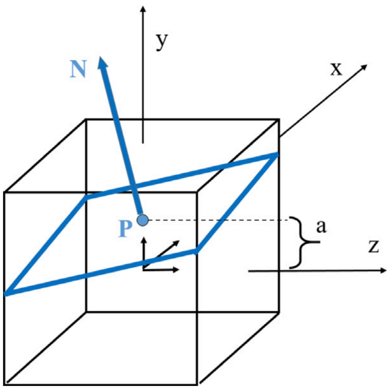


Fig. 6. Position and normal vector of the edge: The subvoxel position of an edge inside a boundary voxel is given by the point $P = (0, a, 0)$, located on the vertical of the central point. The normal vector to the surface of the edge is estimated at this point.

Using the expression for S_M, we obtain the coefficient a as follows:

$$a = \frac{2S_M - 7(A+B)}{2(A-B)} h_y. \tag{4}$$

Proceeding in a similar way with the other columns, we obtain the coefficients b and c:

$$b = \frac{S_R - S_L}{2(A-B)} \frac{h_y}{h_x}, \qquad c = \frac{S_F - S_B}{2(A-B)} \frac{h_y}{h_z}. \tag{5}$$

To estimate the intensities A and B, we can use the four voxels at the opposite corners of the window which are furthest apart from the edge (see Fig. 5**(b)**):

$$\begin{aligned}
A &= \frac{1}{4}\left(F_{1,3,1} + F_{1,3,0} + F_{0,3,1} + F_{1,2,1}\right) \quad \text{and} \\
B &= \frac{1}{4}\left(F_{-1,-3,-1} + F_{-1,-3,0} + F_{0,-3,-1} + F_{-1,-2,-1}\right).
\end{aligned} \tag{6}$$

Once we know the coefficients of the plane, we can extract the edge features. We consider that subvoxel edge position is given by the point of the plane located on the vertical of the voxel center, i.e., the point $P = (0, a, 0)$, where a is the vertical distance from the center of the voxel to the plane, as illustrated in Fig. 6.

The vector normal to the edge at that point is given by:

$$N = \frac{A-B}{\sqrt{1 + b^2 + c^2}} \left[b, -1, c\right],$$

and the norm of this vector indicates the change in intensity between both sides of the edge.

4.2. *Second-order subvoxel edge detection for slopes between* 0 *and* 1

The location and characterization of plane edges (first-order) provides satisfactory results in most cases. However, some scenarios require a more accurate edge characterization, which includes an estimation of the curvatures of the edges. In order to estimate these magnitudes, we approximate the edge by a second-order surface, $y = f(x,z) = a + bx + cz + dx^2 + fxz + gz^2$.

Let us assume again that the slopes on the middle vertical axis of the central voxel of the window are between 0 and 1. We keep the same window size as in the previous subsection, although, in this case, we cannot assure that the edge will always cross the window through the lateral sides. To obtain the volumes inside each column under the edge surface, we compute the equations in (3), but using the second-order surface instead. The final expressions are as follows:

$$\begin{aligned}
V_M &= \frac{h_x h_z}{12} \left(12a + dh_x^2 + gh_z^2 + 42h_y\right); \\
V_L &= \frac{h_x h_z}{12} \left(12a - 12bh_x + 13dh_x^2 + gh_z^2 + 42h_y\right);
\end{aligned}$$

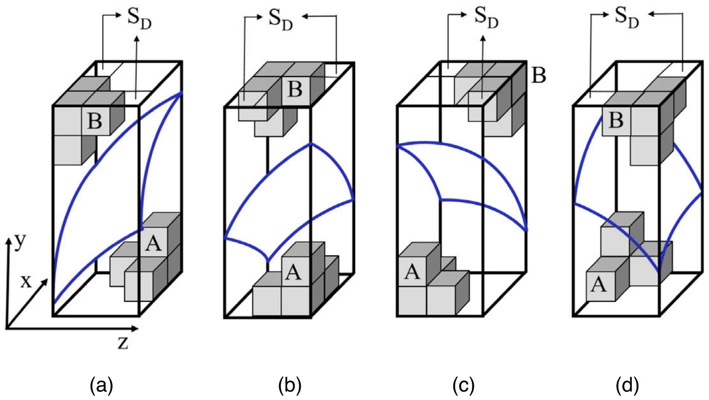


Fig. 7. Voxels selected for the estimation of intensities A and B, and columns chosen to obtain S_D depending on the sign of the partial derivatives: **(a)** $\alpha, \beta < 0$. **(b)** $\alpha > 0$ and $\beta < 0$. **(c)** $\alpha, \beta > 0$. **(d)** $\alpha < 0$ and $\beta > 0$.

$$V_R = \frac{h_x h_z}{12}\left(12a + 12bh_x + 13dh_x^2 + gh_z^2 + 42h_y\right);$$
$$V_B = \frac{h_x h_z}{12}\left(12a - 12ch_z + dh_x^2 + 13gh_z^2 + 42h_y\right);$$
$$V_F = \frac{h_x h_z}{12}\left(12a + 12ch_z + dh_x^2 + 13gh_z^2 + 42h_y\right).$$

Proceeding as in the previous subsection, we compute the sum of the five central columns S_M, S_L, S_R, S_B and S_F. However, we need one more equation in order to estimate all six coefficients of the surface. We could make use of the four columns at the corners, but we discard the two columns where the surface is furthest away from the central voxel to avoid cases with very high slopes, since the surface might cross the bottom or top of the window (the selected diagonal columns are labeled as S_D in Fig. 7). Therefore, our sixth equation is as follows:

$$S_D = \sum_{n=-3}^{3} F_{-1,n,1} + \sum_{n=-3}^{3} F_{1,n,-1} = 14B + \frac{A-B}{h_x h_y h_z} V_D, \tag{7}$$

where V_D represents the sum of the volumes inside the two selected diagonal columns, whose expression is the following:

$$\begin{aligned} V_D = &\int_{h_z/2}^{3h_z/2} \int_{-3h_x/2}^{-h_x/2} \left(a + bx + cz + dx^2 + fxz + gz^2 + \frac{7}{2}h_y\right) dx\,dz \\ &+ \int_{-3h_z/2}^{-h_z/2} \int_{h_x/2}^{3h_x/2} \left(a + bx + cz + dx^2 + fxz + gz^2 + \frac{7}{2}h_y\right) dx\,dz \\ = &\frac{h_x h_z}{6}\left(12a + 13dh_x^2 - 12fh_x h_z + 13gh_z^2 + 42h_y\right) \end{aligned}$$

Using the five equations in (2) and Eq. (7), we can obtain the values for all the coefficients of the surface as follows:

$$\begin{aligned} a &= \frac{I}{12}(28S_M - S_R - S_B - S_F - S_L - 84(A+B)); \\ b &= \frac{I}{h_x}(S_R - S_L); \qquad c = \frac{I}{h_z}(S_F - S_B); \\ d &= \frac{I}{h_x^2}(S_R + S_L - 2 * S_M); \\ f &= \frac{I}{h_x h_z}(S_R + S_L + S_F + S_B - S_D - 2 * S_M); \\ g &= \frac{I}{h_z^2}(S_F + S_B - 2 * S_M), \end{aligned} \tag{8}$$

where $I = h_y/2(A - B)$.

We remark that the first-order detection is a particular case of these new expressions, when d, f and g are equal to zero. In some real cases, first-order estimation might provide more accurate values, since it is less sensitive to certain artifacts or noise. Taking into account that a can be expressed as:

$$a = I\left(2S_M - 7(A+B)\right) - \frac{dh_x^2 + gh_z^2}{12},$$

we obtain the same expressions for a, b, c as in the first-order method if we set $d = f = g = 0$ (see Eqs. (4)–(5)).

Using these six coefficients, the subvoxel position, the change in intensity, and the normal vector, computed on the vertical center line of the voxel, are obtained using the same expressions as in the linear case. To estimate the principal curvatures and their associated vectors, we use the equations described in Appendix A, included as supplementary material.

4.3. Generalization to all orientations

In order to extend this method to locate edges with any orientation, we must previously detect through what voxels the edge crosses, and which is its main orientation (x, y or z) inside each voxel. The simplest way consists in computing an estimation of the gradient in every voxel using traditional derivative masks. Although we have previously argued that these masks do not provide an exact value, they give us a first approximation of the orientation of the surface.

Let F_x, F_y and F_z be the results of applying the partial derivative masks in the x, y and z directions, respectively. Let us first consider the case where $|F_y|h_y > |F_x|h_x, |F_z|h_z$. This case includes the one presented in the previous subsection. The five central sums are computed using the equations in (2). Nevertheless, for the diagonal sum S_D, Eq. (7) must be rewritten in the following way:

$$S_D = \sum_{n=-3}^{3} F_{-1,n,\gamma} + \sum_{n=-3}^{3} F_{1,n,-\gamma} = 14B + \frac{A-B}{h_x h_y h_z} V_D,$$

where $\gamma = \operatorname{sign}(F_x F_z)$. This allows avoiding the two columns containing the edge voxels which are furthest from the central voxel (see Fig. 7).

To estimate the intensities on both sides of the edge, A and B, we must also adapt the equations in (6) in the following way:

$$A = \frac{1}{4}(F_{\alpha,3,\beta} + F_{\alpha,3,0} + F_{0,3,\beta} + F_{\alpha,2,\beta}) \quad \text{and}$$
$$B = \frac{1}{4}(F_{-\alpha,-3,-\beta} + F_{-\alpha,-3,0} + F_{0,-3,-\beta} + F_{-\alpha,-2,-\beta}),$$

where $\alpha = \operatorname{sign}(F_x F_y)$ and $\beta = \operatorname{sign}(F_z F_y)$.

The expressions for the six coefficients of the surface are the same as in Eq. (8), except for f, which is:

$$f = \frac{\gamma I}{h_x h_z}(S_R + S_L + S_F + S_B - S_D - 2 * S_M). \tag{9}$$

Once the coefficients of the edge surface have been obtained, the subvoxel position, the normal vector, and the principal curvatures and directions are computed as in the previous subsection.

In case $F_y < 0$, the sign of the curvatures must be inverted. The reason is that this sign depends not only on the equation of the surface, but also on whether the brighter region (i.e., with a higher intensity) is concave or convex. For instance, given a solid sphere with intensities

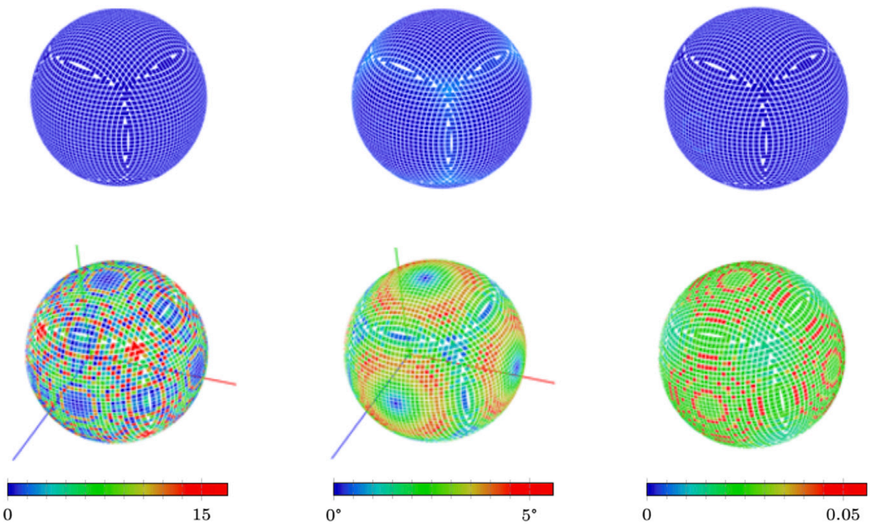


Fig. 8. Comparison of the errors in the estimation of the change in intensity (left), orientation (middle) and minimum curvature (right) on a solid sphere when using the proposed method (top) and the traditional masks (bottom). Blue indicates low errors and red corresponds to high errors.

Table 1
Results obtained with each method when detecting the edge features of a solid sphere of radius 20.

Method	Error						Curvature			
	Intensity		Orient. (ž)		Pos. (%)		Min.		Max.	
	av.	max.	av.	max.	av.	max.	av.	rad.	av.	rad.
Traditional operators	6.98	21.8	2.98	4.85	–	–	0.020	51.3	0.040	24.8
Proposed (First-order)	0.00	0.00	0.15	0.53	0.65	1.50	–	–	–	–
Proposed (Second-order)	0.00	0.00	0.15	0.53	0.25	0.83	0.047	21.1	0.053	19.0

B inside and A outside, where $A > B$, the curvature should be positive for every point on the edge. Therefore, the maximum and minimum curvatures, as well as the principal orientations, must be interchanged when $F_y < 0$ in order to obtain a coherent result.

For the other two scenarios, where $|F_y|h_y$ is not the greatest one, we must proceed in a similar way, but interchanging the variables. For example, if $|F_x|h_x > |F_y|h_y, |F_z|h_z$, the window must be oriented along the x axis, the sums must be computed along the rows instead of the columns, and the variables x, y and z must be exchanged.

4.4. Synthetic examples

In ideal images with planar edges, our method is capable of accurately computing all the parameters. To illustrate how our method overcomes the limitations of traditional approaches, we present the results of applying the detector to ideal images of two different non-planar objects, and compare them with the values obtained using classical derivative operators. Fig. 8 shows a solid sphere of radius 20 voxels with intensities 0 inside and 255 outside, and the result of applying both methods. With the aim of visualizing the results in a suitable way, a small square for every edge voxel has been drawn, centered at the subvoxel position that has been obtained, with the orientation obtained for the edge in that voxel, and using a colormap to illustrate the magnitude of the error. As observed, the accuracy of our method is significantly higher for all the parameters (all voxels are bluish with the proposed method, which means the error is low, whereas numerous greenish and reddish voxels are obtained with the traditional method, which means there are medium and high errors).

Table 1 shows the numerical errors in the estimation of intensity, orientation (degrees), subvoxel position (measured as a percentage over the length of a voxel), as well as the values for the curvatures (average curvature and curvature radii). We remark that the errors with the second-order method are very low when compared with traditional masks. There is no error in the estimation of the change in intensity; the maximum error in the orientation is approximately 0.5 degrees; the maximum error in the subvoxel position is lower than 1 % of the voxel length; and the curvature radii are very close to 20 voxels.

The second example is a solid torus with major radius $R = 30$ and minor radius $r = 10$ voxels. This geometric object has constant maximum curvature $k_{max} = 1/r$ on all its surface, whereas the minimum curvature k_{min} varies from $-1/(R-r)$ at the closest points to the center, to $1/(R+r)$ at the furthest points. The estimations for the minimum and maximum curvatures are shown in Fig. 9. Fig. 9**(a)** represents the error obtained with the proposed method in the computation of the maximum curvature, which is very low compared with the traditional method, shown in Fig. 9**(b)** (blue voxels represent lower errors). Figs. 9**(c)** and Fig. 9**(d)** represent the value obtained for the minimum curvature with both methods (in this case, blue voxels indicate a lower curvature, while red voxels correspond to a higher curvature). The expected values vary from $-1/20$ on the inner side of the torus to $1/40$ on the outer side. This is coherent with the gradual change from bluish to reddish values which has been obtained with our method (Fig. 9**(c)**). However, the values obtained with the traditional operators differ significantly from the expected ones (Fig. 9**(d)**).

In Fig. 10, we can see the principal directions obtained for the torus. The small segments in each voxel indicate the directions obtained for the minimum and maximum curvatures. Minimum and maximum curvature directions are very accurate across the surface using the proposed method. Not only do they vary gradually and smoothly following the shape of the torus, but they also correspond with the expected directions (Fig. 10 top). This does not happen with the traditional masks, which generate inaccurate and oscillating directions (Fig. 10 bottom).

5. Location of edges in noisy images

The acquisition process frequently adds some noise to the image, which translates into small variations in the intensity values. The

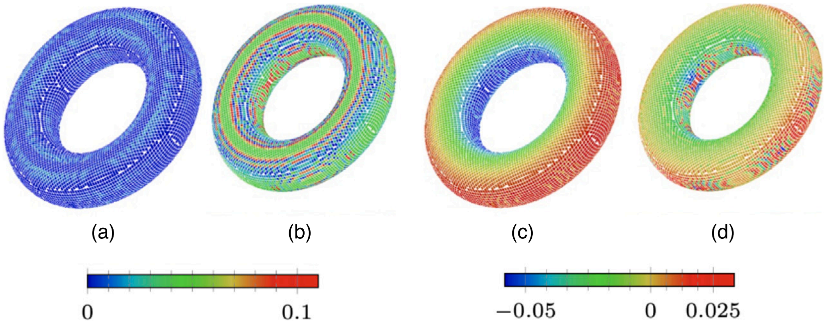


Fig. 9. Estimation of curvatures: Error in the maximum curvature when using the proposed method **(a)** and with the traditional mask **(b)**. Value of the minimum curvature (ranging from $-1/20$ to $1/40$) when using the proposed method **(c)** and with the traditional masks **(d)**.

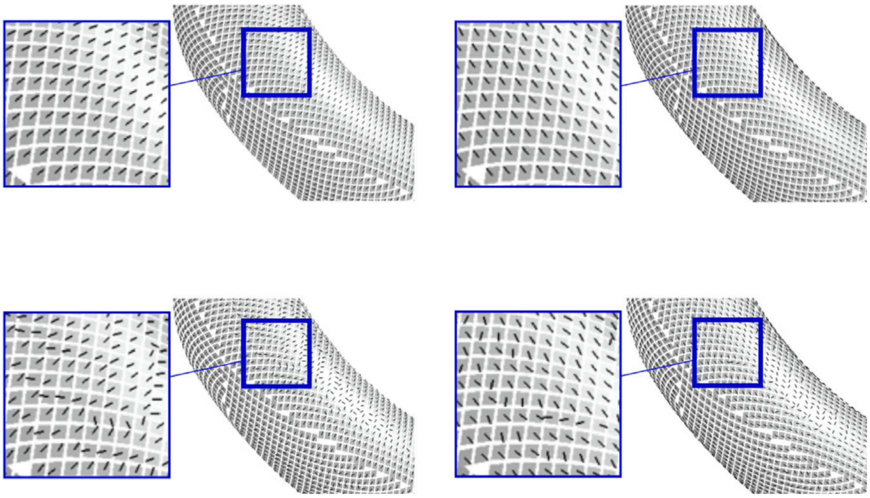


Fig. 10. Detection of maximum (left) and minimum (right) curvature directions on a torus, using the proposed method (top), and using traditional masks (bottom).

traditional way to reduce noise consists in convolving the image with a smoothing mask, such as a Gaussian kernel. The simplest 3D masks are as follows:

$$H = \left[\begin{bmatrix} a_3 & a_2 & a_3 \\ a_2 & a_1 & a_2 \\ a_3 & a_2 & a_3 \end{bmatrix}, \begin{bmatrix} a_2 & a_1 & a_2 \\ a_1 & a_0 & a_1 \\ a_2 & a_1 & a_2 \end{bmatrix}, \begin{bmatrix} a_3 & a_2 & a_3 \\ a_2 & a_1 & a_2 \\ a_3 & a_2 & a_3 \end{bmatrix} \right], \tag{10}$$

where the following two conditions are met:

$$a_0 \geq a_1 \geq a_2 \geq a_3 \quad \text{and} \quad 1 = a_0 + 6a_1 + 12a_2 + 8a_3.$$

Let F be the ideal image in Fig. 3, with a single plane edge that crosses the central voxel. Let $G = F * H$ be the resulting smoothed image, after convolving F with the smoothing mask H. We intend to obtain the edge features in F from the values in G. Nevertheless, the region of voxels with intermediate values between A and B is larger after the smoothing process. For this reason, we must use larger windows ($3 \times 11 \times 3$) centered at every edge voxel to estimate edge parameters properly.

5.1. Edge characterization after smoothing

Let us suppose that the slopes inside an edge voxel are between 0 and 1, and G_x, G_y and G_z are the partial derivatives computed on the smoothed image G. Therefore, we assume that $|G_y|h_y > |G_x|h_x, |G_z|h_z$ at that voxel.

Proceeding in the same way as in Section 4.2, we must define six different sums of columns inside the window centered at each edge voxel in order to obtain the coefficients of the second-order surface. We know that the intensity of an edge voxel is given by Eq. (1). Let $M_{i,m,n,k}$ be the sum of the column of voxels in F from voxel (i,m,k) to voxel (i,n,k). This sum is given by the following expression:

$$M_{i,m,n,k} = \sum_{j=m}^{n} F_{i,j,k} = (A - B)\frac{V_{i,m,n,k}}{h_x h_y h_z} + B\,(n - m + 1)\,,$$

where $V_{i,m,n,k}$ represents the volume inside the column under the edge surface, $y = f(x,z) = a + bx + cz + dx^2 + fxz + gz^2$, whose expression is as follows:

$$\begin{aligned} V_{i,m,n,k} &= \int_{(i-1/2)h_x}^{(i+1/2)h_x} \int_{(k-1/2)h_z}^{(k+1/2)h_z} \left(f(x,z) - \frac{2m-1}{2} h_y \right) dz\,dx = \\ &= a h_x h_z + b i h_x^2 h_z + c k h_x h_z^2 + d h_x^3 h_z \left(i^2 + \frac{1}{12} \right) + \\ &\quad + f i k h_x^2 h_z^2 + g h_x h_z^3 \left(k^2 + \frac{1}{12} \right) - \frac{2m-1}{2} h_x h_y h_z, \end{aligned}$$

as long as the surface intersects neither the bottom nor the top of the column.

Let $S_{i,m,n,k}$ be the sum of the column of voxels in the smoothed image G from voxel (i,m,k) to voxel (i,n,k), whose expression is as follows:

$$\begin{aligned} S_{i,m,n,k} = \sum_{j=m}^{n} G_{i,j,k} &= a_0 M_{i,m,n,k} \\ &+ a_1 \left(M_{i-1,m,n,k} + M_{i,m,n,k-1} + M_{i,m-1,n+1,k} + M_{i,m+1,n-1,k} \right) + \\ &+ a_1 \left(M_{i,m,n,k+1} + M_{i+1,m,n,k} \right) \\ &+ a_2 \left(M_{i-1,m,n,k-1} + M_{i-1,m-1,n+1,k} + M_{i-1,m+1,n-1,k} \right) + \\ &+ a_2 \left(M_{i-1,m,n,k+1} + M_{i,m-1,n+1,k-1} + M_{i,m+1,n-1,k-1} \right) \\ &+ a_2 \left(M_{i,m-1,n+1,k+1} + M_{i,m+1,n-1,k+1} + M_{i+1,m,n,k-1} \right) + \\ &+ a_2 \left(M_{i+1,m-1,n+1,k} + M_{i+1,m+1,n-1,k} + M_{i+1,m,n,k+1} \right) \end{aligned}$$

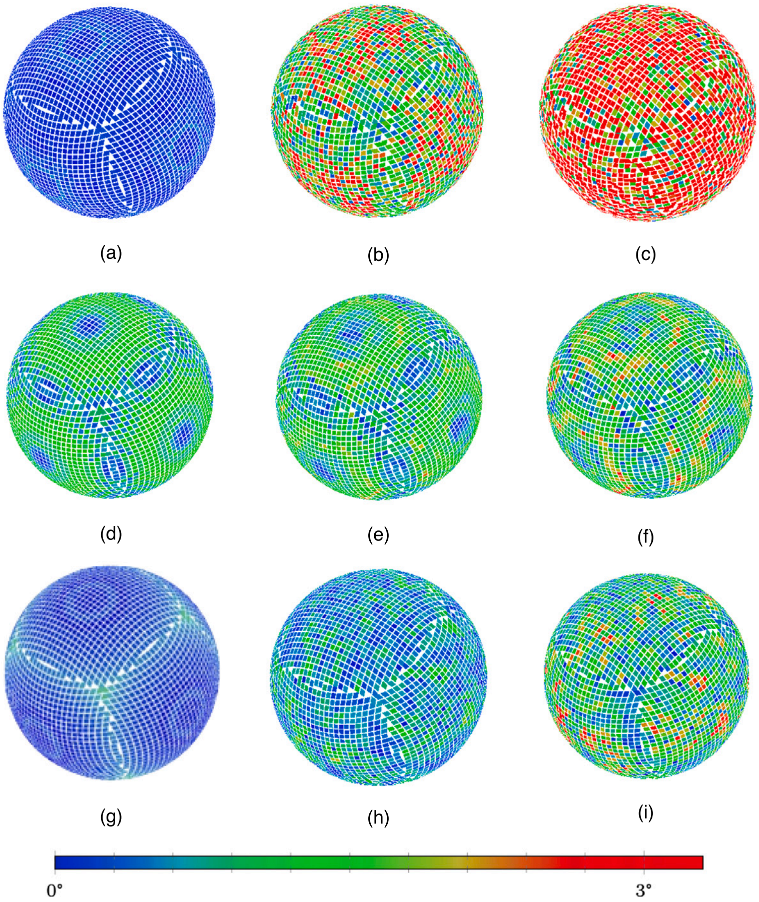


Fig. 11. Errors in the detection of orientation on a solid sphere of radius 20 with noise added (from left to right: $\sigma = 0$, 5, 10). From top to bottom: method proposed in Section 4.1 applied to the original noisy image F; traditional method applied on the smoothed image G; smoothing-based method applied on the smoothed images.

$$
\begin{aligned}
&+ a_3 \left(M_{i-1,m-1,n+1,k-1} + M_{i-1,m+1,n-1,k-1} + M_{i-1,m-1,n+1,k+1} \right) + \\
&+ a_3 \left(M_{i-1,m+1,n-1,k+1} + M_{i+1,m-1,n+1,k-1} + M_{i+1,m+1,n-1,k-1} \right) \\
&+ a_3 \left(M_{i+1,m-1,n+1,k+1} + M_{i+1,m+1,n-1,t+1} \right).
\end{aligned}
$$

From the previous expressions, we can define a six-equation linear system for the edge voxel $(0,0,0)$ as follows:

$$
\begin{aligned}
&S_M = S_{0,-5,5,0}; \qquad S_L = S_{-1,-5,5,0}; \qquad S_R = S_{1,-5,5,0}; \\
&S_B = S_{0,-5,5,-1}; \qquad S_F = S_{0,-5-,5,1}; \qquad S_D = S_{-1,-5,5,\gamma} + S_{1,-5,5,-\gamma};
\end{aligned}
\tag{11}
$$

where $\gamma = \operatorname{sign}(G_x G_z)$.

We also need to estimate the intensities on both sides of the edge. We proceed in the same way as in Section 4.3, select the appropriate corners depending on the orientation of the surface, and make use of the following equations:

$$
A = \frac{1}{4}(G_{\alpha,5,\beta} + G_{\alpha,5,0} + G_{0,5,\beta} + G_{\alpha,4,\beta}) \quad \text{and}
$$

$$
B = \frac{1}{4}(G_{-\alpha,-5,-\beta} + G_{-\alpha,-5,0} + G_{0,-5,-\beta} + G_{-\alpha,-4,-\beta}),
$$

where $\alpha = \operatorname{sign}(G_x G_y)$ and $\beta = \operatorname{sign}(G_z G_y)$.

Finally, solving the previous equation system, we obtain the same expressions for the coefficients of the surface that were obtained in Eq. (8), except for the coefficient f, that uses the generalized Eq. (9), and for the coefficient a, whose expression is the following one:

$$
a = I\left(2S_M - 11(A + B)\right) - k(dh_x^2 + gh_z^2),
$$

where k is a constant that depends on the values of the smoothing mask defined in Eq. (10), and whose value is:

$$
k = \frac{1 + 24a_1 + 96a_2 + 96a_3}{12}.
$$

These expressions are also valid if we are interested in detecting only first-order edges, setting $d = f = g = 0$. With these values, the edge surface is completely specified, and all the parameters of the edge (intensity change, subvoxel position, orientation and principal curvatures) can be computed using the same equations as in the previous sections.

In the presence of significant noise, we have observed that the values $a_0 = a_1 = a_2 = a_3 = {}^1/_{27}$ for the smoothing mask (Eq. (10)) provide a better performance. However, when noise is very low, we can increase the value of a_0 and decrease the values for the rest of the mask, in order to reduce the effect of the smoothing process. If we choose $a_0 = 1$ and 0 for the rest of values, it is equivalent to the method proposed in the previous section, in which no smoothing is applied.

5.2. Synthetic examples

In order to show some examples of the method's performance with noisy images, we have tested our algorithm on two ideal images with solid objects (a sphere and a torus), in which Gaussian noise of different magnitudes has been added. The application of our original method, presented in Section 4.3, and oriented to ideal images, produces significant errors in the presence of noise. However, the proposed smoothing-based method, explained in the previous subsection, is significantly more accurate.

Fig. 11 shows the error in estimating the direction of the normal vector at each voxel, for different levels of noise, and using three different methods: the original quadratic method applied to the original image, the classical derivative operators applied to the smoothed image, and the smoothing-based method proposed in this section, applied to the

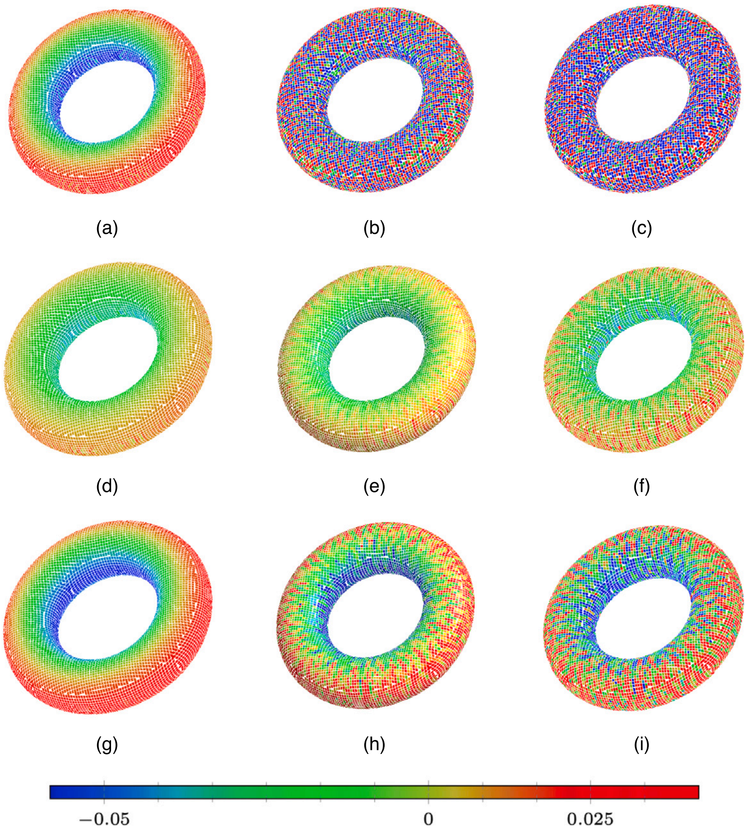


Fig. 12. Minimum curvature computed on a solid torus of radii $R = 40$ and $r = 10$ with noise added (from left to right: 0, 5, and 10). From top to bottom: method proposed in Section 4.1 applied on the original noisy image F; traditional method applied on the smoothed image G; smoothing-based method applied on the smoothed image.

Table 2
Results obtained with different methods (original quadratic method, traditional derivative operators, and smoothing-based method) when detecting the edges of a sphere of radius 20 affected by different noise levels.

Noise	Method	Error						Curvature			
		Intensity		Orient. (ž)		Pos. (%)		Min.		Max.	
		av.	max.	av.	max.	av.	max.	av.	rad.	av.	rad.
	2nd order	0.00	0.00	0.16	0.66	0.21	0.93	0.047	21.3	0.052	19.2
$\sigma = 0$	Trad. oper.	71.7	80.6	1.10	1.80	–	–	0.025	40.6	0.028	35.4
	Smoothed	0.00	0.03	0.28	1.02	0.12	1.03	0.050	20.0	0.051	19.7
	2nd order	1.93	7.53	2.33	8.03	5.36	32.5	0.015	68.1	0.257	3.89
$\sigma = 5$	Trad. oper.	71.7	82.7	1.15	2.57	–	–	0.023	42.8	0.030	33.9
	Smoothed	0.42	2.20	0.78	2.57	2.22	11.8	0.044	22.9	0.058	17.4
	2nd order	2.66	11.7	3.27	12.7	7.43	37.9	0.068	14.7	0.495	2.02
$\sigma = 10$	Trad. oper.	71.9	86.8	1.26	3.68	–	–	0.021	46.9	0.032	31.7
	Smoothed	0.61	2.70	1.08	4.38	3.02	16.5	0.043	23.2	0.061	16.3

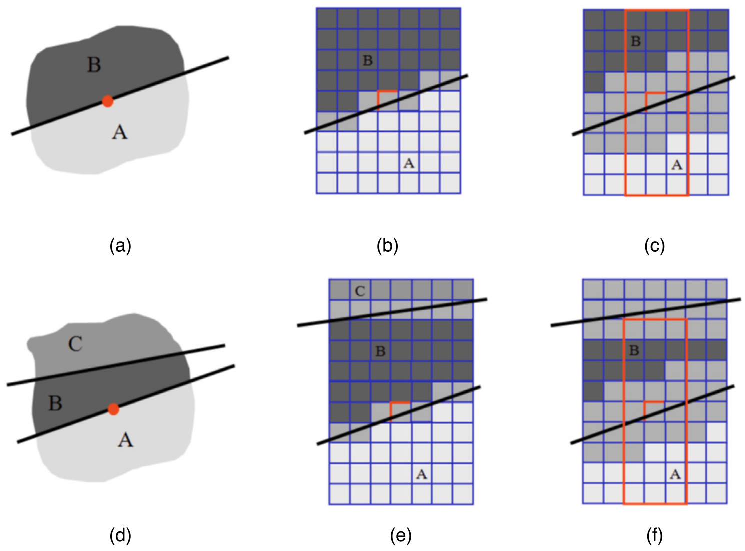
image after smoothing. Important errors are observed when our original quadratic method is applied to noisy images because it has been developed for ideal images (top row). The application of derivative operators on the previously smoothed image improves accuracy in noisy images, but significant errors are observed, even in the noise-free image (middle row). With the smoothing-based method that we propose, errors are smaller for low levels of noise, and accuracy is preserved when there is no noise (bottom row).

Table 2 shows the numerical errors in the estimation of intensity, orientation (degrees), subvoxel position (measured as a percentage over the length of a voxel), and curvatures (average curvature and curvature radii). We can see that the average and maximum errors are lower with the smoothing-based method than with the other two methods for all edge features. Regarding the intensity change, traditional derivative operators generate a very large error (greater than 70 units) due to the previous smoothing of the image, which blurs the edges and makes the magnitude of the intensity change smaller in the edge voxels. Our method, on the other hand, has a fairly good estimation. With respect to the estimation of curvatures, which are very sensitive to noise, we also see that the mean values for the curvature radii are closer to the actual radius of the sphere (20 units) with the proposed method, even with the highest level of noise (see highlighted values in bottom row).

Fig. 12 shows the results obtained with the three methods for the image of a torus of radii 30 and 10 with different noise levels. The color in each voxel indicates the minimum curvature on the surface of the torus. In this case, the expected values vary from $-1/20$ on the inner side of the torus to $1/40$ on the outer side. This curvature is difficult to estimate, especially in noisy conditions. However, we can see that it is reasonably well computed with the smoothing-based method.

6. Location of nearby edges

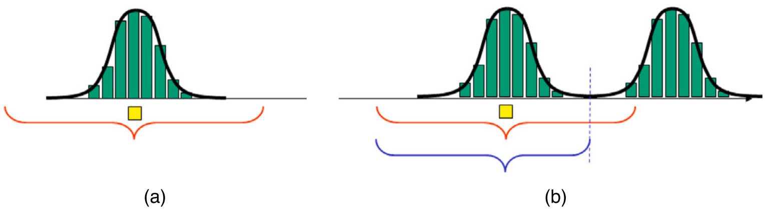
The algorithm proposed in the previous section uses a 3 × 11 × 3 window centered at every edge voxel of the smoothed image to estimate



(a) (b) (c)

(d) (e) (f)

Fig. 13. Comparison between a single and two nearby edges: **(a)** – **(c)** Single edge separating two different intensity values. **(d)** – **(f)** Two close edges where the top region with intensity C affects the values in the window for the lower edge. **(c)** and **(f)** Result of the smoothing stage. Figure extracted from [4].



(a) (b)

Fig. 14. Cross-section of partial derivative G_y with **(a)** single and **(b)** double edges: The red curly brackets indicate the voxels which are considered in the estimation of edge features for the yellow voxel if fixed limits are used. The blue curly bracket indicates the voxels which are considered when floating limits are used. Figure extracted from [4].

edge parameters. This is a relatively large region, where we assume there is no other edge. However, in many real scenarios, we may have to deal with very close edges within the same small subimage. In these cases, the window might have three regions with different intensities, delimited by two edges. If we directly applied the previously described method, feature estimation would be wrong, since the smoothing stage could alter the reference values. This is illustrated in Fig. 13 for a 2D case, where the intensity of the pixels at the top of the window would not be B after the smoothing process (Fig. 13**(f)**).

6.1. Using windows with floating limits

The solution we propose is an extension of the 2D case, originally introduced in [4], and consists in using floating limits for the columns used in the estimation of edge parameters. Therefore, the sums of Eq. (11) become:

$$S_M = S_{0,m_1,m_2,0}; \qquad S_L = S_{-1,l_1,l_2,0}; \qquad S_R = S_{1,r_1,r_2,0};$$
$$S_B = S_{0,b_1,b_2,-1}; \qquad S_F = S_{0,f_1,f_2,1}; \qquad S_D = S_{-1,d_1,d_2,\gamma} + S_{1,d_3,d_4,-\gamma};$$

where the limits l_1, l_2, r_1, r_2, b_1, b_2, f_1, f_2, m_1, m_2, d_1, d_2, d_3 and d_4 are computed for each case. These limits are determined by the voxels where the derivatives are minimum within the corresponding column. To illustrate the reason for such limits, Fig. 14**(a)** shows a cross-section of the derivative across an edge after the smoothing process. If there exists a single edge, fixed limits work properly. However, if there is a second edge in the region used for edge characterization (Fig. 14**(b)**), the sum must be limited as close as possible to the minimum within the corresponding section, depicted by the dashed vertical line.

In order to estimate the intensities A and B, we need to use the voxels located in the limits of the window, selecting the appropriate columns in each case according to the orientation of the edge. Fig. 15 shows the four options.

Finally, we solve the new system of linear equations to obtain the coefficients of the surface, according to the limits of the window:

$$\begin{aligned}
a &= I\left(2S_M - A(1-2m_1) - B(1+2m_2)\right) - k(dh_x^2 + gh_z^2) \qquad (12)\\
b &= \frac{I}{h_x}(S_R - S_L + A(r_1 - l_1) - B(r_2 - l_2))\\
c &= \frac{I}{h_z}(S_F - S_B + A(f_1 - b_1) - B(f_2 - b_2))\\
d &= \frac{I}{h_x^2}(S_R + S_L - 2 * S_M + A(r_1 + l_1 - 2m_1) - B(r_2 + l_2 - 2m_2))\\
f &= \frac{\gamma I}{h_x h_z}(S_R + S_L + S_F + S_B - S_D - 2 * S_M +\\
&\quad + A(r_1 + l_1 + f_1 + b_1 - d_1 - d_3 - 2m_1) -\\
&\quad - B(r_2 + l_2 + f_2 + b_2 - d_2 - d_4 - 2m_2))\\
g &= \frac{I}{h_z^2}(S_F + S_B - 2 * S_M + A(f_1 + b_1 - 2m_1) - B(f_2 + b_2 - 2m_2))
\end{aligned}$$

In order to illustrate the advantages of using a window with floating limits, Fig. 16 shows an image with two close solid spheres of radius 20 separated by 5 voxels (Fig. 16**(a)**). After smoothing, this distance is lower (Fig. 16**(b)**). Let us consider the voxel marked in green. If windows with fixed limits are used for the edges of one of the spheres, these windows reach the other sphere and, therefore, the condition of having a single edge inside the window is not met. For this reason, the application of the method with fixed limits generates large errors in the estimation of edge parameters (Fig. 16**(c)**). However, the use of windows with floating limits solves this problem (Fig. 16**(d)**). Moreover, even in the presence of noise, the method with floating limits is able to work properly, as observed in Fig. 17.

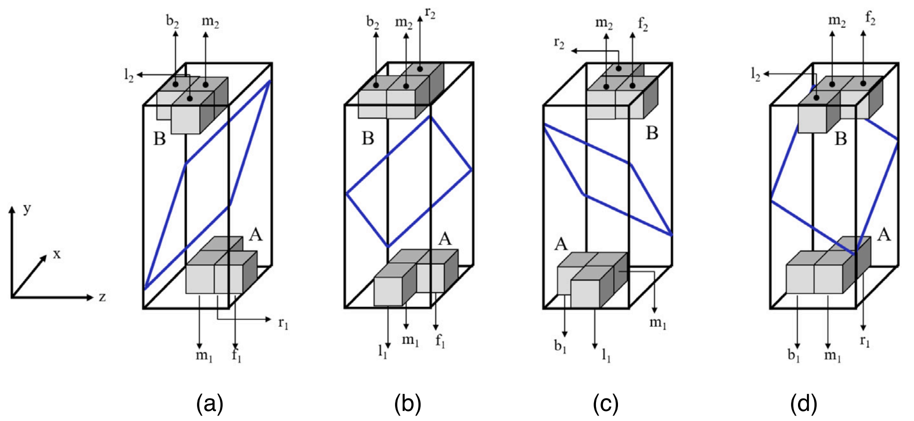


Fig. 15. Voxels chosen to estimate intensities A and B depending on the orientation of the surface: **(a)** $|G_xG_y| < 0$ and $|G_yG_z| < 0$. **(b)** $|G_xG_y| > 0$ and $|G_yG_z| < 0$. **(c)** $|G_xG_y| > 0$ and $|G_yG_z| > 0$. **(d)** $|G_xG_y| < 0$ and $|G_yG_z| > 0$.

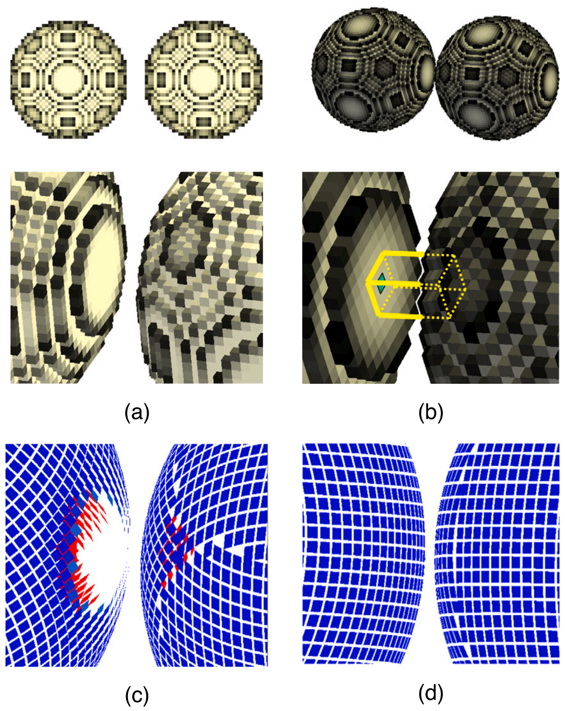


Fig. 16. Example of the detection of close edges: **(a)** Image of two close solid spheres and detail. **(b)** Smoothed image and detail, where the window (in yellow) used to estimate edge parameters of the voxel colored in green reaches the other sphere. **(c)** Result using the proposed method with fixed limits. **(d)** Result using the proposed method with floating limits.

6.2. Tackling very close edges

The improvement proposed in the previous subsection solves the problem of dealing with more than one edge in the same window, except for an extreme situation in which two edges are so close that they mutually interfere when a smoothing process is applied. In these cases, not only must the window be divided into three regions, delimited by both edges, but we also have to recover the original intensity of the central region. It might have been blurred by both surrounding regions, i.e., the intermediate intensity B between the edges could have been lost after the smoothing process, as seen in Fig. 18**(c)**.

As illustrated in Fig. 18**(d)**, we can detect this case by checking that the minimum of the derivative of the intensity is not zero, and values after that point, represented by the dashed vertical line, are not negligible. This scenario is depicted in Fig. 19**(a)**, where there are two very close edges, with intensities 200 between them and 100 outside. After smoothing, we obtain image 19**(b)**. We are interested in estimating edge features in the voxel marked in blue, which belongs to the lower edge. This voxel is marked as edge voxel because its absolute value in the derivative image (Fig. 19**(c)**) is maximum within its column.

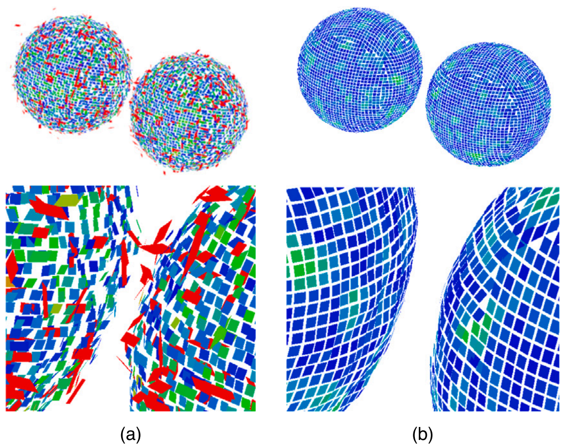


Fig. 17. Detection of close edges in noisy images: **(a)** Proposed method with fixed limits applied to an image of two spheres with added noise with $\sigma = 30$. **(b)** Detection using floating limits on the smoothed image.

If our method was applied as above, the columns marked in red would be used. In this case, the estimation for B would be wrong (voxels marked in green in Fig. 19**(b)**). The solution consists in first, detecting that we are in a case with very close edges (derivative values are significant beyond the window limit). Then, B must be estimated using the values in the original image (green voxels in Fig. 19**(a)**), instead of using the smoothed version.

With this modification, another problem arises: when equations in (12) are applied to obtain the values for the surface coefficients, the estimation might be wrong because the values at the top of the window (Fig. 19**(b)**) are not equal to the expected value B. We can solve this issue by generating a new intermediate subimage F', of size $13 \times 5 \times 5$, centered at the corresponding voxel, containing only the edge that crosses it (Fig. 19**(d)**). This subimage will match the original voxel values, except for the intensities of the voxels above those used to estimate B, which will be substituted with B. Visually, this new subimage is the same as the original one, but with the upper edge removed. Then, subimage F' is smoothed to obtain subimage

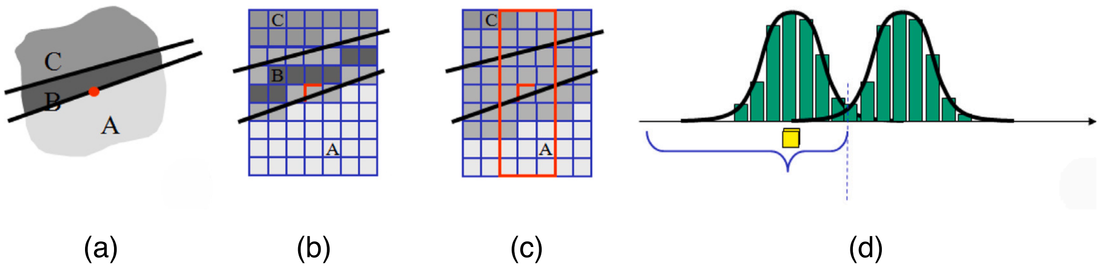
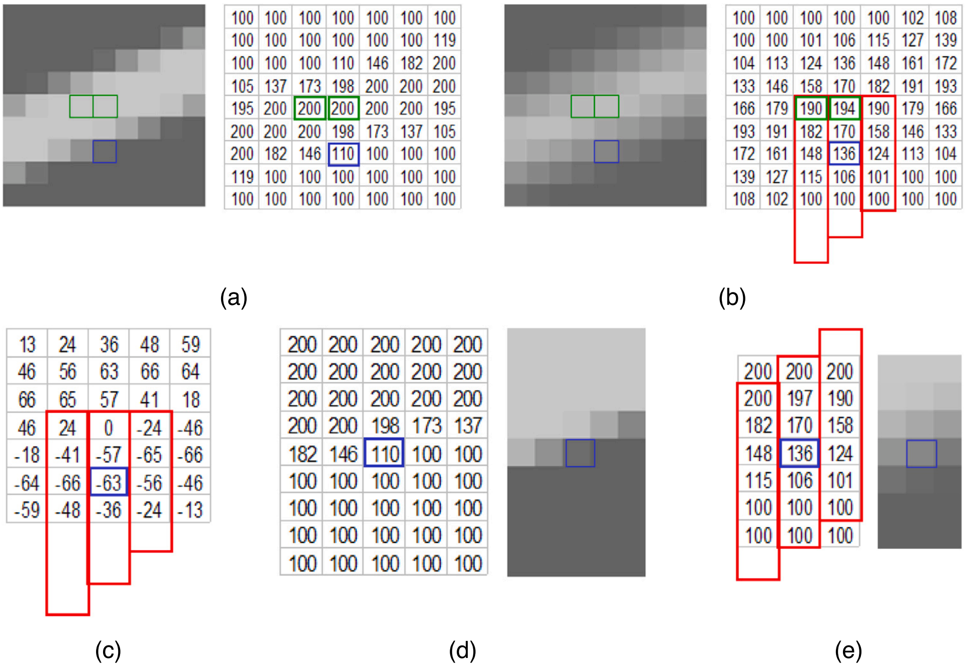


Fig. 18. Extremely close edges: **(a)** Illustration of very close edges. **(b)** Digitized image. **(c)** Smoothed image, where the value of B has been lost. **(d)** Cross-section of derivatives. Figure extracted from [4].



(a)

100	100	100	100	100	100	100
100	100	100	100	100	100	119
100	100	100	110	146	182	200
105	137	173	198	200	200	200
195	200	200	200	200	200	195
200	200	200	198	173	137	105
200	182	146	110	100	100	100
119	100	100	100	100	100	100
100	100	100	100	100	100	100

(b)

100	100	100	100	100	102	108
100	100	101	106	115	127	139
104	113	124	136	148	161	172
133	146	158	170	182	191	193
166	179	190	194	190	179	166
193	191	182	170	158	146	133
172	161	148	136	124	113	104
139	127	115	106	101	100	100
108	102	100	100	100	100	100

(c)

13	24	36	48	59
46	56	63	66	64
66	65	57	41	18
46	24	0	-24	-46
-18	-41	-57	-65	-66
-64	-66	-63	-56	-46
-59	-48	-36	-24	-13

(d)

200	200	200	200	200
200	200	200	200	200
200	200	200	200	200
200	200	198	173	137
182	146	110	100	100
100	100	100	100	100
100	100	100	100	100
100	100	100	100	100
100	100	100	100	100

(e)

200	200	200
200	197	190
182	170	158
148	136	124
115	106	101
100	100	100
100	100	100

Fig. 19. Analysis of very close edges and generation of the intermediate subimage: **(a)** Distribution of intensities in a section with two very close edges. **(b)** Smoothed image. **(c)** Derivative image. **(d)** Generated intermediate subimage F'. **(e)** Smoothed subimage G'. Figure extracted from [4].

G' (Fig. 19**(e)**), and our method is applied on this new subimage to estimate the coefficients.

Fig. 20 shows an image with two solid spheres separated by just 1 voxel. Without noise, the method is able to detect all the edges and extract the intensities on both sides (Fig. 20**(c)**). In the presence of noise, the use of floating limits and the individual estimation of extremely close edges provide a quite satisfactory result, as observed in Fig. 20**(e)**.

7. Experimental results and discussion

In the previous sections, we have illustrated how the different stages and variations introduced in the proposed method improve the results and allow overcoming the difficulties caused by certain circumstances, such as noise, close edges, or interference between edges. In this section, we present a comparison with the 3D Canny edge detector, which is one of the most widely used methods, as well as some additional experiments to assess the proposed method in several synthetic images, a real CT scan of an artificial object (phantom) whose actual measures are known, and a medical image (CT angiography).

It is very difficult to compare the results with other edge detectors in the literature that work at subvoxel level, since we have found no method whose implementation is available, except for the widely used Canny edge detector. The 3D images used in other works and their associated ground truth are not publicly available either, making it impossible to apply our method to these images and conduct a comparative analysis of the results. In order to facilitate the access to ground truth for real and synthetic 3D images, and intending to contribute to research reproducibility, all the examples[1] presented in this paper and the source code[2] are freely available. The configuration of the final method is based on four input parameters: a threshold for the change in intensity to decide whether a voxel is considered edge or not; the order of the function to fit edges (planes or quadratic surfaces); a flag to indicate whether smoothing is previously applied (noisy images) or not; and the size of the voxels in each dimension (in millimeters).

7.1. Comparison with the 3D canny edge detector

This subsection presents a comparison between the proposed method and a state-of-the-art technique. The chosen method for comparison is the 3D Canny detector, which serves as the foundation for numerous detectors in the literature, such as [16,22]. The used implementation is the Matlab source code found in [39]. Although this version does not perform the calculation of principal curvatures, the

[1] http://ctim.ulpgc.es/demo114

[2] https://es.mathworks.com/matlabcentral/fileexchange/121088-accurate-subvoxel-edge-location-for-3d-images

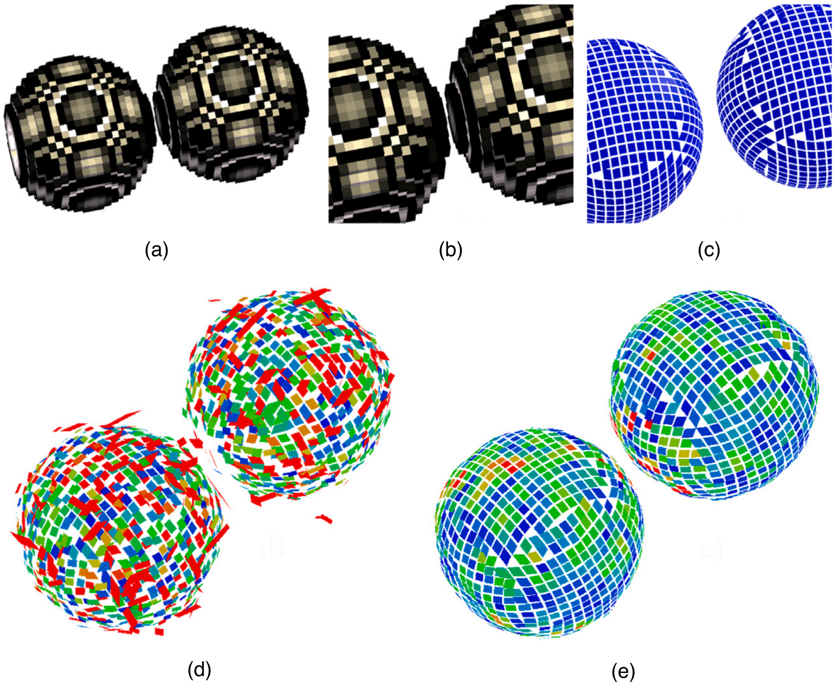


Fig. 20. Detection of edges on two very close spheres without and with noise: **(a)** Two solid spheres separated by 1 voxel. **(b)** Detail of the image. **(c)** Detected edges in noiseless image. **(d)** Edge detection after adding noise to the image. **(e)** Detection of edges on the smoothed image.

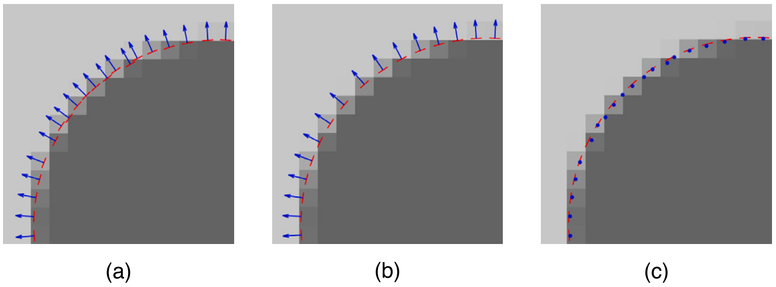


Fig. 21. Edge detection on a cross-section of a solid sphere of radius **10**: **(a)** Position and orientation applying 3D Canny detector. **(b)** Position and orientation using our proposal. **(c)** Comparison of the 3D subvoxel position between Canny – blue dots – and our technique – red lines –.

estimation of three parameters (subvoxel position, edge direction, and intensity difference) allows us to carry out the comparison.

The geometric object used for the comparison is a solid sphere of radius 10 and intensity difference 100. This sphere strictly adheres to the PVE assumption. To illustrate the difference between both methods, Fig. 21 shows a cross section through the center of the sphere. Our method aims to achieve high accuracy by specifically addressing the reversal of PVE assumption. Conversely, the Canny method, based on derivative operators for computing gradients and Hessians, assumes continuity in the image function across all voxels, completely disregarding PVE, and consequently introducing errors in all the parameters. As observed in Fig. 21, edge positions are more homogeneously distributed with our method than with the Canny detector. Moreover, edge orientation changes more gradually with our method than with the Canny detector.

The errors for the three parameters when applying both methods on a sphere are shown in Fig. 22. As with Fig. 21, the distribution of the detected edges is also optimal with our method, in the sense that a single edge per column or row of the image (depending on the surface slope) is returned. This does not occur with the Canny detector in regions with high slope, producing edge voxel cluttering.

Table 3 shows the average and maximum errors. The errors obtained by the proposed method are much lower.

Table 3
Average and maximum errors obtained for both methods when detecting the edge features of a solid sphere of radius 10.

Method	Error					
	Intensity		Orient. (ž)		Pos. (%)	
	av.	max.	av.	max.	av.	max.
3D Canny	5.23	12.6	1.30	2.75	13.9	31.8
Proposed method	0.00	0.00	0.39	1.32	0.17	0.76

7.2. Synthetic images

In this subsection, we present some examples to illustrate the performance of our method with different objects. All the synthetic images in this section have been generated using inner and outer intensities of 100 and 200, respectively. The threshold for the intensity change has been set to 20 and voxel size is $[1, 1, 1]$. The first synthetic example consists of several tori with different radii (see Fig. 23). The errors are negligible for radii of curvature greater than 2 units, but some errors appear for shorter radii of curvature. In particular, these errors arise in those voxels where the surface does not cross the window from one side to the opposite one (none of the four cases in Fig. 7).

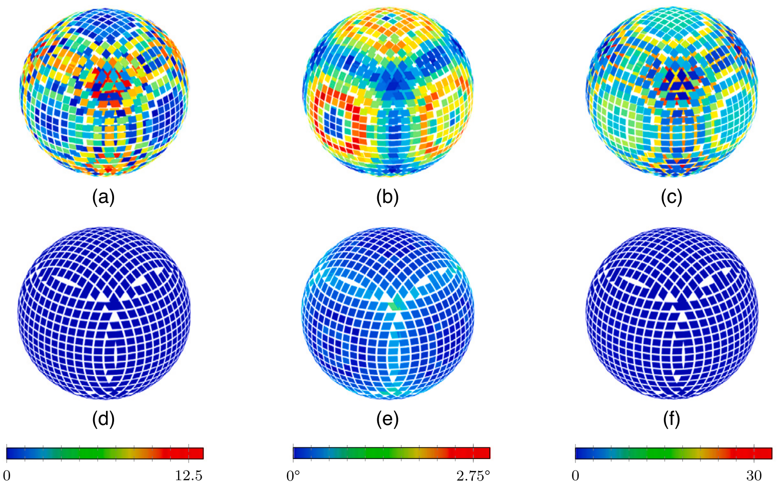


Fig. 22. Errors in the edge detection for a sphere of radius 10 comparing the Canny detector (top) and our proposal (bottom): **(a)** and **(d)** Errors for intensity change. **(b)** and **(e)** Errors for orientation. **(c)** and **(f)** Errors for subvoxel position.

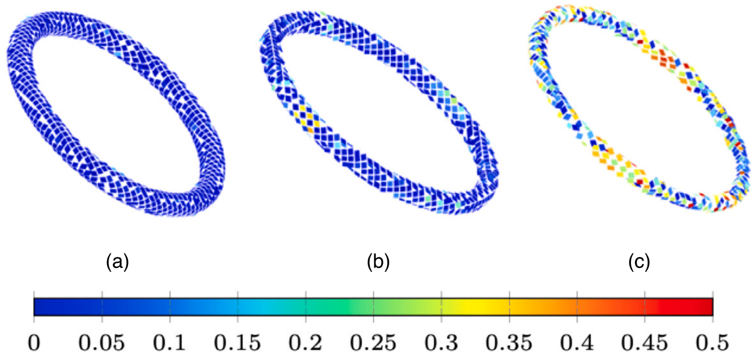


Fig. 23. Errors obtained in the estimation of the subvoxel position for tori with major radius 20 and minor radius: **(a)** 2.0, **(b)** 1.5 and **(c)** 1.0 units. An error of 0.1 corresponds to 10% of the length of a voxel.

The second synthetic example is a tubular object with a varying radius which follows the expression $5 + 3sin(kx)$. In this case, the radius ranges from 2 units in the narrowest region to 8 units in the widest one. To test the robustness of the method in the presence of noise, Gaussian noise has been added to the image before applying the detector (Fig. 24**(b)**). Figs. 24**(c)** and Fig. 24**(d)** show the radius of curvature in every edge voxel (inverse of the maximum curvature) for both the original and the noisy image. Similarly, Figs. 24**(e)** and Fig. 24**(f)** show the directions of principal curvatures. As observed, the results obtained for the edge parameters (subvoxel position, radius and directions of curvature) are coherent with the expected results.

The third example (Fig. 25) is a helix with varying radii. The radius of the tubular structure ranges from 6 units at the bottom to 2 units at the top. In this case, the radius of the section of the helix decreases linearly. As in the previous case, the method was tested on the original image as well as on a noisy version. Figs. 25**(c)** and Fig. 25**(d)** show the radius of maximum curvature obtained for each edge voxel. As observed, the estimated radius varies from 6 units at the bottom (red) to 2 units at the top (blue). Even with noise added to the image (Fig. 25**(d)**), this tendency is maintained.

7.3. Phantom

Medical images are a clear example of 3D images that need to be processed to compute different measurements of organs or tissues and help the medical specialists in the diagnosis of a large variety of diseases. To test the accuracy of the algorithms, artificial objects whose measures are known, named phantoms, are constructed and scanned with the same devices that will be used with the patients. In this case, we have used the phantom built by Martínez-Mera et al. in [40]. This phantom represents a long blood vessel, consisting of a curved glass tube with a mixture of water and a radiographic contrast media inside (see Fig. 26**(a)**). The thickness of the glass is 1.5 mm, and the tube has an outer radius of 6.0 mm and an inner radius of 4.5 mm (Fig. 26**(b)**). This phantom has been introduced in a CT scanner to obtain a 3D image of 512 × 512 × 273 voxels, where the size of each voxel is 0.7 × 0.7 × 0.625 mm (non-isotropic voxels). To speed up the process and ignore other surfaces present in the image that do not belong to the phantom, our method is only applied to a subvolume of size 231 × 141 × 273 voxels, where the phantom is located. The image has three regions with different mean intensities in Hounsfield units: 34240 inside, 35340 for the glass, and 32820 outside the tube (Fig. 26**(c)**). In what follows, the references to intensity values are indicated in Hounsfield units.

We must note that some edge voxels separate the glass from the outer side of the tube, while others separate the glass from the inner side, which represents a clear case of very close edges. For a better validation of the results of our algorithm, all edge voxels have been classified into one of these two groups. To identify to which group each voxel belongs, we must take into account that the inner region is significantly brighter than the outer region, and glass is brighter than both of them. For this reason, we classify each edge voxel as outer or inner edge according to whether the intensity of its darkest region is lower or higher than 33530, respectively. Fig. 27 shows both groups of voxels, and the color represents the change in intensity between both sides of the edge (i.e. the difference between the intensity of the tube

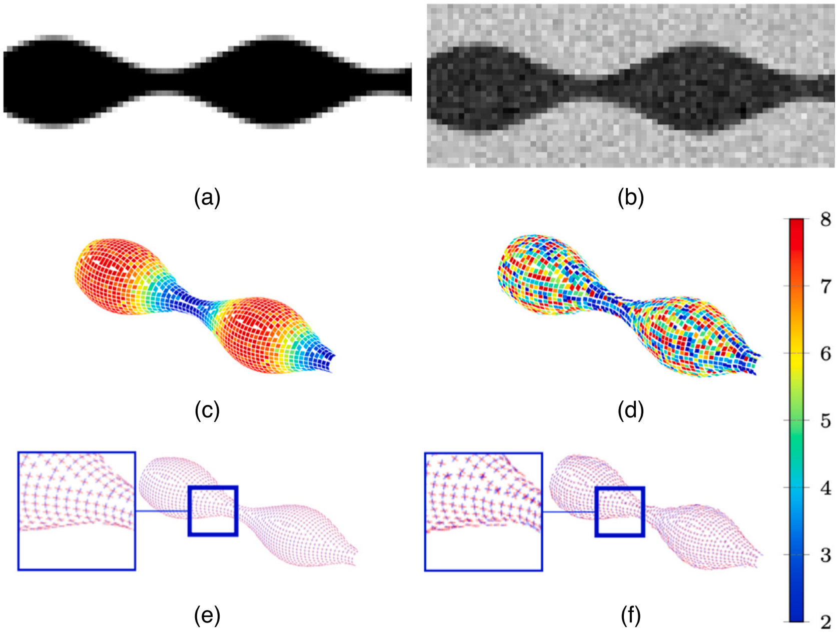


Fig. 24. Subvoxel detection in a variable-radius cylinder: **(a)** Cross-section of the original noise-free image. **(b)** Cross-section of the noisy image (with Gaussian noise added). **(c)** and **(d)** Radius of curvature for both cases. **(e)** and **(f)** Directions of principal curvatures for both cases.

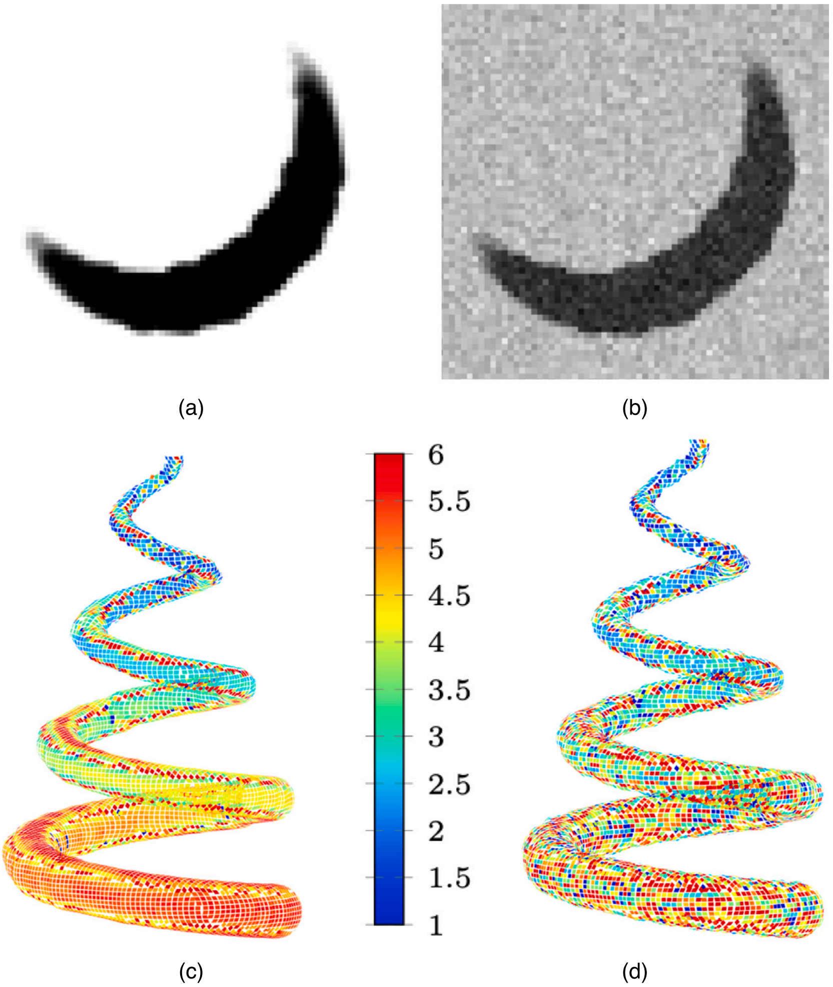


Fig. 25. Subvoxel detection in a helix with varying radii: **(a)** and **(b)** Cross-sections of the original noise-free image, and the noisy image (with Gaussian noise added), respectively. **(c)** and **(d)** Radius of maximum curvature for both cases.

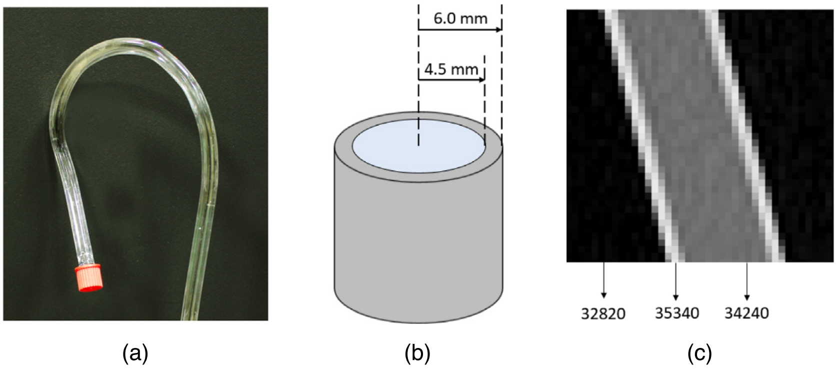


Fig. 26. Detail of the phantom used to test the accuracy of the method: **(a)** Photograph of the phantom. **(b)** Measures of the tube. **(c)** Sample of the CT with approximate mean intensities of the three regions.

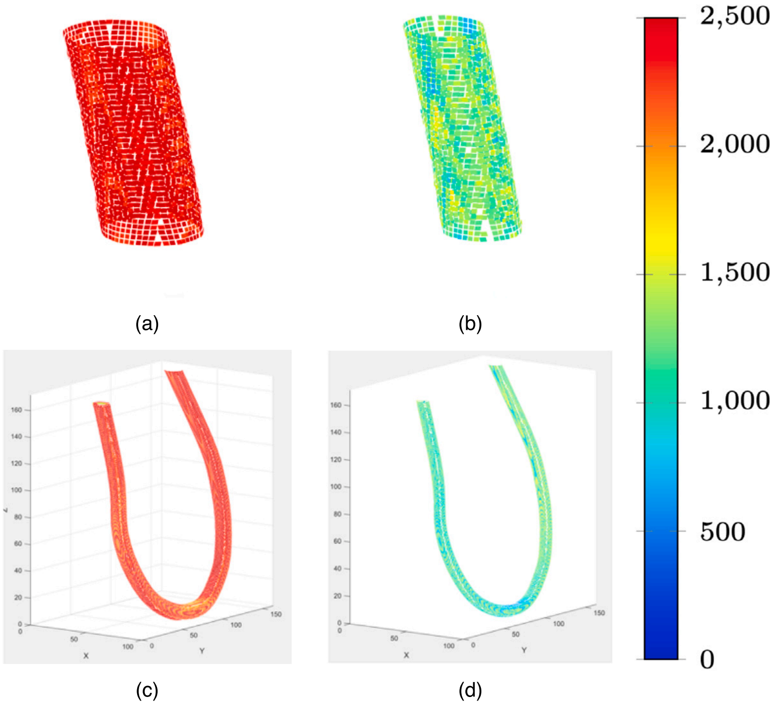


Fig. 27. Change in intensity obtained for each edge voxel of the phantom: **(a)** Detail of the edges on the outer side. **(b)** Detail of the edges on the inner side. **(c)** Whole view of the outer side of the tube. **(d)** Whole view of the inner side of the tube.

and that of the inner or outer region). The mean value obtained for each group is 2489 for the outer side (Figs. 27**(a)** and Fig. 27**(c)**, close to the expected value 2520), and 1145 for the inner side (Figs. 27**(b)** and 27**(d)**, close to the expected value 1100). The input parameters used for the edge detector were 50 as intensity threshold, second order edges, and previous smoothing. The execution time of the proposed method for this image (8.9 megavoxels) is 5.4 s.

According to the measures of the tube, the maximum curvature for the voxels belonging to the outer side of the tube must be close to $^{-1}/_{6} \approx -0.1666$, whereas it must be $^{1}/_{4.5} \approx 0.222$ for the inner side. It is negative for the outer side because the surface blends toward the region with higher intensity. Fig. 28 shows the inverse of the maximum curvature for all edge voxels. For those on the outer side (Figs. 28**(a)** and Fig. 28**(c)**), the average of the maximum curvatures is -0.174, which is equivalent to a mean outer radius of 5.75 mm, very close to the expected 6.0 mm. For voxels belonging to the inner side (Figs. 28**(b)** and 28**(d)**), the average of the maximum curvatures is 0.280, which is equivalent to a mean inner radius of 3.57 mm. This latter value differs somehow from the expected 4.5 mm, but we must bear in mind that curvature values are extremely sensitive.

The minimum curvature vector should be oriented in the main direction of the tube for all edge voxels. Conversely, the maximum curvature vector must be oriented along the section of the tube. Both vectors must be perpendicular. Fig. 29 shows these two vectors for each

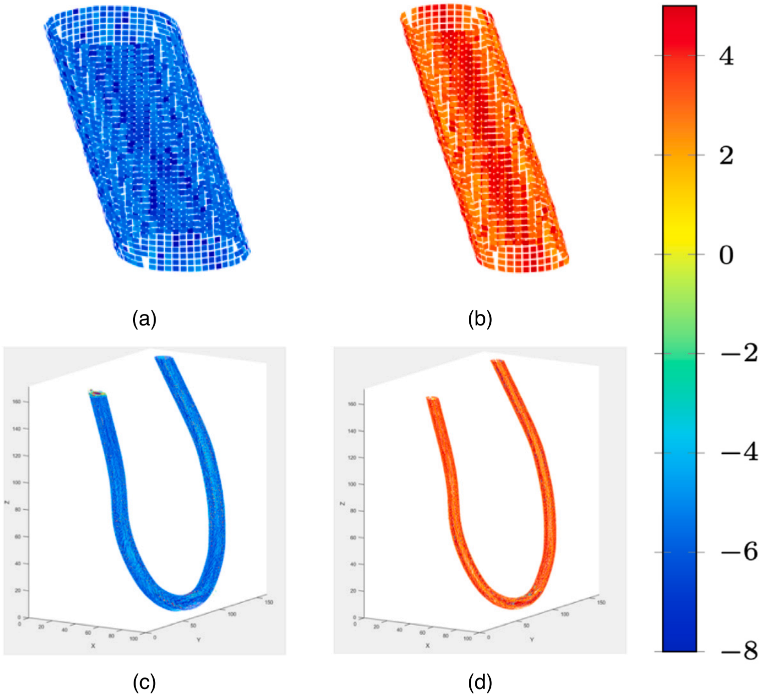


Fig. 28. Radius of maximum curvature obtained for each edge voxel: **(a)** Detail of the edges on the outer side. **(b)** Detail of the edges on the inner side. **(c)** Whole view of the outer side of the tube. **(d)** Whole view of the inner side of the tube.

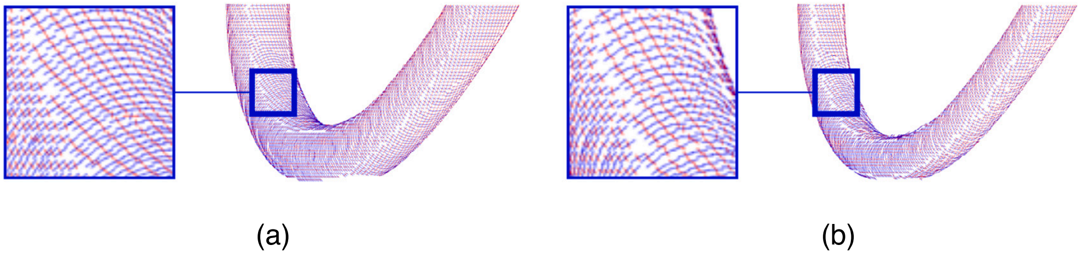


Fig. 29. Directions of principal curvatures for each edge voxel. Minimum curvature vectors are displayed in red, and maximum curvature vectors in blue: **(a)** Outer side of the tube. **(b)** Inner side of the tube.

edge voxel belonging to the outer (Fig. 29**(a)**) or inner side (Fig. 29**(b)**) of the tube. As observed, the result is coherent with the expected directions.

In order to measure the accuracy in the estimation of the position for both surfaces, several pairs of opposite points have been selected. To identify these pairs of points, we look for those whose normal vectors are 180° apart from each other and are also aligned with the same straight line (up to a certain threshold) (see Fig. 30). The mean distance between pairs of aligned points belonging to the outer side is 12.02 mm (Fig. 30**(a)**), whereas the mean distance is 9.08 mm for points on the inner side (Fig. 30**(b)**). Therefore, the estimated radii for the outer and inner sides are 6.01 and 4.54 mm, respectively, very close to the expected values (6.0 and 4.5 mm).

7.4. Medical images

A real chest CT angiography has been used to test the method on images of non-synthetic objects in a real application scenario. This image was scanned in the Department of Radiology of the University Hospital of Santiago de Compostela, Spain [41]. Intravenous contrast was injected in order to enhance the opacity of the blood in the image. The resolution of the image is 512 × 512 × 500 voxels, with a voxel size of 0.5957 mm in (x, y), and a distance of 0.62 mm between slices (non-isotropic voxels). Voxel intensities range from zero to 3071 Hounsfield units.

Fig. 31 shows three different slices of the image, S_1, S_2 and S_3 (Figs. 31**(a)**, 31**(b)**, and 31**(c)**, respectively). In order to assess the result of the edge detection more clearly, let us focus on a smaller subvolume of size 200 × 180 × 150 voxels, marked in red inside these slices. Fig. 31**(d)** shows all the edges detected inside this subvolume. The color represents the change in intensity between both sides of the edge. The aorta (more specifically, the aortic arch) is the thickest vessel in blue, with a mean intensity of approximately 470 units. There are three thinner branches emerging from its upper side, which appear in the slices of the original image as three small circular sections (as observed in Fig. 31**(a)**). The other large vessel (in red in Fig. 31**(d)**) is the vena cava, with an average intensity of 2800 units (brighter region in the original image – see Figs. 31**(a)**–31**(c)**).

If we want to focus on a particular organ or structure in the image, we need to have an approximate location or delimitation of that element. For instance, in Fig. 31**(e)**, we have used a voxel-level segmentation of the aorta obtained using the method in [41], and we only show the subvoxel parameters for those voxels which are close to the boundary of this segmentation, removing all other vessels and structures. The intensity threshold used for the edge detector was 110

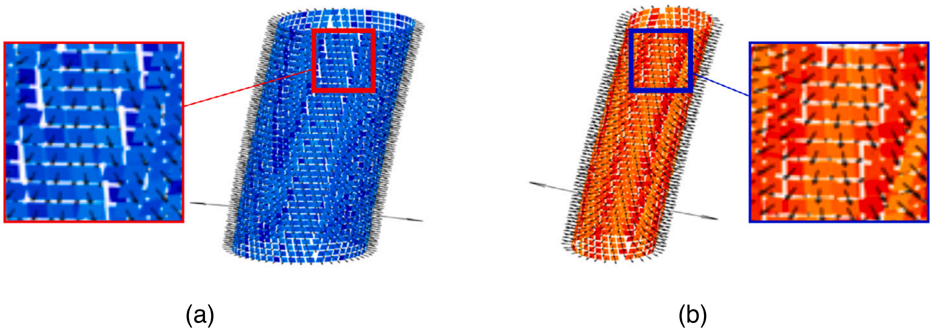


Fig. 30. Estimation of the radii of both sides of the tube: Pairs of points are selected so that they are aligned and their normal vectors are 180° apart from each other, approximately. **(a)** Estimation for the outer side of the tube. **(b)** Estimation for the inner side. In each case, a sample pair of vectors has been enlarged.

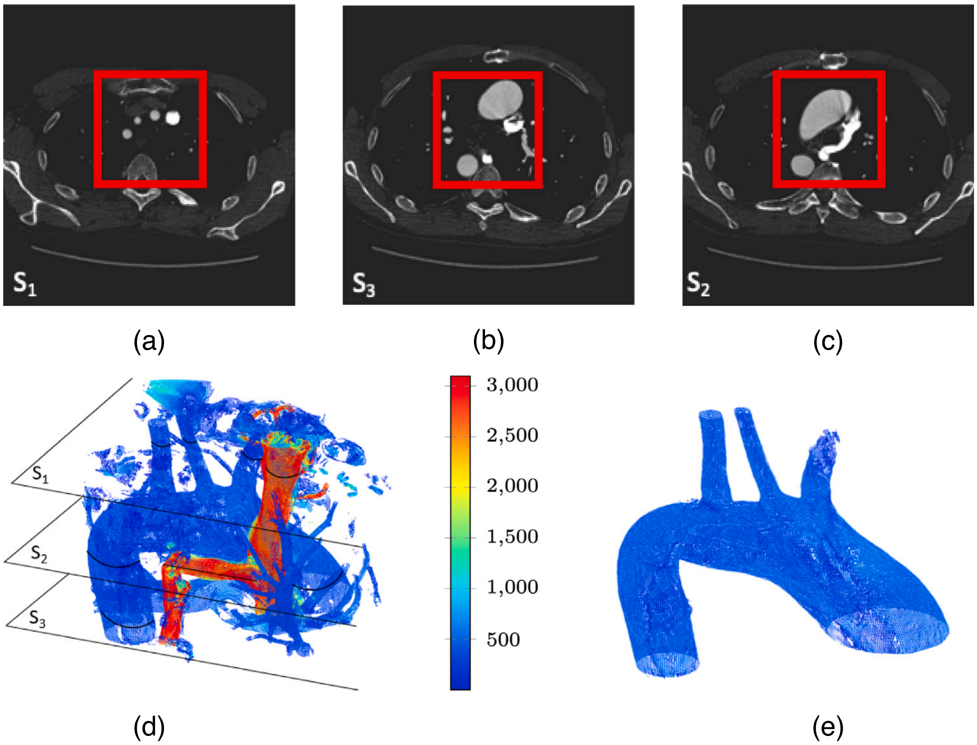


Fig. 31. Result of applying the proposed technique to a real CT scan: **(a)**, **(b)**, and **(c)** Different slices in a CT angiography, with a selected subvolume marked in red. **(d)** Edge detection inside the subvolume indicating the slices in **(a)**, **(b)**, and **(c)**. **(e)** Selection of the edge voxels belonging to the aorta. Color in **(d)** and **(e)** indicates change in intensity between both sides of the edge.

in Fig. 31**(d)** and 60 in Fig. 31**(e)**. This is because we could be more precise with the threshold when focusing on voxels belonging to the aorta. The execution times for edge detection in these two subfigures (5.5 megavoxels) were 8.4 (all edges) and 1.7 s (only aorta).

Fig. 32 shows the edge detection for a different subvolume of the same CT image, corresponding to the abdominal area. Fig. 32**(a)** shows all the edge voxels detected inside the subvolume, with the color indicating the change in intensity. A different part of the aorta (more specifically, the abdominal aorta) can be seen in blue, with some ramifications (from top to bottom: celiac, superior mesenteric, and left and right renals). We can also see the vena cava, which is the long vessel whose color varies from cyan to red.

To illustrate the accuracy of our algorithm, Fig. 32**(b)** only shows those edge voxels close to the boundary of the aorta. In this case, the color of each voxel indicates the maximum curvature obtained. Although this parameter is very sensitive, we can see that voxels belonging to the main body of the aorta have a lower curvature value, indicating that the surface is smoothly curved, while voxels in the ramifications have a higher curvature value, since they correspond to thinner tubular structures. The execution times for edge detection in these two subfigures (5.5 megavoxels) were 4.9 and 7.9 s.

7.5. Discussion

The experiments shown in the previous sections prove that the proposed method is able to extract edge parameters with a high accuracy and overcome the difficulties of certain adverse scenarios. The introduction of the smoothing stage and the use of floating limits allow dealing with noisy images and extremely close edges. Not only are the position and orientation of the edges extracted with high accuracy, but the main curvatures are also satisfactorily estimated. Moreover, the distribution of the detected edges is also optimal in the sense that a

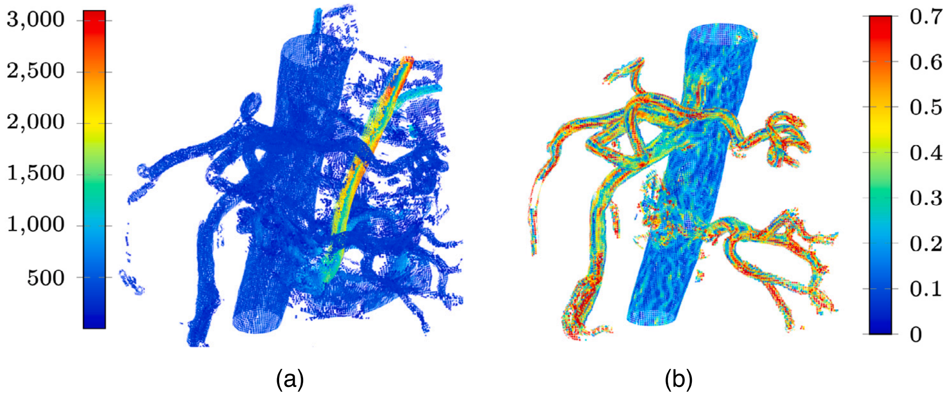


Fig. 32. Result of applying the proposed technique to a real CT scan of the abdominal area: **(a)** Edge detection in a CT angiography, with the color indicating change in intensity. **(b)** Representation of the edge voxels belonging to the aorta, where color indicates the maximum curvature.

single edge per column or row of the image (depending on the surface slope) is returned, thus avoiding the edge voxel cluttering that occurs with other detectors.

If we are interested in the position and orientation of the surface, but the curvatures are not so relevant, the first-order method is more robust and homogeneous. This is due to the fact that curvature estimation requires the computation of the second-order coefficients of the function that fits the edge, which are quite sensitive. Similarly, in images without noise, the original method may be more accurate than the smoothing-based method. The reason is that the latter requires a larger window for the estimation of edge features after applying the smoothing mask, which can interfere with areas further away.

Our technique is also compatible with images of non-isotropic voxels where the three dimensions of every voxel are not equal, such as those used in our non-synthetic examples (the phantom and the chest CT angiography) and most 3D medical images. In these cases, the parameters (position and curvature radii) are measured in millimeters instead of voxels. This way, an ideal solid sphere scanned with a non-isotropic sensor would produce edge voxels with equal principal curvatures for all of them.

From a computational point of view, our method is very fast, because it is mainly based on the sum of intensities within the rows (or columns) of certain windows centered at each edge voxel. All the experiments have been executed on an Intel Core i7-4790 CPU at 3.60 Hz. Considering run times, we can see that the size of the 3D image is not so relevant as the number of edge voxels which are found. Additionally, the method is slower if the number of voxels with nearby edges is larger. The reason is that, in these cases, a synthetic subimage of size 13 × 5 × 5 must be created and smoothed for each edge voxel to extract edge features from it, as described in Section 6.2. This is the case of angiographic images with many thin vessels (like some of the figures in Section 7.4), and represents the slowest part of the method.

Considering large images, the experiments in Section 7.4 were carried out in different subvolumes of a real chest CT angiography with 500 slices of size 512 × 512 voxels (131 megavoxels). Applying our edge detector to the entire image produces 1.27 million edge voxels, with a computing time of 131.1 s. In any case, in this work we have focused on accuracy and not on speed. The proposed method can be parallelized and could benefit from the power of GPU to greatly speed up the times on large images. Moreover, providing the method with a pre-segmentation or a region of interest can significantly reduce the computational time.

Regarding the limitations of the proposed method, we must take into account that our method aims to detect edges locally. Therefore, when extracting the surfaces of higher-level structures with corners or connected faces, such as polyhedral objects, our method would produce errors in those voxels with sharp orientation changes. In these cases, proceeding first to detect the edges with smooth orientation variations to obtain the faces of the objects, and then extracting their intersections, represents a better approach. Metrology applications embody a clear example of this type of scenario.

Another limitation of our method is related to surfaces with very high curvature. The equations of the method have been designed considering the case in which the surfaces intersect the side walls of the window that is centered at each edge voxel. However, those surfaces with slopes close to 1 and a very high curvature could intersect the floor or the ceiling of that window. In this case, small errors in the estimation of the edge features would appear (the higher the curvature, the greater the errors).

As regards noise, the method is quite robust and able to cope with a certain amount of noise, although a very high level of noise would obviously alter the estimations. With respect to contrast, one of the configurable parameters indicates what level of contrast is considered an edge. If this threshold is too low, many false edges will appear. If it is too high, some actual edges will be missing. Therefore, we should determine how selective we want to be depending on the context.

8. Conclusions

Image processing applications often require an extremely accurate location of the edges in different modalities of 3D images, as well as a precise description of certain features, such as orientation, curvature, or change in intensity. Traditional methods frequently fail to provide this information with a satisfactory accuracy.

A new and highly accurate method to obtain all the parameters of the edges at a subvoxel level has been presented in this paper. The detector is based on the assumption that the intensity of an edge voxel between two regions with different intensities depends on the portion of the voxel which is included within each region, known as partial volume effect (PVE). The proposed algorithm has proved to be very accurate on a comprehensive set of synthetic, phantom, and 3D medical images. In the case of ideal images with isolated edges, all edge parameters (subvoxel position, orientation, principal curvatures, and change in intensity between both sides of the edge) are accurately extracted.

In the presence of noise, the image is first smoothed and the method is applied on the smoothed version by using larger windows, which provides excellent results. The use of windows with floating limits addresses the problem of nearby edges, avoiding interference between the edges and providing more reliable estimations. Furthermore, when the distance between two edges is very short, a synthetic subimage centered at each edge voxel is created, which allows focusing on each one of the edges, and avoids the interference produced by the other one.

The estimation of both the magnitudes and the directions of the principal curvatures is especially sensitive, because these parameters are related to second-order derivatives. Most edge detectors do not compute these parameters accurately enough, but we have shown that, in most cases, the proposed approach is capable of satisfactorily estimating them.

Future directions of improvement include the parallelization of code to use the GPU for faster run times on large images. An additional functionality which could be included consists in obtaining surface fragments for sets of contiguous voxels from the features extracted for each of them. For instance, the complete surface of a vessel segment, such as those in the experiments, could be generated. Furthermore, we are developing a restoration algorithm that uses the results of the edge detector to progressively improve image quality.

In summary, the proposed method for detecting edges with subvoxel precision can be applied in all those fields which require a very accurate estimation of the surface of the objects. The results support this applicability, and the robustness of the estimations in challenging situations show that this method overcomes the limitations of traditional masks and can tackle challenging scenarios which could not be accurately characterized using a classical approach.

CRediT authorship contribution statement

Agustín Trujillo-Pino: Method conception and description, Method implementation, Writing – review & editing. **Miguel Alemán-Flores:** Method conception and description, Writing – review & editing. **Daniel Santana-Cedrés:** Experimental design and setup, Writing – review & editing. **Nelson Monzón:** Experimental design and setup, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data is shared in a link included in the paper.

Acknowledgments

This work was partially supported by Vicepresidencia Primera, Consejería de Vicepresidencia Primera *y* de Obras Públicas, Infraestructuras, Transporte *y* Movilidad from Cabildo de Gran Canaria, through the project of reference Resolution No. 45/2021.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.jvcir.2023.103928.

References

[1] M.A.G. Ballester, A.P. Zisserman, M. Brady, Estimation of the partial volume effect in MRI, Med. Image Anal. 6 (4) (2002) 389–405.

[2] M. Soret, S.L. Bacharach, I. Buvat, Partial-volume effect in PET tumor imaging, J. Nucl. Med. 48 (6) (2007) 932–945.

[3] Q. Min, R.J. Sadleir, An edge-based prediction approach for medical image compression, in: 2012 IEEE-EMBS Conference on Biomedical Engineering and Sciences, IEEE, 2012, pp. 717–722.

[4] A. Trujillo-Pino, K. Krissian, M. Alemán-Flores, D. Santana-Cedrés, Accurate subpixel edge location based on partial area effect, Image Vis. Comput. 31 (1) (2013) 72–90.

[5] S.W. Zucker, R.A. Hummel, A three-dimensional edge operator, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-3 (3) (1981) 324–331.

[6] O. Monga, R. Deriche, 3D edge detection using recursive filtering: Application to scanner images, in: Proceedings CVPR'89: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 1989, pp. 28–35.

[7] L.M. Luo, C. Hamitouche, J.-L. Dillenseger, J.-L. Coatrieux, A moment-based three-dimensional edge operator, IEEE Trans. Biomed. Eng. 40 (7) (1993) 693–703.

[8] P. Bhattacharya, D. Wild, A new edge detector for gray volumetric data, Comput. Biol. Med. 26 (4) (1996) 315–328.

[9] L. Wang, T.-T. Wong, P.A. Heng, J.C.Y. Cheng, Template-matching approach to edge detection of volume data, in: Proceedings International Workshop on Medical Imaging and Augmented Reality, IEEE, 2001, pp. 286–291.

[10] S. Zhan, R. Mehrotra, A zero-crossing-based optimal three-dimensional edge detector, CVGIP Image Underst. 59 (2) (1994) 242–253.

[11] L. Ibanez, C. Hamitouche, C. Roux, Moment-based operator for sub-voxel surface extraction in medical imaging, in: Proceedings of 3rd IEEE International Conference on Image Processing, Vol. 2, IEEE, 1996, pp. 277–280.

[12] C. Hamitouche, C. Roux, J.L. Coatrieux, Design of new surface detection operators in the case of an anisotropic sampling of 3D volume data, in: International Conference on Computer Vision, Virtual Reality, and Robotics in Medicine, Springer, 1995, pp. 523–532.

[13] M. Brejl, M. Sonka, Directional 3D edge detection in anisotropic data: detector design and performance assessment, Comput. Vis. Image Underst. 77 (2) (2000) 84–110.

[14] X. Wu, W.G. Wee, Surface detection with subvoxel accuracy using 3D directional derivatives, in: Nondestructive Evaluation of Aging Aircraft, Airports, and Aerospace Hardware III, Vol. 3586, International Society for Optics and Photonics, 1999, pp. 319–329.

[15] K. Wang, D.-h. Zhang, X.-b. Zhao, K.-d. Huang, Y.-y. Cheng, Surface detection with subvoxel accuracy using facet model and IDDG operator, in: 2006 7th International Conference on Computer-Aided Industrial Design and Conceptual Design, IEEE, 2006, pp. 1–5.

[16] C. Bähnisch, P. Stelldinger, U. Köthe, Fast and accurate 3D edge detection for surface reconstruction, in: Joint Pattern Recognition Symposium, Springer, 2009, pp. 111–120.

[17] H. Bouma, A. Vilanova, L.J. Van Vliet, F.A. Gerritsen, Correction for the dislocation of curved surfaces caused by the PSF in 2D and 3D CT images, IEEE Trans. Pattern Anal. Mach. Intell. 27 (9) (2005) 1501–1507.

[18] Y. Hirano, S. FelsÞ, An edge detection method for three-dimensional gray images using optimum surface fitting, IEICE Tech. Rep. 105 (501) (2006) 305–309.

[19] J.-Y. Lim, I.H.S. Stiehl, On Higher Dimensional Multiscale Edge Extraction, Univ., Bibliothek des Fachbereichs Informatik, 2002.

[20] D.A. Schug, Three Dimensional Edge Detection Using Wavelet and Shearlet Analysis (Ph.D. thesis), University of Maryland, College Park, 2012.

[21] T.-C. Ma, C.-s. Park, K. Suthunyatanakit, M.-j. Oh, T.-w. Kim, M.-j. Kang, Features detection from industrial noisy 3D CT data for reverse engineering, in: Software and Network Engineering, Springer, 2012, pp. 89–101.

[22] J. Yagüe-Fabra, S. Ontiveros, R. Jiménez, S. Chitchian, G. Tosello, S. Carmignato, A 3D edge detection technique for surface extraction in computed tomography for dimensional metrology applications, CIRP Ann. 62 (1) (2013) 531–534.

[23] S. Ontiveros, R. Jiménez, J. Yagüe-Fabra, M. Torralba, Analysis of surface extraction methods based on gradient operators for computed tomography in metrology applications, Materials 11 (8) (2018) 1–16.

[24] J. Stopp, R. Christoph, H. Weise, Surface point determination in subvoxel accuracy from pre-segmented multi-material volume data for metrological applications, 2020.

[25] M. Busch, T. Hausotte, Application of an edge detection algorithm for surface determination in industrial X-ray computed tomography, Prod. Eng. (2022) 1–12.

[26] A. Shah, M.D. Abràmoff, X. Wu, Optimal surface segmentation with subvoxel accuracy in spectral domain optical coherence tomography images, in: Diabetes and Retinopathy, Elsevier, 2020, pp. 69–91.

[27] K.M. van Hespen, J.J. Zwanenburg, J. Hendrikse, H.J. Kuijf, Subvoxel vessel wall thickness measurements of the intracranial arteries using a convolutional neural network, Med. Image Anal. 67 (2021) 101818.

[28] P. Heng, L. Wang, T. Wong, K. Leung, J.C.Y. Cheng, Edge surface extraction from 3D images, in: Medical Imaging 2001: Image Processing, Vol. 4322, International Society for Optics and Photonics, 2001, pp. 407–416.

[29] L. Wang, J. Bai, P. He, P.-A. Heng, X. Yang, A computational framework for approximating boundary surfaces in 3-D biomedical images, IEEE Trans. Inf. Technol. Biomed. 11 (6) (2007) 668–682.

[30] L. Wang, P. Wang, L. Cheng, Y. Ma, S. Wu, Y.-P. Wang, Z. Xu, Detection and reconstruction of an implicit boundary surface by adaptively expanding a small surface patch in a 3D image, IEEE Trans. Vis. Comput. Graphics 20 (11) (2014) 1490–1506.

[31] C. Vinhais, M. Kociński, A. Materka, Centerline-radius polygonal-mesh modeling of bifurcated blood vessels in 3D images using conformal mapping, in: 2018 Signal Processing: Algorithms, Architectures, Arrangements, and Applications, SPA, IEEE, 2018, pp. 180–185.

[32] T. Liu, S. Zhao, Y. Xu, X. Huang, Y. Tian, Q. Wang, Vascular cross-section extraction algorithm based on centerline, in: 2019 9th International Conference on Information Science and Technology, ICIST, IEEE, 2019, pp. 128–134.

[33] D. Kim, H. Kye, J. Lee, Y.-G. Shin, Confidence-controlled local isosurfacing, IEEE Trans. Vis. Comput. Graphics 27 (1) (2020) 29–42.

[34] A. Shah, M.D. Abámoff, X. Wu, Optimal surface segmentation with convex priors in irregularly sampled space, Med. Image Anal. 54 (2019) 63–75.

[35] M.D. Abràmoff, X. Wu, K. Lee, L. Tang, Subvoxel accurate graph search using non-euclidean graph space, PLoS One 9 (10) (2014) 1–13.

[36] S. Ontiveros, J. Yagüe, R. Jiménez, F. Brosed, Computer tomography 3D edge detection comparative for metrology applications, Procedia Eng. 63 (2013) 710–719.

[37] D.A. Hafiz, W.M. Sheta, S. Bayoumi, B.A. Youssef, A new approach for 3d range image segmentation using gradient method, J. Comput. Sci. 7 (4) (2011) 475.

[38] J. Toriwaki, H. Yoshida, Fundamentals of Three-Dimensional Digital Image Processing, Springer Science & Business Media, 2009.

[39] D. Young, Canny Edge Detection in 2-D and 3-D, MATLAB Central File Exchange, 2023, URL: https://es.mathworks.com/matlabcentral/fileexchange/45459-canny-edge-detection-in-2-d-and-3-d.

[40] J.A. Martínez-Mera, P.G. Tahoces, J.M. Carreira, J.J. Suárez-Cuenca, M. Souto, Automatic characterization of thoracic aortic aneurysms from CT images, Comput. Biol. Med. 57 (2015) 74–83.

[41] P.G. Tahoces, L. Alvarez, E. González, C. Cuenca, A. Trujillo, D. Santana-Cedrés, J. Esclarín, L. Gomez, L. Mazorra, M. Alemán-Flores, et al., Automatic estimation of the aortic lumen geometry by ellipse tracking, Int. J. Comput. Assist. Radiol. Surg. 14 (2) (2019) 345–355.