IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 18, 2021, accepted July 13, 2021, date of publication July 19, 2021, date of current version July 28, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3098234

Finding the Largest Volume Parallelepipedon of Arbitrary Orientation in a Solid

RUBÉN MOLANO[1], DANIEL CABALLERO[2], PABLO G. RODRÍGUEZ[2], MARÍA DEL MAR ÁVILA[2], JUAN P. TORRES[2], MARÍA LUISA DURÁN[2], JOSÉ CARLOS SANCHO[2], AND ANDRÉS CARO[2]

[1]Department of Mathematics, School of Technology, University of Extremadura, 10003 Cáceres, Spain
[2]Media Engineering Group (GIM), School of Technology, University of Extremadura, 10003 Cáceres, Spain

Corresponding author: Andrés Caro (andresc@unex.es)

This work was supported by the Junta de Extremadura (European Regional Development Fund), Consejería de Economía, Ciencia y Agenda Digital, under Project GR18138 and Project IB16089. The work of Daniel Caballero was supported by the Junta de Extremadura for the Postdoctoral under Grant PO17017.

ABSTRACT 3D Computer Vision algorithms are a subject of research and application for several industrial processes. The Volume of Interest (VOI) usually refer to sub-objects with basic shapes for computing these algorithms. However, in many cases the objects are available as irregular shapes with many vertices, and in order to apply algorithms effectively, it is essential to compute the largest volume parallelepipedon contained in 3D objects. There are no other approximation algorithms for finding the largest volume parallelepipedon of arbitrary orientation inscribed in a closed 3D contour with a computational cost better than the algorithm proposed in this paper, been $O(n^3)$.

INDEX TERMS Parallelogram, parallelepipedon, polyhedron, volume of interest (VOI).

I. INTRODUCTION

3D Computer vision algorithms are a subject of research and application for several industrial processes. These techniques have been successfully applied in many engineering fields such as industrial image processing, robotics [1], medical image processing [2], and food technology [3]–[5]. Many of these algorithms focus on parts of a 3D object instead of processing the whole object.

3D objects often show irregular and complex shapes with many vertices.Therefore, volume of interest (VOI) is widely used in segmentation tasks, as a significant requirement for later image analysis. In order to avoid time-consuming global segmentation, VOIs allow to focus the processing just in the area of interest. Segmentation and volumetry are essential tasks for many applications, where 3D volume of interest can be used as input for the algorithms which estimate the shape of final objects [6]–[8].

An accurate segmentation of these final objects is often an essential requirement, being mandatory that these final objects are within the VOI, for medical reasons, for industrial engineering purposes, for food technology requirements... Therefore, sometimes, it is not enough to identify an internal VOI at a specific volume, but specifically the largest VOI, for extracting the maximum amount of information possible from that VOI. Accordingly, it is essential to compute the largest volume parallelepipedon contained in 3D objects, as discussed in [9].

The algorithm proposed and developed in this paper comes from the need for a practical application in the field of food technology. The dry-cured meat products, mainly Iberian loin and Iberian ham, constitute products of great importance in the southwestern region of the Iberian Peninsula. This usually targets the dry-cured product market, reaching high quality rates in sensory terms and consumer acceptance [10]. In order to analyze some key factors for their quality, the ripened meat is studied by means of the images obtained by magnetic resonance imaging (MRI), a non-destructive, non-invasive and innocuous method [11]. On these images, active contours techniques are applied to recognize the main muscle structures [12]. The shape of the muscles is complex and irregular, so, they could be represented as closed polyhedral. Then, computational texture features are extracted from these muscles as 3D images, whereas these feature extraction algorithms work better over parallelepipedon VOIs.

Several quality parameters related to dry-cured meat can be computed using computer vision techniques, applied on MRI. The extraction of computer vision features is needed

The associate editor coordinating the review of this manuscript and approving it for publication was Davide Patti.

to properly correlate the results obtained by means of computer vision techniques with the results obtained by means of traditional food technology analysis. The computer vision features must be extracted from the same muscular structures in which the physicochemical analysis was carried out [13]. On the other hand, in other occasions it is necessary to achieve a data set obtained from a VOI. The larger the VOI, the more information can be obtained from it, which can improve the classification or prediction algorithms used later [14], [15]. Therefore, developing an efficient algorithm that allows the computation of the largest possible VOI is really important [16].

Besides the applications for food technology presented in [9], [12], [15], there are different interesting applications for searching the largest rectangle or parallelepipedon in any arbitrary orientation, such as biomedical applications (to detect 2D or 3D regions in MRI, X-ray or computed tomography), or engineering applications (to segment areas of interest in civil engineering or building engineering)... [6]–[8].

The purpose of this paper is to describe our development of a computationally efficient method to obtain the largest VOI contained in 3D object images in order to extract the maximum possible information with a high degree of precision.

As will be presented in the technical issues section, all the source code of the algorithm proposed in this paper, scripts, documentation and non-sensitive data are available for the scientific community in a GitHub repository. The main contributions of this paper can be summarized as follows: i) an efficient algorithm is presented to compute the largest VOI of arbitrary orientation inscribed in 3D volume; ii) the proposed procedure is compared to other similar approaches, evaluating the time complexity and the space complexity, being the new proposal the best option; iii) all the source code, scripts, and documents are provided in a GitHub repository for the scientific community.

The paper is organized as follows: Section 2 presents a review of related works. Section 3 introduces the concept of quasi-lattice polyhedrons (P), and computes the largest volume parallelepipedon contained within (P) by Algorithm 2. Then, Section 4 defines a procedure for computing the largest volume parallelepipedon in an arbitrary solid, and Section 5 and Section 6 describes the feasibility of computing the largest volume parallelepipedon for a practical application and the repository for download the algorithm. Finally, Section 7 sums up the conclusions of our research.

II. RELATED WORKS

Various geometric problems have emerged in last decades relative to polygons and polyhedra. Thus, for polygons many researchers have studied inclusion problems.

Several studies are based on the largest figures, considering shapes other than rectangles or parallelepipeds. DePano *et al.* [17] solved the problem of finding the largest inscribed square in $O(n^2)$ time, selecting the optimal solution from a finite range of solutions in $O(n^2)$ space. Fekete and Sándor [18] provided an algorithm for finding all anchored squares in $O(n \log^2 n)$ time and $O(n \log n)$ space. Melissaratos and Souvaine [19] demonstrated that in $O(n^3)$ time and $O(n * k)$ space, being k the number of vertices, it is possible to compute the largest triangle inscribed in a simple polygon by applying the shortest path among the polygonal regions for the maximization of the triangles areas. Keikha *et al.* [20] computed the largest area triangles from an imprecise set of points, in $O(n^2)$ time, or even in $O(n \log n)$ time for unit segments. They also minimized the largest triangles, in $O(n^4)$ time. However, the final figures were not a quadrilateral for these approaches.

For rectangles, Chazelle *et al.* [21], [22] studied the largest empty area only for inscribed parallel rectangles, in computation time $O(n \log^3 n)$ and $O(n)$ space. Chang and Yap [23] focused on looking for a largest convex polygon that can be defined inside any available simple polygon, with high computation time $O(n^7)$ or $O(n^6)$ if the desired polygon in maximized with respect to perimeter. Aggarwal and Suri [24] found the largest rectangle with largest perimeter within empty shapes, with time complexity $O(n \log^3 n)$, and $O(n)$ memory space. Daniels *et al.* [25] considered the largest parallel rectangle to n given points, improving the time complexity up to $O(n \log^2 n)$ of previous algorithms, with $O(n)$ storage. Like this, Boland *et al.* [26] considered the problem of finding the largest area axis-aligned rectangle contained in a polygon, in $O(n \log n)$ time and space. Knauer *et al.* [27], [28] considered approximation algorithms to calculate the rectangle with the largest area of arbitrary orientation in a convex polygon with n vertices in $O(\frac{1}{\varepsilon} \log \frac{1}{\varepsilon} \log n)$ for simple polygons. In the same way, Cabello *et al.* [29] obtained the largest rectangle in $O(n^3)$ time, and Choi *et al.* [30] considered also the maximun area and perimeter rectangles in $O(n^3 \log n)$ time and $O(n^3)$ space.

Besides, for rectangles, in Molano *et al.* [31] the authors showed how to compute in $O(n^3)$ time and $O(n^3)$ space the largest area rectangle of arbitrary orientation in a closed contour working with Region Of Interests (ROIs). To do this, they first applied the algorithm by Freeman and Shapira [32] to compute the minimum area rectangle that encloses the closed contour. Then, they defined a regular partition leading to build the quasi-lattice polygon S. Finally, they computed the coordinates of the rectangles with the largest area in S. This work was successfully developed in a practical application, previously published in several papers [14], [33]–[35].

It is important to emphasize that Molano *et al.* computed the largest rectangle in any arbitrary orientation. Consequently, as the authors explained in [31] there is a big difference in the complexity order with simple polygons, convex polygons and arbitrary polygons [29]. Years later, Sarkar *et al.* [36] approach also obtained a computation time $O(n^3)$ for arbitrary rectangles, and finally, Abuqasmieh *et al.* [37] presented an unrestricted-shape geometry algorithm which run in $O(n \log^2 n)$ time to find the axis-parallel largest rectangle inside a given region of interest.

In relation to parallelograms, Jin and Matulef [38] were a bit further and found the maximum area parallelogram inside a convex polygon in $O(n^2)$ time and $O(n^2)$ space. Again, the complexity is not comparable with the largest parallelepipedon of any arbitrary orientation complexity [31]. Instead of considering the largest parallelepipedon inscribed within a volume, Ausserhofer *et al.* [39] proposed an algorithm to find maximum area polygons circumscribed about a convex polygon, in $O(n^3)$ time. Although the approach could be considered slightly similar, time complexity is highly dependent on the application.

Table1 summarizes the main algorithms based on quadrilateral approaches, showing the computational cost.

TABLE 1. Computational cost: rectangle/parallelogram.

Ref.	Author	Year	Polygon	Quadrilateral	Computational cost
[21], [22]	Chazelle *et al.*	1984-86	convex	rectangle	$O(n\log^3 n)$
[23]	Chang & Yap	1986	convex	rectangle	$O(n^7)$
[24]	Aggarwal & Suri	1987	convex	rectangle	$O(n\log^3 n)$
[25]	Daniels *et al.*	1997	convex	rectangle	$O(n\log^2 n)$
[26]	Boland & Urrutia	2001	convex	rectangle	$O(n\log n)$
[27], [28]	Knauer *et al.*	2010-12	convex	rectangle	$O(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon}\log n)$
[29]	Cabello *et al.*	2016	convex	rectangle	$O(n^3)$
[30]	Choi *et al.*	2021	convex	rectangle	$O(n^3\log n)$
[31]	Molano *et al.*	2012	arbitrary	rectangle	$O(n^3)$
[36]	Sarkar *et al.*	2018	arbitrary	rectangle	$O(n^3)$
[37]	Abuqasmieh & Alquran	2018	arbitrary	rectangle	$O(n\log^2 n)$
[38]	Jin & Matulef	2011	convex	parallelogram	$O(n^2)$
[39]	Ausserhofer *et al.*	2019	convex	parallelogram	$O(n^3)$
This paper	Molano *et al.*	2021	arbitrary	parallelogram	$O(n^3)$

The algorithm presented in this paper computes the largest volume parallelepipedon in any arbitrary orientation, inscribed in a volume, in $O(n^3)$ time and $O(n^3)$ space. The approach is an enhanced 3D version of the previous 2D approach presented in [31]. By removing the condition of perpendicularity and calculating the absolute value of the determinant of two adjacent sides, the parallelogram could be obtained.

For polyhedral the problem is more complex. It has been solved for polyhedral being circumscribed with simple shapes and minimal volume:

- Spheres: Chien *et al.* [40] computed the circumscribed sphere in $O(n^4)$ time and $O(kn^3)$ space. And Danciger *et al.* [41] developed an algorithm to compute the sphere in $O(n^4)$ time.
- Cylinders: Schömer *et al.* [42] computed the smaller enclosing cylinder in $O(n^4\log n)$ time and $O(n^3)$ space, whereas the approach of Danciger *et al.* [41] implied $O(n^4)$ time.
- Parallelepipedons: Vivien *et al.* in [43] obtained a computational cost $O(n^6)$ for time, although could be $O(n^2)$ time whether the number of vertices is lower than $O(n^5)$.
- Rectangular box: Baraquet and Har-Peled in [44], and Rourke in [45] developed approaches with complexities higher than $O(n^2\log n)$ for time, and $O(n^3)$ for space.
- Polyhedral, being inscribed with maximal volume: regular polyhedron [46] with a computational cost of $O(n^3)$ time and $O(n^3)$ space.

III. LARGEST VOLUME PARALLELEPIPEDON IN A QUASI-LATTICE POLYHEDRON

Given the rectangular box $R = [a,b]\times[c,d]\times[e,f]$, $a,b,c,d,e,f \in \mathbb{Z}$ a `regular partition` Q of R are three ordered collections of equally spaced points which satisfy:

$$Q_1 = \{a = x_0 < x_1 < \ldots < x_r = b\} \in \mathcal{P}([a,b])$$
$$Q_2 = \{c = y_0 < y_1 < \ldots < y_s = d\} \in \mathcal{P}([c,d])$$
$$Q_3 = \{e = z_0 < z_1 < \ldots < z_t = f\} \in \mathcal{P}([e,f])$$

where $Q = Q_1\times Q_2\times Q_3 = \{R_{ijk} = [x_{i-1},x_i]\times[y_{j-1},y_j]\times[z_{k-1},z_k] : 1\le i\le r,\ 1\le j\le s,\ 1\le k\le t\} \in \mathcal{P}(R)$.

We denote $G_L = \{(x_i,y_j,z_k) : 0\le i\le r,\ 0\le j\le s,\ 0\le k\le t\}$, the cube grid composed of points of the partition Q, where L, called `partition size`, is the length of the side of each cube formed by the cube grid. We state that the partition $\dot{Q}$ is finer than the partition Q, if it is verified that all points of Q belong to $\dot{Q}$. We denote $Q \preceq \dot{Q}$.

Let P be a `simple` (surface can be deformed continuously into the surface of a sphere) and `proper polyhedron` (can be represented as the finite union of 3-dimensional lattice simplexes) and supposing that we have defined an algorithm to enumerate all vertices of the polyhedron whose coordinates belong to the cube grid G_L for a regular partition Q. Moreover, if v_i and v_j are two adjacent vertices of P, then they are only connected in the directions, $k\frac{\pi}{4}$, $k = 0,\ldots,7$. A polyhedron P, defined in this way, is said to be a **quasi-lattice polyhedron** (Fig. 1).

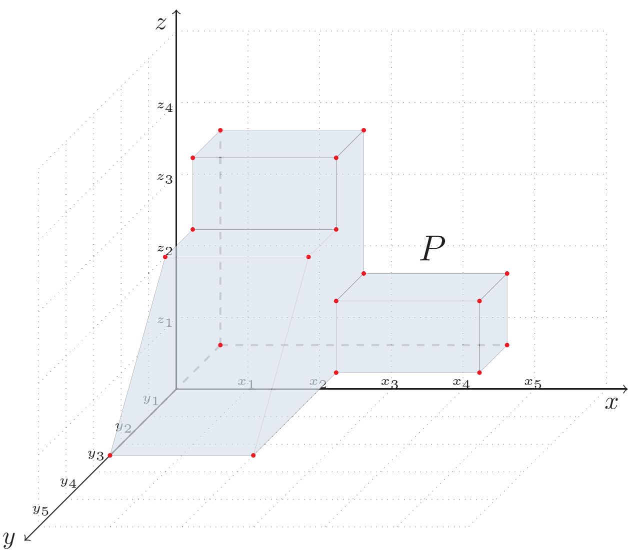


FIGURE 1. Quasi-lattice polyhedron *P* on a regular partition of partition size.

We denote ∂P as the family of boundary nodes of P belonging to the cube grid G_L, and its complementary in P, $\imath P$, the interior points with coordinates in G_L, i.e. $P = \partial P \cup \imath P$. Similarly, we denote V as the family of vertices of ∂p, and its complementary in ∂P, we denote $\imath\partial P$, i.e. $\partial P = V \cup \imath\partial P$.

Then we decompose quasi-lattice polyhedron P as follows:

$$P = V \cup \imath\partial P \cup \imath P = \{p_1, p_2, \ldots, p_{n+m+o}\}$$

where $\#(V) = n$, $\#(\imath\partial P) = m$, $\#(\imath P) = o$ and $\#(P) = N = n + m + o \simeq kn$, $k \in \mathbb{N}$, if # represents the cardinality of the set.

Thus, for Figure 1: $\#(V) = 18$ and $V = \{(x_0, y_0, z_0), (x_4, y_0, z_0), (x_4, y_1, z_0), \ldots\}$.

A. VOLUME OF A QUASI-LATTICE POLYHEDRON

Pick's theorem [47], [48] provides an elegant formula for the area of a lattice polygons (S) from the numbers of points on the boundary (∂S) and in the interior ($\imath S$) belonging to a square grid of side l.

$$A(S) = \left(\#(\imath S) + \frac{\#(\partial S)}{2} - 1\right) \cdot l^2$$

The question now is whether it can be generalized to higher dimensions. Reeve [49], [50] showed that it is not possible to find an expression like the one above to compute the volume of a lattice polyhedron (P) in terms of the number of points, in $\mathbb{Z}^3$, on the boundary (frP) and in the interior ($intP$) of P. However, Kolodziejczyk and Reay [51] obtained a formula (TheoremIII-A) which uses only `rational lattice points`:

$$\mathbb{Z}^3_n = \{x \in \mathbb{R}^3 : nx \in \mathbb{Z}^3\}, \quad n \geq 1 (\mathbb{Z}^3_1 = \mathbb{Z}^3)$$

Theorem 3.1: If P is a proper lattice polyhedron in $\mathbb{R}^N$, then its volume $V(P)$ is given by

$$V(P) = \frac{1}{(N+1)!}\sum_{k=1}^{N}(-1)^{N-k}\binom{N-1}{k-1}(B_k + 2I_k)$$

where $B_k = \#(\mathbb{Z}^3_k \cap frP)$ and $I_k = \#(\mathbb{Z}^3_k \cap intP)$

Thus, in $\mathbb{R}^3$ and for a quasi-lattice polyhedron P:

$$\begin{aligned} V(P) &= \frac{1}{4!}\sum_{k=1}^{3}(-1)^{3-k}\binom{2}{k-1}(B_k + 2I_k) \\ &= \frac{1}{4!}[B_1 + 2I_1 - 2(B_2 + 2I_2) + B_3 + 2I_3] \end{aligned}$$

where $B_k = \#(\mathbb{Z}^3_k \cap \partial P)$ and $I_k = \#(\mathbb{Z}^3_k \cap \imath P)$, $k = 1, 2, 3$.

B. POINT_IN_POLYHEDRON-VECTOR MATRIX OF P

For the reaminder of this work, it is essential to check whether a point p is inside or outside of a polyhedron P, since the largest volume parallelepipedon in a quasi-lattice polyhedron is formed by those points of P in such a way that if we choose two points, p_i and p_j, the segment $\overline{p_ip_j}$ is within P. For the first problem, `point-in-polyhedron`, we used the algorithm proposed by Liu *et al.* [52] with complexity in $O(logn)$ where n is the vertice number of the polyhedron. For the second problem, each segment in P, we define the function `Intersection(`A, B`:point;` P`:polyhedron)` computed in $O(1)$ which returns true if it is verified that A and B as two points of P, then the segment $\overline{AB}$ does not intersect with any edge of the polyhedron.

All this is reflected in Algorithm 1 in $O(n^2logn)$ where the upper triangular matrix W of dimension N satisfies the conditions above.

Algorithm 1 *Function Compute_W(P: Polyhedron) Return W: Matrix*

Input: $P = V \cup \imath\partial P \cup \imath P$,
  $\#(P) = N = n + m + o \simeq kn, k \in \mathbb{N}$
Output: Matrix W of vectors $\overrightarrow{p_ip_j} = p_j - p_i$

```
for i ← 1 to N do
    for j ← 1 to N do
        if i ≥ j then
            W(i, j) ← 0 // Upper triangular:
                avoid calculations.
        else
            if not Intersection(p_i, p_j, P) then
                W(i, j) ← 0
            else
                // p_ip_j is inside or outside
                    of P. We choose a point
                    p.
                p ← (p_i + p_j) div 2 ;
                if not point_in_polyhedron(p, P) then
                    W(i, j) ← 0
                else
                    W(i, j) ← p_j − p_i
```

C. MAIN ALGORITHM

To compute the largest volume parallelepipedon in a quasi-lattice polyhedron $P = \{p_1, p_2, \ldots, p_{n+m+o}\}$ with coordinates belonging to the cube grid G_L, we follow the next process defined in three steps:

`Step1:` Compute the set of all parallelograms contained in the polyhedron P by Algorithm 2 in $O(n^3)$ time.

If $LP = \langle p_i, p_j, p_k, p_s\rangle$ is one of the previous parallelograms then $\overrightarrow{p_ip_j} = \overrightarrow{p_kp_s}$, that is, $p_s = p_j - p_i + p_k$. We define function in $O(1)$ complexity `edge_in_polyhedron(`W`:matrix;` i, j`:integer)` which returns true if the vector $\overrightarrow{p_ip_j}$ is inside the polyhedron. Uses the matrix W defined above in the Algorithm 1.

It is also possible to compute the largest area parallelogram updating the solutions with equal or better area. The complexity order will be same, $O(n^3)$.

`Step2:` Sort in descending order the Parallelograms set by area $\{A_1, \ldots, A_u\}$.

$$\begin{aligned} A_1 &= \{LP^1_1, \ldots, LP^1_{n_1}\} \\ &= \{\langle p^1_{i_1}, p^1_{j_1}, p^1_{k_1}, p^1_{s_1}\rangle, \ldots, \langle p^1_{i_{n_1}}, p^1_{j_{n_1}}, p^1_{k_{n_1}}, p^1_{s_{n_1}}\rangle\} \\ &\vdots \end{aligned}$$

Algorithm 2 Procedure Compute_Parallelograms (in P: Polyhedron; Out Parallelograms: Set ⟨Parallelogram⟩)

Input: $P = V \cup \imath\partial P \cup \imath P$,
$\#(P) = N = n + m + o \simeq kn, k \in \mathbb{N}$

Output: Parallelograms: set⟨parallelogram, A⟩, set of parallelograms LP with area A

for $i \leftarrow 1$ **to** $N - 3$ **do**
  for $j \leftarrow i + 1$ **to** $N - 2$ **do**
    if $edge\_in\_polyhedron(W, i, j)$ **then**
      for $k \leftarrow j + 1$ **to** $N - 1$ **do**
        if $edge\_in\_polyhedron(W, i, k)$ **then**
          $p_s \leftarrow p_j - p_i + p_k$;
          if $edge\_in\_polyhedron(W, k, s)$
          and $edge\_in\_polyhedron(W, j, s)$
          then
            $area \leftarrow |\overrightarrow{p_ip_j} \times \overrightarrow{p_ip_k}|$
              `// cross product`
            LP $\leftarrow (p_i, p_j, p_k, p_s, area)$;
            Parallelograms.insert(LP);

$$A_u = \{LP^u_1, \ldots, LP^u_{n_u}\}$$
$$= \{\langle p^u_{i_1}, p^u_{j_1}, p^u_{k_1}, p^u_{s_1}\rangle, \ldots, \langle p^u_{i_{n_u}}, p^u_{j_{n_u}}, p^u_{k_{n_u}}, p^u_{s_{n_u}}\rangle\}$$

where $area(LP^m_t) = \overline{A_m}, m \in \{1, \ldots, u\}, t \in \{1, \ldots, n_m\}$ and $\overline{A_p} > \overline{A_q}$ if $p < q, \forall p, q \in \{1, \ldots, u\}$

Furthermore, fixed $A_m, m \in \{1, \ldots, u\}$ and chosen two paralelogramos $LP^m_a = \langle p^m_{i_a}, p^m_{j_a}, p^m_{k_a}, p^m_{s_a}\rangle$ and $LP^m_b = \langle p^m_{i_b}, p^m_{j_b}, p^m_{k_b}, p^m_{s_b}\rangle$ with $a < b$ any of the following possibilities is verified:

$$\begin{cases} i_a < i_b \\ i_a = i_b \Rightarrow j_a < j_b \end{cases}$$

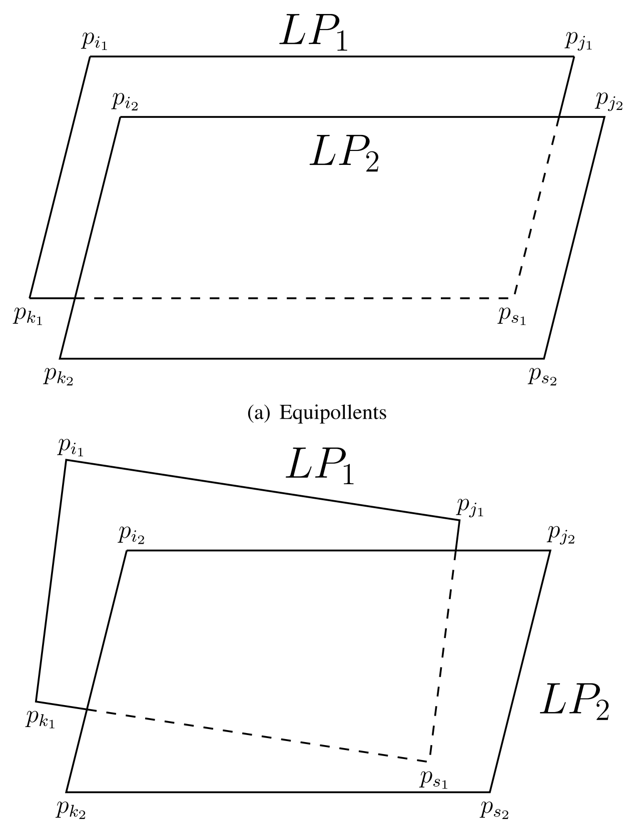
If we denote LP as the set of all parallelograms contained in P, then:

$$\begin{aligned} LP &= \{A_1, \ldots, A_u\} \\ &= \{LP^1_1, \ldots, LP^1_{n_1}, LP^2_1, \ldots, LP^2_{n_2}, \ldots, LP^u_1, \ldots LP^u_{n_u}\} \\ &= \{LP_1, \ldots, LP_{n_1}, LP_{n_1+1}, \ldots, LP_{n_1+n_2}, \ldots, LP_{n_1+\ldots+n_u}\} \\ &= LP_M, \quad M \simeq kn, \ k \in \mathbb{N} \end{aligned}$$

`Step3`: Compute the set of the largest volume parallelepipedon for the polyhedron P.

Definition 3.2: Let $LP_1, LP_2 \in \mathcal{LP}$ and $\pi_1 \equiv A_1x + B_1y + C_1z + D_1 = 0$, $\pi_2 \equiv A_2x + B_2y + C_2z + D_2 = 0$ the planes defined by $LP_1 = \langle p_{i_1}, p_{j_1}, p_{k_1}, p_{s_1}\rangle$ and $LP_2 = \langle p_{i_2}, p_{j_2}, p_{k_2}, p_{s_2}\rangle$ from the triples $(p_{i_1}, \overrightarrow{p_{i_1}p_{j_1}}, \overrightarrow{p_{i_1}p_{k_1}})$ and $(p_{i_2}, \overrightarrow{p_{i_2}p_{j_2}}, \overrightarrow{p_{i_2}p_{k_2}})$, respectively. Then, LP_1 and LP_2 are **equipollents** (Fig. 2) if:

- π_1 and π_2 are parallel, ie, rg $\begin{pmatrix} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \end{pmatrix} = 1$
- $\overrightarrow{p_{i_1}p_{j_1}} = \overrightarrow{p_{i_2}p_{j_2}}$ and $\overrightarrow{p_{i_1}p_{k_1}} = \overrightarrow{p_{i_2}p_{k_2}}$



(a) Equipollents

(b) No equipollent

FIGURE 2. Parallelograms $LP_1, LP_2 \subset P$.

Furthermore, LP_1 and LP_2 are **fundamentals** and we denote $\langle LP_1LP_2\rangle$, if there exists a parallelepipedon $LPP \subset P$ generated from previous parallelograms, ie, $LPP = \langle LP_1LP_2\rangle \subset P$. We denote $Vol(LPP) = |\overrightarrow{p_{i_1}p_{i_2}} \cdot (\overrightarrow{p_{i_1}p_{j_1}} \times \overrightarrow{p_{i_1}p_{k_1}})|$, as the scalar triple product, the volume of parallelepipedon LPP.

We define the function `face_in_polyhedron` $(LP_1, LP_2$`:parallelogram)` computed in $O(1)$ which returns true if LP_1 and LP_2 are `fundamentals`.

Proposition 3.3: Let $LPP_{1,2} = \langle LP_1LP_2\rangle$, $LPP_{2,3} = \langle LP_2LP_3\rangle$ be fundamentals parallelograms. Then, there exists a LPP parallelepipedon such that $Vol(LPP) \geq Vol(LPP_{1,2})$ and $Vol(LPP) \geq Vol(LPP_{2,3})$.

Figure 3 shows one of the possible situations in which LP_1, LP_2, LP_3 may occur.

The function `LPP_Max`$(LPP_a, LPP_b$`: parallelepipedon)`, where $LPP_a = \langle LP^1_aLP^2_a\rangle$ and $LPP_b = \langle LP^1_bLP^2_b\rangle$, computed in $O(1)$ returns the largest volume parallelepipedon from the above parallelograms.

Algorithm 4 computes the largest volume parallelepipedons contained in a quasi-lattice polyhedron P, and runs the set of parallelograms sorted in descending order by area, $\{A_1, \ldots, A_u\}$. If a solution is found, it updates the Parallelepipedons set by the procedure `Update_solution` computed in $O(1)$ (Algorithm 3). The computational cost is

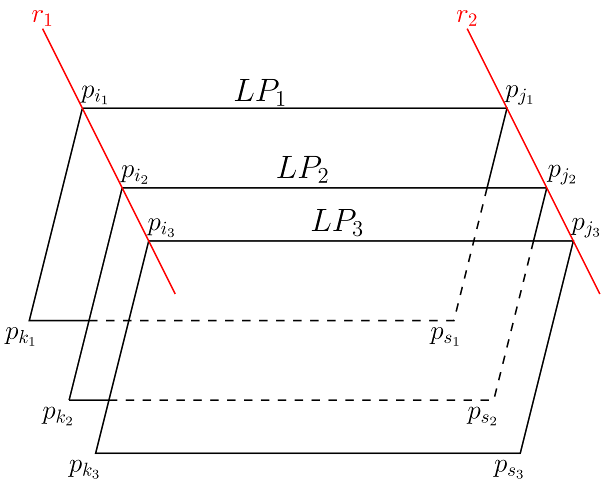


FIGURE 3. **Fundamentals parallelograms with the largest volume parallelepipedon $LPP = \langle LP_1 LP_3 \rangle$.**

Algorithm 3 Procedure Update_Solution (in Volume: Integer; LPP: Parallelepipedon; $\overline{LPP}$: Set⟨Parallelepipedon⟩; Out $\overline{LPP}$: Set⟨Parallelepipedon⟩)

---

if $Vol(LPP) > volume$ **then**
  $volume \leftarrow Vol(LPP)$;
  $\overline{LPP}$.clear(); $\overline{LPP}$.insert(LPP);
else if $Vol(LPP) = volume$ **then**
  $\overline{LPP}$.insert(LPP);

$O(n^3)$ where n is the number of vertices in the quasi-lattice polyhedron P.

IV. APPROXIMATION FOR THE LARGEST VOLUME PARALLELEPIPEDON IN A SOLID

Let solid $S \subset \mathbb{R}^3$ and a way to calculate the largest volume parallelepipedon contained in it has been showed.

First, in Borgefors and Strand [53] there exists a convex body K such that K is the largest convex body enclosed in S. Besides, K admits an inscribed parallelepipedon [54], [55], which ensures the existence of the largest parallelepipedon. Then, we apply the algorithm by Barequet and Har-Peled [44] and we compute the rectangular box R of minimal volume that encloses K.

Definition 4.1: Let K be a convex body and Q a regular partition of the rectangular box R, that encloses K, with partition size L.

We define **lower volume** $\underline{V}(K, Q)$ and **upper volume** $\overline{V}(K, Q)$ as the largest volume quasi-lattice polyhedron $\underline{P}$ contained in K and the smallest volume quasi-lattice polyhedron $\overline{P}$ that encloses K, respectively, and both built by points of G_L. By Theorem III-A:

$$\underline{V}(K, Q) = \frac{1}{4!}\sum_{k=1}^{3}(-1)^{3-k}\binom{2}{k-1}(\underline{B_k} + 2\underline{I_k})$$

$$\overline{V}(K, Q) = \frac{1}{4!}\sum_{k=1}^{3}(-1)^{3-k}\binom{2}{k-1}(\overline{B_k} + 2\overline{I_k})$$

Algorithm 4 Procedure Compute_Largest_Parallelepipedons (in Parallelograms: Set ⟨Parallelogram⟩; Out Parallelepipedons: Set ⟨Parallelepipedon⟩)

Input: $\{A_1, \ldots, A_u\} = \{LP_1, \ldots, LP_{n_1}, \ldots, LP_{n_1+n_2+\ldots+n_u} = LP_M\}$: set⟨parallelogram⟩, set of parallelograms, $M \simeq kn, k \in \mathbb{N}$
Output: Parallelepipedons: set⟨parallelepipedon⟩, set of largest volume parallelepipedons

$Max\_volume \leftarrow 0$; Parallelepipedons.clear();
$a \leftarrow 1$;
while $a \leq M - 1$ **do**
  $b \leftarrow a + 1$;
  while $b \leq M$ ***and*** $\overline{A_a} = \overline{A_b}$ **do**
    if $face\_in\_polyhedron(LP_a, LP_b)$
      `// fundamentals parallelograms`
    then
      Aux.clear();
      $LPP_{a,b} \leftarrow \langle LP_a LP_b \rangle$;
      Aux.insert($LPP_{a,b}$);
      $Update\_solution(Max\_volume, LPP_{a,b}, Parallelepipedons)$;
      $c \leftarrow b + 1$;
      while $c \leq M$ ***and*** $\overline{A_b} = \overline{A_c}$ **do**
        if $face\_in\_polyhedron(LP_b, LP_c)$ **then**
          $LPP_{b,c} \leftarrow \langle LP_b LP_c \rangle$;
          Aux.insert($LPP_{b,c}$);
          $LPP \leftarrow LPP\_Max(Aux)$;
          Aux.clear(); Aux.insert(LPP);
          $Update\_solution(Max\_volume, LPP, Parallelepipedons)$;
        $c \leftarrow c + 1$;
    $b \leftarrow b + 1$;
  $a \leftarrow a + 1$;

---

where $\underline{B_k} = \#(\mathbb{Z}_k^3 \cap \partial \underline{P})$, $\underline{I_k} = \#(\mathbb{Z}_k^3 \cap \imath \underline{P})$, $\overline{B_k} = \#(\mathbb{Z}_k^3 \cap \partial \overline{P})$ and $\overline{I_k} = \#(\mathbb{Z}_k^3 \cap \imath \overline{P})$, $k = 1, 2, 3$.

Clearly, $\underline{V}(K, Q) \leq \overline{V}(K, Q)$.

Applying Algorithm 4, the largest volume parallelepipepon contained in $\underline{P}$ and $\overline{P}$, $\underline{LPP}$ and $\overline{LPP}$ respectively exists, whose volume we denote by $\underline{V_{LPP}}(K, Q)$ and $\overline{V_{LPP}}(K, Q)$.

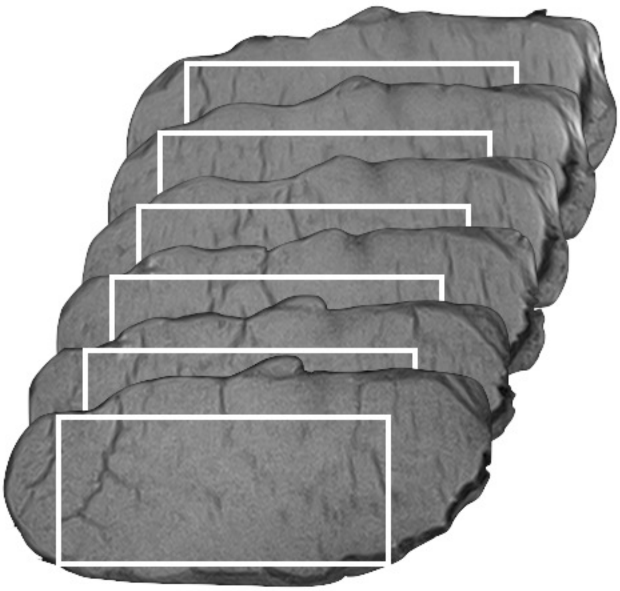
We now denote:

$$\underline{V}(K) = \sup\{\underline{V}(K, Q) : Q \; regularpartition\}$$
$$\overline{V}(K) = \inf\{\overline{V}(K, Q) : Q \; regularpartition\}$$

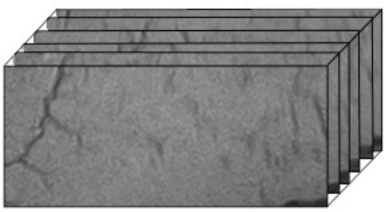
Proposition 4.2: Let K be a convex body.

(a) If Q_1, Q_2 are regular partitions, then $\underline{V}(K, Q_1) \leq \overline{V}(K, Q_2)$.
(b) $\underline{V}(K) = \overline{V}(K)$.

Proposition 4.3: Let a regular partition Q as finer than Q_1 and Q_2. Then, $G_{L_1} \subset G_L$ with $\underline{P_1} \subset \underline{P}$ where $\underline{P_1}$ and $\underline{P}$ are



(a) Select the largest ROI



(b) Select the largest VOI

FIGURE 4. The real case application with food technology (two modules).

the largest quasi-lattice polyhedron contained in K for the regular partitions Q_1 and Q, respectively. Furthermore, $G_{L_2} \subset G_L$ with $\overline{P} \subset \overline{P_2}$ where $\overline{P}$ and $\overline{P_2}$ are the smallest quasi-lattice polyhedron that enclose K for the regular partitions Q and Q_2, respectively. Therefore, $\underline{V}(K, Q_1) \leq \underline{V}(K, Q)$ and $\overline{V}(K, Q) \leq \overline{V}(K, Q_2)$. Then, $\underline{V}(K, Q_1) \leq \overline{V}(K, Q_2)$. This proves (a).

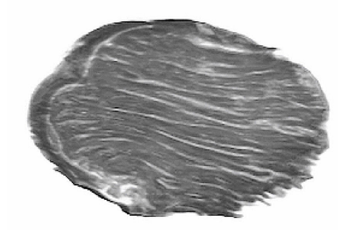
We prove (b). By (a), $\underline{V}(K) \leq \overline{V}(K, \dot{Q})$ for all $\dot{Q}$ regular partition, and as $\overline{V}(K) \leq \overline{V}(K, \dot{Q})$ for all $\dot{Q}$ regular partition, $\underline{V}(K) \leq \overline{V}(K)$. Moreover, as K is a convex body, $\underline{V}(K) = \overline{V}(K)$. The common value $\underline{V}(K) = \overline{V}(K)$ is called **volume of** K and is denoted by $V(K)$.

The following Theorem4.4 shows how the calculation of the largest volume parallelepipedon contained in a solid S can be accomplished by finer partitions and applying Algorithm 4, about the rectangular box R of minimal volume that encloses the largest convex body K enclosed in S.

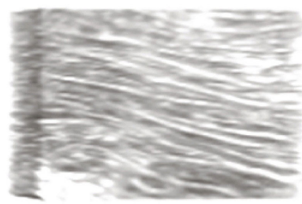
Theorem 4.4: Let K be a convex body. Then, there exists a sequence of regular partitions $\{Q_n\}_{n\in\mathbb{N}}$ with $Q_i \preceq Q_{i+1}$ for all i, such that $\lim_{n\to\infty}(\underline{V_{LPP}}(K, Q_n)) = V_{LPP}(K)$, where $V_{LPP}(K)$ is the largest volume parallelepipedon contained in K.

Proposition 4.5: By Proposition4.2(b), $\underline{V}(K) = \overline{V}(K)$, then there exists regular partitions $\dot{Q}, \ddot{Q}$ such that $|\overline{V}(K, \ddot{Q}) - \underline{V}(K, \dot{Q})| < \varepsilon$ for all $\varepsilon > 0$. We consider a regular partition Q as finer than $\dot{Q}$ and $\ddot{Q}$ at once. Then, $\overline{V}(K, Q) \leq \overline{V}(K, \ddot{Q})$ and $\underline{V}(K, Q) \geq \underline{V}(K, \dot{Q})$. Therefore $|\overline{V}(K, Q) - \underline{V}(K, Q)| \leq |\overline{V}(K, \ddot{Q}) - \underline{V}(K, \dot{Q})| < \varepsilon$.

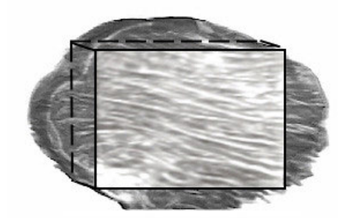
Moreover, as $\underline{V_{LPP}}(K, Q) \leq \underline{V}(K, Q)$ and $\overline{V_{LPP}}(K, Q) \leq \overline{V}(K, Q)$, $|\overline{V_{LPP}}(K, Q) - \underline{V_{LPP}}(K, Q)| < \varepsilon$. Then, there exists a sequence of regular partitions $\{Q_n\}_{n\in\mathbb{N}}$ with $Q_i \preceq Q_{i+1}$ for all i such that $\lim_{n\to\infty}(\overline{V_{LPP}}(K, Q_n) - \underline{V_{LPP}}(K, Q_n)) = 0$ and so, $\lim_{n\to\infty}(\underline{V_{LPP}}(K, Q_n)) = V_{LPP}(K)$.



(a) Iberian loin reconstruction in $3D$



(b) Largest volume parallelepipedon contained in Iberian loin



(c) Largest volume parallelepipedon inscribed in the solid

FIGURE 5. The largest volume parallelepipedon in the meat pieces.

V. TECHNICAL ISSUES

The evaluation and the implementation of the proposed algorithm, were performed by using C++ programming language. All scripts, documentation and non-sensitive data are available in [56]. As previous studies [57], [58] the repository can be downloaded under an LGPL V3 license.

VI. PRACTICAL APPLICATION

It was mathematically proven that it is possible to obtain an approximation algorithm for the problem of finding the largest volume parallelepipedon of arbitrary orientation in a solid. As mentioned above, we have developed a real case application related to food technology, consisting of two modules (Figure 4). The initial module (Figure 4a) aims to select the largest area ROIs for each image according to the method described in [31]. The second module (Figure 4b) computes the VOIs according to the ROIs calculated in the previous step and selects the largest VOI. At this stage, this selection draws up the maximum volume parallelepipedon inscribed in the object. Our paper focuses on these two modules to be followed for feature extraction. The sample needs to be sufficiently representative. The method returns the largest volume parallelepipedon VOI enclosed by the muscle.

Once the largest VOI insided the muscle is determined, different computer vision algorithms for feature extraction are applied to compute quality characteristics of the samples [14], [15], [59]–[62]. This practical application has been successfully applied in previous studies [13], [16]. Figure 5a shows the Iberian loin reconstruction in $3D$ by interpolation of all the images from MRI. Figure 5b identifies the largest volume parallelepipedon contained in Iberian loins, and Figure 5c depicts how this volume can be inscribed in the solid.

VII. CONCLUSION

This paper presents an algorithm for finding the largest volume parallelepipedon of arbitrary orientation in a solid (3D). The new proposal is compared to other similar approaches, achieving the best computational complexities for time and space. The algorithm also is the only approach allowing any arbitrary orientation inside the VOI. Any researcher will be able to use the proposed algorithm since all source code and documentation are provided in a public GitHub repository.

REFERENCES

[1] D. Banerjee, K. Yu, and G. Aggarwal, "Robotic arm based 3D reconstruction test automation," *IEEE Access*, vol. 6, pp. 7206–7213, 2018.

[2] O. Al-Iedani, K. Ribbons, N. Gholizadeh, J. Lechner-Scott, S. Quadrelli, R. Lea, O. Andronesi, and S. Ramadan, "Spiral MRSI and tissue segmentation of normal-appearing white matter and white matter lesions in relapsing remitting multiple sclerosis patients," *Magn. Reson. Imag.*, vol. 74, pp. 21–30, Dec. 2020, doi: 10.1016/j.mri.2020.09.001.

[3] T. Brosnan and D. W. Sun, "Improving quality inspection of food products by computer vision—A review," *J. Food Eng.*, vol. 61, no. 1, pp. 3–16, 2004.

[4] R. Mahendran, G. C. Jayashree, and K. Alagusundaram, "Application of computer vision technique on sorting and grading of fruits and vegetables," *J. Food Process. Technol.*, vol. 10, pp. 2157–7110, Dec. 2012, doi: 10.4172/2157-7110.S1-001.

[5] E. Wari and W. Zhu, "A survey on metaheuristics for optimization in food manufacturing industry," *Appl. Soft Comput.*, vol. 46, pp. 328–343, Sep. 2016.

[6] J. Lee and R. M. Nishikawa, "Cross-organ, cross-modality transfer learning: Feasibility study for segmentation and classification," *IEEE Access*, vol. 8, pp. 210194–210205, 2020.

[7] G. Milani and P. B. Lourenço, "3D non-linear behavior of masonry arch bridges," *Comput. Struct.*, vols. 110–111, pp. 133–150, Nov. 2012, doi: 10.1016/j.compstruc.2012.07.008.

[8] M. Fernández-de Arriba, M. E. Díaz-Fernández, C. González-Nicieza, M. I. Álvarez-Fernández, and A. E. Álvarez-Vigil, "A computational algorithm for rock cutting optimisation from primary blocks," *Comput. Geotechnics*, vol. 50, pp. 29–40, May 2013, doi: 10.1016/j.compgeo.2012.11.010.

[9] M. M. Ávila, M. L. Durán, T. Antequera, R. Palacios, and M. Luquero, "3D reconstruction on MRI to analyse marbling and fat level in Iberian loin," in *Pattern Recognition and Image Analysis* (Lecture Notes in Computer Science), vol. 4477. Heidelberg, Germany: Springer-Verlag, 2007, pp. 145–152.

[10] E. Cernadas, M. L. Durán, and T. Antequera, "Recognizing marbling in dry-cured iberian ham by multiscale analysis," *Pattern Recognit. Lett.*, vol. 23, no. 11, pp. 1311–1321, Sep. 2002.

[11] A. Caro, M. L. Durán, P. G. Rodríguez, T. Antequera, and R. Palacios, "Mathematical morphology on MRI for the determination of Iberian ham fat content," in *Progress in Pattern Recognition, Speech and Image Analysis* (Lecture Notes in Computer Science), vol. 2905. Heidelberg, Germany: Springer-Verlag, 2003, pp. 359–366.

[12] T. Antequera, A. Caro, P. G. Rodríguez, and T. Pérez, "Monitoring the ripening process of iberian ham by computer vision on magnetic resonance imaging," *Meat Sci.*, vol. 76, no. 3, pp. 561–567, Jul. 2007.

[13] M. M. Ávila, D. Caballero, M. L. Durán, A. Caro, T. Pérez-Palacios, and T. Antequera, "Including 3D-textures in a computer vision system to analyze quality traits of loin," in *Computer Vision Systems* (Lecture Notes in Computer Science), vol. 9163. Cham, Switzerland: Springer, 2015, pp. 456–465.

[14] D. Caballero, T. Antequera, A. Caro, J. M. Amigo, B. K. ErsbØll, A. B. Dahl, and T. Pérez-Palacios, "Analysis of MRI by fractals for prediction of sensory attributes: A case study in loin," *J. Food Eng.*, vol. 227, pp. 1–10, Jun. 2018.

[15] D. Caballero, A. Caro, A. B. Dahl, B. K. ErsbØll, J. M. Amigo, T. Pérez-Palacios, and T. Antequera, "Comparison of different image analysis algorithms on MRI to predict physico-chemical and sensory attributes of loin," *Chemometric Intell. Lab. Syst.*, vol. 180, pp. 54–63, Sep. 2018.

[16] M. Ávila, D. Caballero, T. Antequera, M. L. Durán, A. Caro, and T. Pérez-Palacios, "Applying 3D texture algorithms on MRI to evaluate quality traits of loin," *J. Food Eng.*, vol. 222, pp. 258–266, Apr. 2018.

[17] A. DePano, Y. Ke, and J. O'Rourke, "Finding largest inscribed equilateral triangles and squares," in *Proc. 25th Allerton Conf. Commun. Control Comput.*, Montecillo, IL, USA, 1987, pp. 869–878.

[18] S. P. Fekete and P. F. Sándor, "Finding all anchored squares in a convex polygon in subquadratic time," in *Proc. 4th Can. Conf. Comput. Geom.*, St. John's, NF, Canada, 1992, pp. 71–76.

[19] E. A. Melissaratos and D. L. Souvaine, "Shortest paths help solve geometric optimization problems in planar regions," *SIAM J. Comput.*, vol. 21, no. 4, pp. 601–638, Aug. 1992.

[20] V. Keikha, M. Loffler, and A. Mohades, "Largest and smallest area triangles on imprecise points," *Comput. Geometry* 95, Apr. 2021, Art. no. 101742.

[21] B. Chazelle, R. L. Drysdale, and D. T. Lee, "Computing the largest empty rectangle," in *Proc. Annu. Symp. Theor. Aspects Comput. Sci.*, 1984, p. 11.

[22] B. Chazelle, R. L. Drysdale, and D. T. Lee, "Computing the largest empty rectangle," *SIAM J. Comput.*, vol. 15, no. 1, pp. 300–315, Feb. 1986, doi: 10.1137/0215022.

[23] J. S. Chang and C. K. Yap, "A polynomial solution for the potato-peeling problem," *Discrete Comput. Geometry*, vol. 1, no. 2, pp. 155–182, Jun. 1986, doi: 10.1007/BF02187692.

[24] A. Aggarwal and S. Suri, "Fast algorithms for computing the largest empty rectangle," in *Proc. 3rd Annu. Symp. Comput. Geometry (SCG)*, 1987, pp. 278–289, doi: 10.1145/41958.41988.

[25] K. Daniels, V. Milenkoviec, and D. Roth, "Finding the largest area axis-parallel rectangle in a polygon," *Comput. Geometry*, vol. 7, nos. 1–2, pp. 125–148, 1997, doi: 10.1016/0925-7721(95)00041-0.

[26] R. P. Boland and J. Urrutia, "Finding the largest axis-aligned rectangle in a Polygon in O(nlogn) time," in *Proc. 13th Canad. Conf. Comput. Geometry* 2001, pp. 41–44.

[27] C. Knauer, L. Schlipf, J. M. Schmidt, and H. R. Tiwary, "Largest inscribed rectangles in convex polygons," in *Proc. 26th Eur. Workshop Comput. Geometry*, Dortmund, Germany, 2010, pp. 22–25. [Online]. Available: http://page.mi.fu-berlin.de/jeschmid/Knauer2010.pdf

[28] C. Knauer, L. Schlipf, J. M. Schmidt, and H. R. Tiwary, "Largest inscribed rectangles in convex polygons," *J. Discrete Algorithms*, vol. 13, pp. 78–85, May 2012, doi: 10.1016/j.jda.2012.01.002.

[29] S. Cabello, O. Cheong, C. Knauer, and L. Schlipf, "Finding largest rectangles in convex polygons," *Comput. Geometry*, vol. 51, pp. 67–74, Jan. 2016, doi: 10.1016/j.comgeo.2015.08.001.

[30] Y. Choi, S. Lee, and H.-K. Ahn, "Maximum-area and maximum-perimeter rectangles in polygons," *Comput. Geometry*, vol. 94, Mar. 2021, Art. no. 101710, doi: 10.1016/j.comgeo.2020.101710.

[31] R. Molano, P. G. Rodríguez, A. Caro, and M. L. Durán, "Finding the largest area rectangle of arbitrary orientation in a closed contour," *Appl. Math. Comput.*, vol. 218, no. 19, pp. 9866–9874, Jun. 2012.

[32] H. Freeman and R. Shapira, "Determining the minimum-area encasing rectangle for an arbitrary closed curve," *Commun. ACM*, vol. 18, no. 7, pp. 409–413, 1975.

[33] T. Pérez-Palacios, T. Antequera, R. Molano, P. G. Rodríguez, and R. Palacios, "Sensory traits prediction in dry-cured hams from fresh product via MRI and lipid composition," *J. Food Eng.*, vol. 101, no. 2, pp. 152–157, Nov. 2010.

[34] T. Pérez-Palacios, T. Antequera, M. L. Durán, A. Caro, P. G. Rodríguez, and R. Palacios, "MRI-based analysis of feeding background effect on fresh Iberian ham," *Food Chem.*, vol. 126, no. 3, pp. 1366–1372, Jun. 2011.

[35] T. Pérez-Palacios, D. Caballero, T. Antequera, M. L. Durán, M. Ávila, and A. Caro, "Optimization of MRI acquisition and texture analysis to predict physico-chemical parameters of loins by data mining," *Food Bioprocess Technol.*, vol. 10, no. 4, pp. 750–758, Apr. 2017.

[36] A. Sarkar, A. Biswas, M. Dutt, and A. Bhattacharya, "Finding a largest rectangle inside a digital object and rectangularization," *J. Comput. Syst. Sci.*, vol. 95, pp. 204–217, Aug. 2018.

[37] I. Abu-Qasmieh and H. H. Al-Quran, "Unrestricted LR detection for biomedical applications using coarse-to-fine hierarchical approach," *IET Image Process.*, vol. 12, no. 9, pp. 1639–1645, 2018, doi: 10.1049/iet-ipr.2017.0509.

[38] K. Jin and K. Matulef, "Finding the maximum area parallelogram in a convex polygon," in *Proc. 23rd Can. Conf. Comput. Geometry* Toronto, ON, Canada, 2011, pp. 1–6.

[39] M. Ausserhofer, S. Dann, Z. Lángi, and G. Tóth, "An algorithm to find maximum area polygons circumscribed about a convex polygon," *Discrete Appl. Math.*, vol. 255, pp. 98–108, Feb. 2019, doi: 10.1016/j.dam.2018.08.017.

[40] M.-T. Chien and H. Nakazato, "Circumscribed sphere of a convex polyhedron," *Appl. Math. Lett.*, vol. 18, no. 11, pp. 1199–1203, Nov. 2005.

[41] J. Danciger, S. Maloni, and J.-M. Schlenker, "Polyhedra inscribed in a quadric," *Inventiones mathematicae*, vol. 221, no. 1, pp. 237–300, Jul. 2020.

[42] E. Schömer, J. Sellen, M. Teichmann, and Y. Chee, "Smallest enclosing cylinders," *Algorithmica*, vol. 27, no. 2, pp. 170–186, Jun. 2000.

[43] F. Vivien and N. Wicker, "Minimal enclosing parallelepiped in 3D," *Comput. Geometry*, vol. 29, no. 3, pp. 177–190, Nov. 2004.

[44] G. Barequet and S. Har-Peled, "Efficiently approximating the minimum-volume bounding box of a point set in three dimensions," *J. Algorithms* vol. 38, no. 1, pp. 91–109, 2001.

[45] J. O'Rourke, "Finding minimal enclosing boxes," *Int. J. Comput. Inf. Sci.*, vol. 14, no. 3, pp. 183–199, Jun. 1985.

[46] H. T. Croft, "On maximal regular polyhedra inscribed in a regular polyhedron," *Proc. London Math. Soc.*, vols. s3–41, no. 2, pp. 279–296, Sep. 1980.

[47] G. Pick, "Geometrisches zur zahlenlehre, Naturwiss," *Z. Lotos, Prag.*, vol. 8, pp. 311–319, May 1899.

[48] M. Bruckheimer and A. Arcavi, "Farey series and Pick's area theorem," *Math. Intell.*, vol. 17, no. 4, pp. 64–67, 1995.

[49] J. E. Reeve, "A further note on the volume of lattice polyhedra," *J. London Math. Soc.*, vol. 34, no. 1, pp. 57–62, 1959.

[50] J. E. Reeve, "The volume of a lattice polyhedron," *Proc. London Math. Soc.*, vol. 17, pp. 378–395, May 1957.

[51] K. Kołodziejczyk and J. Reay, "Polynomials and spatial pick-type theorems," *Expositiones Mathematicae*, vol. 26, no. 1, pp. 41–53, Feb. 2008.

[52] J. Liu, Y. Q. Chen, J. M. Maisog, and G. Luta, "A new point containment test algorithm based on preprocessing and determining triangles," *Comput.-Aided Des.*, vol. 42, no. 12, pp. 1143–1150, Dec. 2010.

[53] G. Borgefors and R. Strand, "An approximation of the maximal inscribed convex set of a digital object," in *Proc. 13th Int. Conf. Image Anal. Process.*, Cagliari, Italy, 2005, pp. 438–445.

[54] T. Hausel, E. Makai, and A. Szücs, "Polyhedra inscribed and circumscribed to convex bodies," in *Proc. 3rd Int. Workshop Differ. Geometry*, Montpellier, France, 1997, pp. 1–8.

[55] A. Bielecki and K. Radziszewski, "Sur les parallélépipedes inscrits dans les corps convexes," *Ann. Univ. Mariae Curie-Sk lodowska Sect.*, vol. A8, pp. 97–100, Feb. 1954.

[56] *Scripts, Documentation and Non-Sensitive Data*. Accessed: Jul. 18, 2021. [Online]. Available: http://github.com/UniversidadExtremadura/Example-Volume-Extractor

[57] P. García-Sánchez, J. Ortega, J. González, and A. J. J. Castillo Merelo, "Distributed multi-objective evolutionary optimization using island-based selective operator application," *Appl. Soft Comput.*, vol. 85, Dec. 2019, Art. no. 105757, doi: 10.1016/j.asoc.2019.105757.

[58] A. Wojtowicz, P. Zywica, A. Stachowiak, and K. Dyczkowski, "Solving the problem of incomplete data in medical diagnosis via interval modeling," *Appl. Soft Comput.*, vol. 47, pp. 424–437, Oct. 2016.

[59] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.

[60] C. Sun and W. G. Wee, "Neighboring gray level dependence matrix for texture classification," *Comput. Vis., Graph., Image Process.*, vol. 23, no. 3, pp. 341–352, Sep. 1983.

[61] M. M. Galloway, "Texture classification using gray level run length," *Comput. Graph. Image Process.*, vol. 4, no. 2, pp. 172–179, 1975.

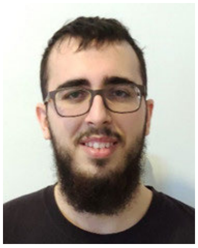
[62] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis and Machine Vision*, 2nd ed. Pacific Grove, CA, USA: PWS Publishing, 1999.

RUBÉN MOLANO received the B.Sc. degree in mathematics, in 1998, and the B.Sc. degree in computer science, in 2009. He is currently pursuing the Ph.D. degree with the Department of Mathematics, University of Extremadura. He has been an Assistant Professor with the Department of Mathematics, University of Extremadura, since 2008. His research interests include the development of statistical analysis in the field of pattern recognition and image analysis.

DANIEL CABALLERO received the M.Sc. degree in computer science engineering in 2013, and the Ph.D. degree *(cum laude)* from the University of Extremadura, in 2017. He is currently a Postdoctoral Researcher with the University of Extremadura. He has participated in numerous research and development projects. He is the coauthor of several SCI journal articles and book chapters. His research interests include the application of non-destructive techniques employing computer vision techniques and data analysis models to obtain quality parameters of meat products. He received the International Mention and the Extraordinary Doctorate Award for his Ph.D. degree from the University of Extremadura.

PABLO G. RODRÍGUEZ received the Ph.D. degree in computer science from the University of Extremadura, in 2000. He was the Director of the School of Technology, Cáceres, from 2017 to 2019, where he teaching engineering studies, such as civil, building, computing, and telecommunications. He is currently the General Director of the Digital Agenda of the Government of the Autonomous Community of Extremadura, Spain. He is also the Doctor in computer science with the first mention of European Thesis with the University of Extremadura. His teaching was mainly centered on subjects of programming and information systems in computer engineering. His research interests include the Internet of Things (IoT), big data and pattern recognition, and image analysis.

MARÍA DEL MAR ÁVILA received the B.Sc. and M.Sc. degrees in computer science from the University of Extremadura, in 1997 and 1999, respectively, and the Ph.D. degree in computer science, in 2018. She has been an Assistant Professor with the Department of Computer and Telematics Systems Engineering, University of Extremadura, since 2002. She has participated in several research projects, being the coauthor of several SCI journal articles. Her research interests include pattern recognition, data mining and machine learning, information retrieval, and cyber security.



JUAN P. TORRES received the B.Sc. degree in computer engineering from the University of Extremadura, Spain, in 2017. He is currently a researcher in a research and development project focused in MRI, and image and data analysis applied to meat products in order to extract and predict quality parameters. His research interests include MRI, image processing, data mining, prediction techniques, data analysis, and computer systems.

MARÍA LUISA DURÁN received the B.Sc. and M.Sc. degrees in computer science from the Polytechnic University of Catalonia and the Ph.D. degree in computer science section from the School of Technology, University of Extremadura, Cáceres, in 2002. She currently works as an Associate Professor with the School of Technology, University of Extremadura. She has participated in several research and development projects. She is the coauthor of numerous research SCI journal articles and book chapters. Her research interests include pattern recognition and image analysis, MRI, big data, and machine learning.

JOSÉ CARLOS SANCHO received the M.Sc. and Ph.D. degrees in computer science from the University of Extremadura, Spain, in 2014 and 2021, respectively. He has been an Assistant Professor with the Department of Computer and Telematics Systems Engineering, University of Extremadura, since 2018. He is the coauthor of several research SCI journal articles. His research interest includes audit and security software development.

ANDRÉS CARO received the B.Sc., M.Sc., and Ph.D. degrees in computer science from the University of Extremadura, Spain, in 1993 and 1998, and 2006, respectively. He has been an Associate Professor with the Department of Computer and Telematics Systems Engineering, University of Extremadura, since 1999. He is currently the Lab Head of the Media Engineering Group, University of Extremadura. He is the coauthor of numerous research SCI journal articles and book chapters. He has participated in the management of both national and international research and development projects and private financing. His research interests include cyber security, big data, machine learning, and pattern recognition.