ACTA UNIVERSITARIA
Multidisciplinary Scientific Journal

| VOLUMEN 29 | 2019 |
ISSN online 2007-9621
http://doi.org/10.15174/au.2019.2122

Partitioning of the pore space based on a non-hierarchical decomposition model

Particionamiento del espacio poroso basado en un modelo de descomposición no-jerárquico

Irving Cruz-Matías[1]*
[1]Departamento de Ciencias Computacionales, Universidad de Monterrey. Av. Ignacio Morones Prieto 4500 Pte., 66238.
E-mail: irving.cruz@udem.edu
*Corresponding author.

Abstract

Bio-CAD and in-silico experimentation currently have a growing interest in biomedical applications where scientific data coming from real samples are used to evaluate physical properties. In this sense, analyzing the pore-size distribution is a demanding task to help interpret the characteristics of porous materials by partitioning it into its constituent pores. Pores are defined intuitively as local openings that can be interconnected by narrow apertures called throats that control a non-wetting phase invasion in a physical method. There are several approaches to characterize the pore space in terms of its constituent pores, several of them requiring prior computation of a skeleton. This paper presents a new approach to characterize the pore space, in terms of a pore-size distribution, which does not require the skeleton computation. Throats are identified using a new decomposition model that performs a spatial partition of the object in a non-hierarchical sweep-based way consisting of a set of disjoint boxes. This approach enables the characterization of the pore space in terms of a pore-size distribution.

Keywords: Porous media; pore-size distribution; skeleton; computer simulation.

Resumen

El bio-diseño asistido por computadora (Bio-CAD) y la experimentación in-silico están teniendo un creciente interés en aplicaciones biomédicas, donde se utilizan datos científicos provenientes de muestras reales para calcular propiedades físicas. En este sentido, analizar la distribución de tamaño de poros es una tarea demandante para ayudar a interpretar las características de materiales porosos, al particionarlo en sus poros constituyentes. Los poros se definen intuitivamente como aperturas locales que pueden estar interconectadas por aberturas llamadas gargantas, que controlan una invasión de fase no-humedad en un método físico. Existen diversos enfoques para caracterizar el espacio poroso en términos de sus poros constituyentes, muchos de ellos requieren el cálculo previo de un esqueleto. Este artículo presenta un nuevo enfoque para caracterizar el espacio poroso, en términos de una distribución del tamaño de poro, que no requiere el cálculo del esqueleto. Las gargantas se identifican utilizando un nuevo modelo de descomposición que realiza una partición espacial del objeto en una forma basada en un barrido no-jerárquico, formado por un conjunto de cajas disjuntas. Este enfoque permite la caracterización del espacio poroso en términos de una distribución de tamaño de poros.

Palabras clave: **Medios porosos; distribución de tamaño de poros; esqueleto; simulación por computadora.**

Recibido: 30 de octubre de 2017 | Aceptado: 8 de febrero de 2019 | Publicado: 25 de septiembre de 2019

Como citar: Cruz-Matías, I. (2019). Partitioning of the pore space based on a non-hierarchical decomposition model. *Acta Universitaria 29*, e2122. doi. http://doi.org/10.15174.au.2019.2122

Introduction

Geometrical and topological representations of the internal structures of samples used in Bio-CAD (bones, biomaterials, rocks and other material samples) are necessary to evaluate their physical properties. Such samples have two disjoint spaces, the pore space and solid space; thus, they can be represented with binary models. The pore space, in turn, is made up of a collection of pores that can be intuitively defined as local openings interconnected by narrow apertures called throats that limit the access to a larger pore (Garboczi, Bentz & Martys, 1999) (figure 1).

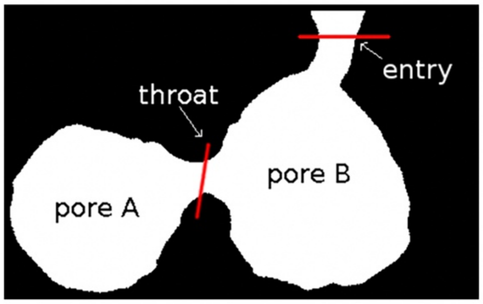


Figure 1. Two pores separated by a throat.
Source: Authors' own elaboration

Usually, porous datasets are considered adequate for analysis if they have their holes homogeneously distributed along the sample, and the pore size is notably smaller than the dataset size (Garboczi *et al*., 1999); that is, they contain complete holes and the object resolution is high enough to clearly define their shape.

The study of porous materials' properties is of great utility in several disciplines. In medicine, they are used to evaluate the degree of osteoporosis and the adequacy of synthetic biomaterial implants for bone regeneration, among other applications. Bone regeneration occurs in the cavities of the implants where blood can flow. Biomaterial implants can be designed as tissue scaffolds, that is, extracellular matrices onto which cells can attach and then grow and form new tissues (Giannitelli, Accoto, Trombetta & Rainer, 2014; León-Mancilla, Araiza-Téllez, Flores-Flores & Piña-Barba, 2016). In geology, the porous morphology and pore sizes are related to the mineral fabric (Klaver, Desbois, Urai & Littke, 2012) and to its oil-bearing and hydrological properties (Silin, Jin & Patzek, 2004).

Likewise, brine inclusions in sea ice can be formalized in terms of their morphology and pore connectivity (Golden *et al*., 2007). Silica sands need to have a high sphericity, in that where the rounder and more spherical the particle is, the more resistant that particle is to crushing or fragmenting (Schroth, Istok, Ahearn & Selker, 1996). In engineering, the durability of cementitious materials is associated with certain mechanical and transport properties that can be evaluated based on the properties of the pore space (Stroeven & Guo, 2006).

There are several methods for describing porous materials. Some of them have been traditionally computed using 2D methodologies, and their results extended to 3D by means of stereologic techniques, while other parameters can be computed directly from the volume dataset.

Background

Pore-size distribution

Pore-size distribution is usually computed with mercury intrusion porosimetry (MIP), an in-vitro experiment based on the capillary law governing liquid penetration into small pores. In this technique, mercury is intruded into the sample at increasing pressures, causing the fluid to flow through smaller apertures.

Virtual porosimetry is an in-silico experimentation for general porous materials that simulates MIP at incremental pressures (Garboczi *et al*., 1999). All separated components of the pore space filled at a given pressure are considered pores and labeled with the diameter corresponding to the applied pressure. This is a flood-fill methodology that uses a previously computed 2D (surface) skeleton as a guide to simulate mercury intrusion from the entry points into the pore space. Moreover, the skeleton is labeled at each point with the corresponding distance (the maximum radius) used to allow or stop the mercury intrusion simulation, i.e., when this radius is smaller than the radius for the current pressure, the simulated intrusion at this pressure stops. Regions corresponding to these smaller radii are the throats. This method is applied to biomaterial samples (Vergés, Ayala, Grau & Tost, 2008b) and soil samples (Delerue, Lomov, Parnas, Verpoest & Wevers, 2003). The latter uses the obtained pore network to compute the permeability. Several methods define the initial set of entry points as those pore voxels which are connected to exterior; however, it can be predefined to allow more freedom to the user in the simulation.

There are other methods that do not require skeleton computation to simulate MIP (Rodríguez, Cruz, Vergés & Ayala, 2011); instead, they use an iterative process that considers the diameters corresponding to pressures. At each iteration, geometric tests detect throats for the corresponding diameter, and a connected-component labeling process collects the region invaded by the mercury.

There are other approaches to compute the pore-size histogram and the pore graph. Some are related to the shape-analysis discipline and use a 2D skeleton as a tool to devise the shape and size of pores. These approaches are heuristic methods that cover the pore space with overlapping spheres, so that the pores are computed as the unions and differences of maximal spheres centered at skeleton points. These methods can be applied to sand samples (Silin *et al*., 2004), bone scaffolds (Cleynenbreugel, 2005), and biomaterial samples (Vergés *et al*., 2008a).

Other approaches based on the 1D (curve)-skeleton computation detect throats as the absolute minima of the skeleton. Thus, pores are defined as the regions limited by throats and solid space (Lindquist & Venkatarangan, 1999). A graph is obtained directly from the 1D skeleton, in which nodes correspond to pores and edges to throats (Liang, Ioannidis & Chatzis, 2000). Alternatively, the minimal cost paths connecting boundary points can be computed instead of skeletons to allow methods based on porosimetry, as well as sphere positioning, to be applied (Schena & Favretto, 2007).

Another technique for computing pore-size histograms is granulometry. This methodology consists in the application of successive morphological openings and does not require the computation of a skeleton (Cnudde, Cwirzen, Maddchaele & Jacobs, 2009; Hilpert & Miller, 2001; Schulz, Becker, Wiegmann, Mukherjee & Wang, 2007; Vogel, Tölke, Schulz, Krafczyk & Roth, 2005). A specific set of discrete spheres with diameters D such that $S(D) \subset S(D+1)$ can be used, thereby ensuring that a larger ball will never reach a cavity in which a smaller ball cannot enter (Hilpert & Miller, 2001). Physical MIP has been compared with a granulometry-based method (Cnudde *et al*., 2009).

Throats can also be detected based on the negative Gaussian curvature of the surface. This method is used to decompose the solid space into single particles and to compute mechanical properties such as grain size and coordination number (number of contacts per grain) (Theile & Schneebeli, 2011).

Representation Models

In most of the reported literature, the operations to study binary volume datasets are performed directly on the classical voxel model. However, in the field of volume analysis and visualization, several alternative models have been devised for specific purposes.

Hierarchical structures such as octrees and kd-trees have been used for Boolean operations (Samet, 1990), connected-component labeling (CCL) (Dillencourt, Samet & Tamminen, 1992), and thinning (Quadros, Shimada & Owen, 2004; Wong, Shih & Su, 2006). Octrees are used as a means of compacting regions and getting rid of the large amount of empty space in the extraction of isosurfaces (Wilhems & Gelder, 1992). Their hierarchy is suitable for multi-resolution when dealing with very large data sets (Andújar, Brunet & Ayala, 2002; LaMar, Hamann & Joy, 1999), as well as to simplify isosurfaces (Vanderhyde & Szymczak, 2008). Kd-trees have been used to extract two-manifold isosurfaces (Greß & Klein, 2004).

There are models that store surface voxels, thereby gaining storage and computational efficiency. The semi-boundary representation affords direct access to surface voxels and performs fast visualization and manipulation operations (Grevera, Udupa & Odhner, 2000). Certain methods of erosion, dilation and CCL use this representation (Thurfjell, Bengtsson & Nordin, 1995). The slice-based binary shell representation stores only surface voxels and is used to render binary volumes (Kim, Seo & Shin, 2001).

In this paper, the Compact Union of Disjoint Boxes (CUDB) decomposition model is used. With this model the geometry is partitioned in a non-hierarchical, sweep-based way. An algorithm has been devised to detect the throats based on this model.

OPP and the CUDB model

A binary voxel model represents an object as the union of its foreground voxels, and its continuous analog is an orthogonal pseudo-polyhedra (OPP) (Lachaud & Montanvert, 2000). OPP have been used in 2D to represent the extracted polygons from numerical control data (Park & Choi, 2001). Some 3D applications of OPP are: general computer graphics applications, such as geometric transformations and Boolean operations (Bournez, Maler & Pnueli, 1999; Esperança & Samet, 1998); skeleton computation (instead of iterative peeling techniques) (Eppstein & Mumford, 2010; Martínez, Pla & Vigo, 2013); orthogonal hull computation (Biedl & Genç, 2011; Biswas, Bhowmick, Sarkar & Bhattacharya, 2012); boundary extraction (Vigo, Pla, Ayala & Martínez, 2012); and model simplification (Cruz-Matías & Ayala, 2014). OPP have been also used in theory of hybrid systems to model the solutions of reachable states (Bournez *et al*., 1999; Dang & Maler, 1998).

The Compact Union of Disjoint Boxes (CUDB) (Cruz-Matías & Ayala, 2017) is a special kind of spatial partitioning representation, where an OPP is decomposed in a list of disjoint boxes. These models can be obtained from the voxel model and other alternative models. The conversions algorithms have been published (Cruz-Matías & Ayala, 2017).

Let P be an OPP and Π_c a plane whose normal is parallel (without loss of generality) to the X axis, intersecting it at $x = c$, where c ranges from $-\infty$ to $+\infty$. Then, this plane sweeps the whole space as c varies within its range, intersecting P at certain intervals. Let us assume that this intersection changes at $c =$

$C_1, \dots, C_n$. More formally, $P \cap \Pi_{C_i - \delta} \neq P \cap \Pi_{C_i + \delta}, \forall i = 1, \dots, n$, where δ is an arbitrarily small quantity. Then, $C_i(P) = P \cap \Pi_{c_i}$ is called a cut of P and $S_i(P) = P \cap \Pi_{C_s}$, for any C_s such that $C_i < C_s < C_{i+1}$, is called a section of P. Figure 2 shows an OPP with its cuts and sections perpendicular to the X axis, since work is done with bounded regions, $S_0(P) = \emptyset$ and $S_n(P) = \emptyset$, where n is the total number of cuts along a given coordinate axis.

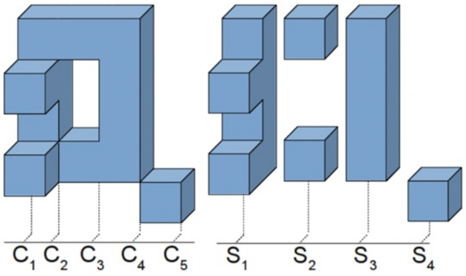


Figure 2. Left: an orthogonal polyhedron with five cuts. Right: its sequence of four prisms with the representative sections (X direction).
Source: Authors' own elaboration.

An OPP can be represented with a sequence of orthogonal prisms represented by their section. Moreover, if the same reasoning is applied to the representative section of each prism, an OPP can be represented as a sequence of boxes. CUDB represents an OPP with such an ordered sequence of boxes in a compact way, as many boxes generated by the aforementioned split process are merged into one in several parts of the model. CUDB is axis-aligned like octrees and bintrees, but the partition is done along the object geometry as in binary space partitioning (BSP). Depending on the order of the axes, chosen to split the data, a 3D object can be decomposed into six different sets of boxes: XYZ, XZY, YXZ, YZX, ZXY, ZYX, and the set will be ordered according to the chosen configuration. Figure 3 illustrates two possible decompositions of the model in figure 2 (left). In 2D, an object can be decomposed into two different sets: XY and YX.

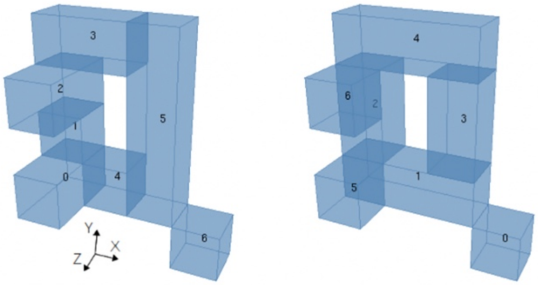


Figure 3. XYZ-CUDB (left) and ZYX-CUDB (right) representation for the model in figure 2, both with seven boxes.
Source: Authors' own elaboration

In the CUDB model, the adjacency information of the boxes is stored. Each box has neighboring boxes in only two orthogonal directions: A- and B-direction, and for each one there are two opposite senses; so, four arrays of pointers to the neighboring boxes (two for each direction) are enough to preserve the adjacency information that is required for future operations. These arrays are defined as A-backward neighbors (ABN), A-forward neighbors (AFN), B-backward neighbors (BBN) and B-forward neighbors (BFN). For more details of this model see (Cruz-Matías & Ayala, 2017).

Methodology

Geometric Pore Space Partition

Unlike traditional approaches, which require the skeleton, the method adopted in this study uses the 2D CUDB-encoded pore space as input. It consists of two different sorted sets of boxes for 2D samples, in which a box represents a cluster of pixels.

The basic steps of our algorithm are:

1. First remove the unreachable regions.
2. Compute all throats with length less than the smaller dimension of the model.
3. Analyze each throat in order to detect the shorter ones that join the solid regions. These resulting throats will partition the pore space.

Remove Unreachable Regions

As in real porosimetry, only those components connected to the exterior (reachable components) are able to be filled, so totally interior components (cavities) will not be considered.

In order to remove the unreachable regions, a CUDB-based connected component labeling is applied, where those boxes that are connected to the exterior are labeled with the same label; boxes with a different label are removed. Figure 4 illustrates the removal process with a 2D example.

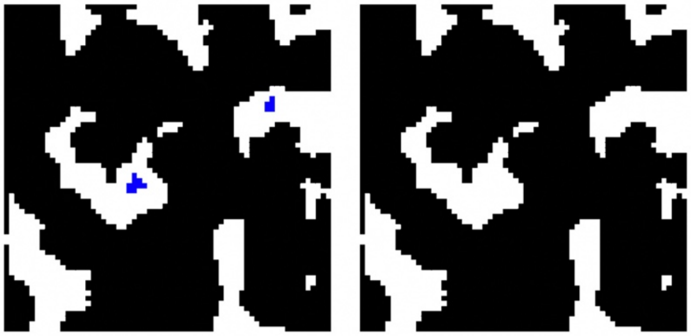


Figure 4. Removal process for a sample. Left: pore space (in black), unreachable regions are marked in blue.
Right: resulting pore space.
Source: Authors' own elaboration.

2D Throats Detection

In the method used in this study, throats can be orthogonal or oblique and all of them are shaped as lines (figure 5). In order to detect orthogonal and oblique throats, the pore space must be exhaustively scanned in the two 2D main directions.

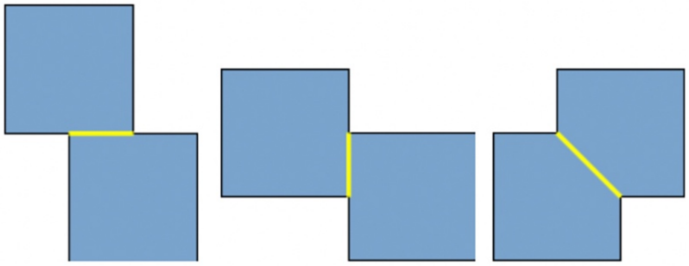


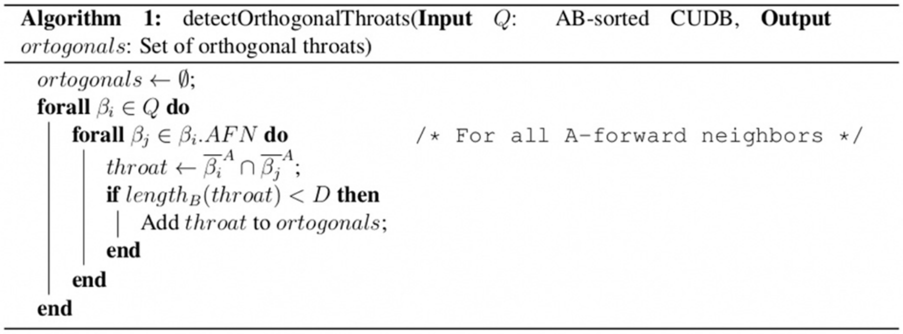
Figure 5. 2D throats configurations (in yellow). Left and middle: Orthogonal throats. Right: Oblique throat.
Source: Authors' own elaboration.

Orthogonal Throats Detection

Boxes in 2D-CUDB have neighboring boxes only in the A-direction; therefore, orthogonal throats can exist in any of these directions. As the main axis that splits the model is A, orthogonal throats are detected in this direction. Thus, the two AB-sorted CUDB encodings of the pore space are required to detect all orthogonal throats.

Let β_i and β_j be two A-adjacent boxes and $\overline{\beta_i}^A$ and $\overline{\beta_j}^A$ the open sets1 of their projections, respectively, over a segment perpendicular to the A-coordinate. There is an orthogonal throat between β_i and β_j if $\left(\overline{\beta_i}^A \cap \overline{\beta_j}^A\right) \neq \emptyset$. This condition can be easily evaluated with the neighborhood information in CUDB.

Note that the orthogonal throat is the intersection of two adjacent boxes, as shown in figure 5 (left) and figure 5 (middle). Algorithm 1 shows the steps of the orthogonal throat detection process for a single AB-sorting (figure 6).



Algorithm 1: detectOrthogonalThroats(**Input** Q: AB-sorted CUDB, **Output** $ortogonals$: Set of orthogonal throats)

```
ortogonals ← $\emptyset$;
forall $\beta_i \in Q$ do
    forall $\beta_j \in \beta_i.AFN$ do          /* For all A-forward neighbors */
        throat ← $\overline{\beta_i}^A \cap \overline{\beta_j}^A$;
        if $length_B(throat) < D$ then
            Add throat to ortogonals;
        end
    end
end
```

Figure 6. Algorithm to detect the ortogonal throats.
Source: Authors' own elaboration.

[1] A set is called an open set if it does not contain any of its boundaries.

Oblique Throats Detection

Note that an oblique 2D throat is an oblique segment. The next process generates the oblique throats for an AB-sorted CUDB which, due to the characteristics of the CUDB model, are the same generated for the corresponding BA-sorted CUDB.

Let β_{i-1}, β_i and β_{i+1} be three consecutive A-adjacent boxes in an AB-sorted CUDB, and let $\overline{\beta_i}^A$ be the open set of the β_i projection over a segment perpendicular to the A-coordinate. Then, an oblique throat exists between β_{i-1} and β_{i+1} if the next conditions are satisfied (figure 7 (a) illustrates these conditions):

1. $\left(\overline{\beta_{i-1}}^A \cap \overline{\beta_{i+1}}^A\right) \neq \emptyset$
2. $\overline{\beta_{i-1}}^A \not\subseteq \overline{\beta_{i+1}}^A$
3. $\overline{\beta_{i-1}}^A \not\supseteq \overline{\beta_{i+1}}^A$
4. $\left(\overline{\beta_{i-1}}^A \cap \overline{\beta_{i+1}}^A\right) \subset \overline{\beta_i}^A$

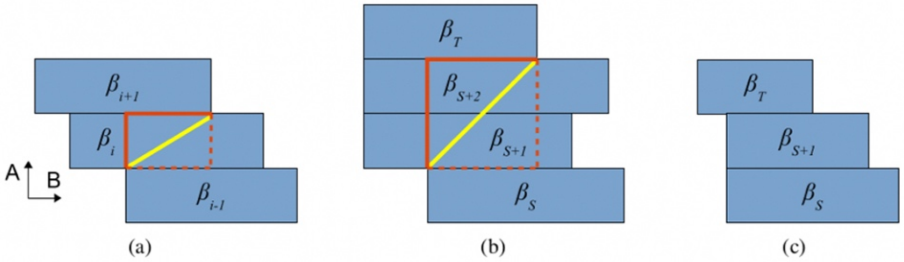


Figure 7. Oblique throats. (a) Simplest case of oblique throat (with just one intermediate box). (b) Oblique throat with more than one intermediate box. (c) Non oblique throat.
Source: Authors' own elaboration.

However, oblique throats are not restricted to three consecutive boxes. Depending on the geometry of the object, an oblique throat can occur between two boxes, β_S and β_T, which have more than one intermediate box between them, as shown in figure 6(b). In these cases, all the previous conditions must be satisfied for β_S and β_T instead of β_{i-1} and β_{i+1}, respectively. Therefore, these conditions are rewritten as:

1. $\left(\overline{\beta_S}^A \subset \overline{\beta_T}^A\right) \neq \emptyset$
2. $\overline{\beta_S}^A \not\subset \overline{\beta_T}^A$
3. $\overline{\beta_S}^A \not\supseteq \overline{\beta_T}^A$
4. $\left(\overline{\beta_S}^A \cap \overline{\beta_T}^A\right) \subset \overline{\beta_i}^A, \forall i \vee S < i < T$

Condition 1 means that $\overline{\beta_S}^A \cap \overline{\beta_T}^A$ must be contained in every intermediate box projection between β_S and β_T. In addition, figure 2 shows why it is necessary to consider open sets for the above conditions.

Otherwise, the aforementioned conditions would be satisfied, and the algorithm would detect oblique throats where none exists.

The previous conditions are used to detect an oblique throat between β_S and β_T. Although a rectangle is obtained (see the rectangle highlighted in orange color in figures 7a and 7b), the strategy to compute the throat is by considering the diagonal included in the mentioned rectangle (see the inclined line highlighted in yellow color in figures 7a and 7b). This strategy allows to separate the pore space in two regions in the current scan direction.

A box β_S can generate several oblique throats with its consecutive boxes in the A-direction (figure 8a). Therefore, in order to detect all the possible oblique throats, all consecutive boxes of β_S must be scanned. The method used here searches for any configuration that satisfies the oblique throat conditions.

Conditions 1 and 3 are required conditions to stop the search process, while the others are not. The search stops in case that Condition 1 is not satisfied, as every subsequent box β_T will not satisfy Condition 4 in the following iterations (figure 7). When Condition 1 is not satisfied, every subsequent box β_T will not either satisfy Condition 2 or 4 in the following iterations (figure 7). Moreover, the search must be also stopped when the distance between $\beta_S.V1$ and $\beta_T.V1$ in the A-coordinate is greater than a certain distance D, this distance has been defined as the smaller dimension of the model. However, when conditions 2 and 4 are not satisfied, there may still be subsequent boxes that define throats that satisfy all conditions (figures 8d and 8e).

Algorithms 2 and 3 show the steps of the oblique throat detection using a recursive strategy for a single AB-sorting (figure 9 and 10). Figure 11 shows an example of throats computation. In this figure, black and white represent the pore and solid spaces respectively.

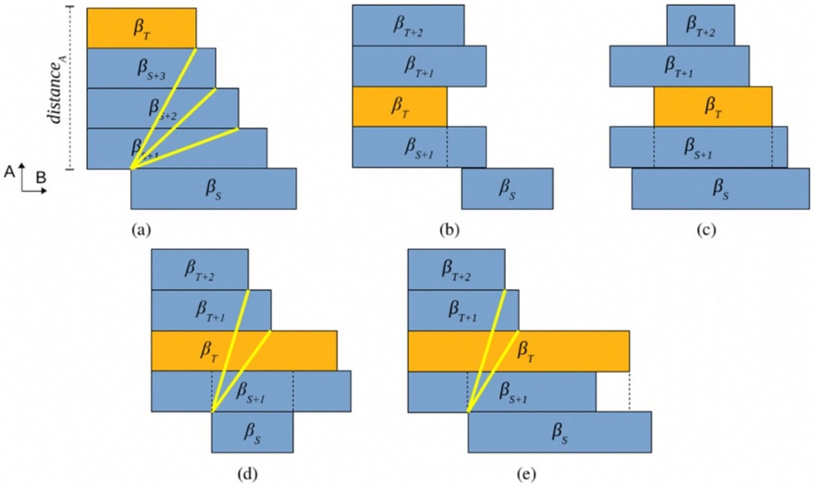
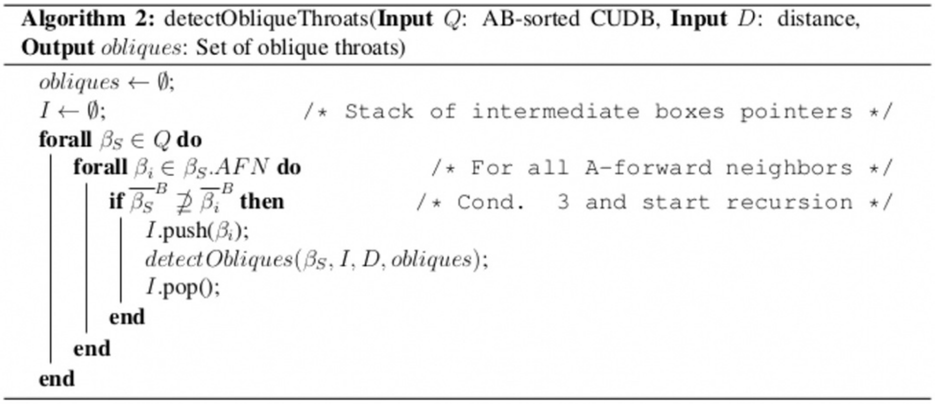


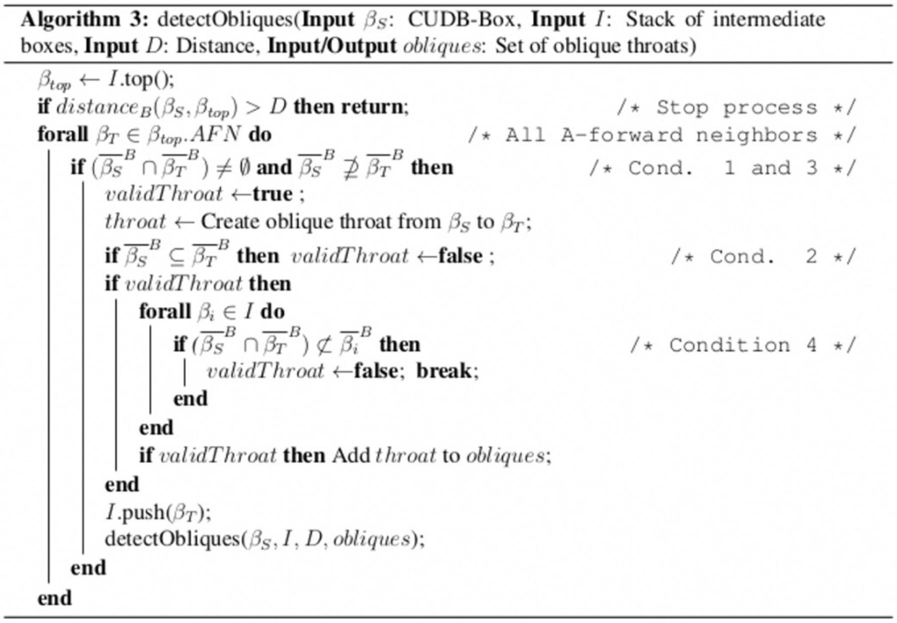
Figure 8. 2D oblique throats scanning process. (a) A box that generates several oblique throats. (b) Condition 1 is not satisfied. (c) Condition 3 is not satisfied. (d) Condition 2 is not satisfied. (e) Condition 4 is not satisfied.
Source: Authors' own elaboration.



```
Algorithm 2: detectObliqueThroats(Input $Q$: AB-sorted CUDB, Input $D$: distance,
Output $obliques$: Set of oblique throats)

obliques ← ∅;
I ← ∅;                           /* Stack of intermediate boxes pointers */
forall $\beta_S \in Q$ do
    forall $\beta_i \in \beta_S.AFN$ do            /* For all A-forward neighbors */
        if $\overline{\beta_S}^B \not\supseteq \overline{\beta_i}^B$ then            /* Cond. 3 and start recursion */
            $I$.push($\beta_i$);
            $detectObliques(\beta_S, I, D, obliques)$;
            $I$.pop();
        end
    end
end
```

Figure 9. Recursive algorithm to detect the oblique throats.
Source: Authors' own elaboration.



```
Algorithm 3: detectObliques(Input $\beta_S$: CUDB-Box, Input $I$: Stack of intermediate
boxes, Input $D$: Distance, Input/Output $obliques$: Set of oblique throats)

$\beta_{top} \leftarrow I$.top();
if $distance_B(\beta_S, \beta_{top}) > D$ then return;                /* Stop process */
forall $\beta_T \in \beta_{top}.AFN$ do                   /* All A-forward neighbors */
    if $(\overline{\beta_S}^B \cap \overline{\beta_T}^B) \neq \emptyset$ and $\overline{\beta_S}^B \not\supseteq \overline{\beta_T}^B$ then             /* Cond. 1 and 3 */
        $validThroat$ ←true ;
        $throat$ ← Create oblique throat from $\beta_S$ to $\beta_T$;
        if $\overline{\beta_S}^B \subseteq \overline{\beta_T}^B$ then $validThroat$ ←false ;                /* Cond. 2 */
        if $validThroat$ then
            forall $\beta_i \in I$ do
                if $(\overline{\beta_S}^B \cap \overline{\beta_T}^B) \not\subset \overline{\beta_i}^B$ then                 /* Condition 4 */
                    $validThroat$ ←false; break;
                end
            end
            if $validThroat$ then Add $throat$ to $obliques$;
        end
        $I$.push($\beta_T$);
        detectObliques($\beta_S, I, D, obliques$);
    end
end
```

Figure 10. Algorithm to detect the obliques throats.
Source: Authors' own elaboration.

Partitioning of the Pore Space

An accurate identification of pore throats is important in the partitioning of the pore space. It is necessary to be careful not to generate too many throats which can produce a large number of small pores, which would correspond primarily to the roughness of the pore wall interface.

From the point of view of the utility of the resulting pore and throats size distributions in network simulations, it is desirable to partition the pore space into nodal pores (pores that are also topological nodes) separated by throat cross-sectional surfaces (Ioannidis, Chatzis & Kwiecien, 1999). This is possible if pore necks are identified as the minimum distance that connect the solid space. In methods like those by Liang *et al*. (2000) and Lindquist & Venkatarangan (1999), throats are identified through a search for minima in the hydraulic radius of individual pore space, where each branch of the skeleton (link) is considered as a separate pore.

Although the method used follows the previous idea, the authors of the study do not rely on prior computation of the skeleton; instead, all the possible throats generated with the strategy described below are computed, then, such throats are analyzed in order to obtain those that better split the pore space.

The strategy to split the pore space is to determine the shortest throats that join the solid regions. Therefore, the first step consists in sorting the detected throats by length. To obtain the solid regions, a CCL process is applied to the CUDB-encoded solid space. For instance, figure 11 shows that this sample has 10 solid regions.

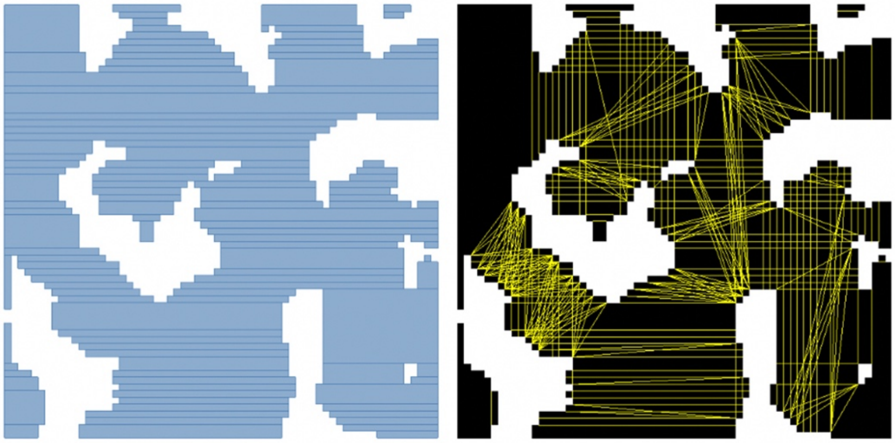


Figure 11. Throats detection for a sample. Left: YX CUDB-encoded pore space. Right: detected orthogonal and oblique throats.
Source: Authors' own elaboration.

Once the throats have been sorted and the solid regions have been detected, it is necessary to determine the final throats that will split the pore space. Let S_A and S_B be two solid regions, each of the ordered throats is analyzed to obtain the pair (S_A, S_B) joined by this throat. This detection is done by evaluating each box of each CUDB-encoded solid region. However, this process can be speeded up by using the bounding box to do previous discards.

Figure 12 (left) shows the resulting throats after the previous strategy. Note in this sample that although there are no throats joining the same solid regions, several intersected throats exist. Some of these throats are longer than some paths formed by other throats. This problem can be solved by applying a Dijkstra's algorithm (Dijkstra, 1959). Every time a throat is found for the pair (S_A, S_B), it is examined whether there is a shorter path formed with previously calculated throats; if this is the case, the throat is omitted. Implementation of these criteria is shown in figure 12 (middle).

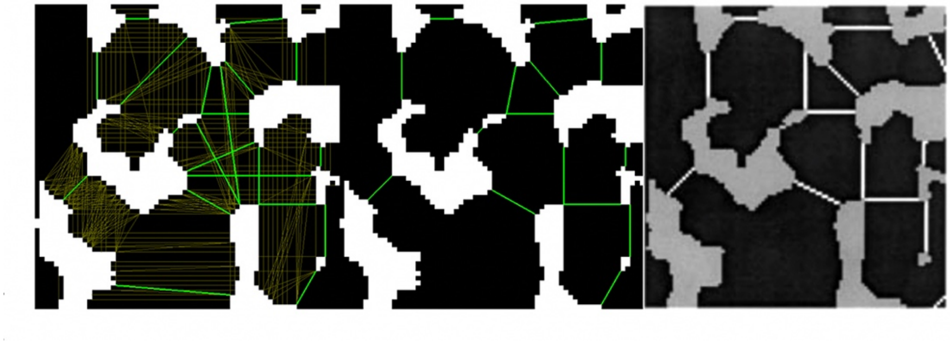


Figure 12. Partitioning of the pore space. Left: partitioning at the resulting throats (21 throats). Middle: Implementation of the final criteria to detect throats (15 throats). Right: Partitioning using a skeleton-based method.
Source: Authors' own elaboration.

Once the final throats have been detected, the object is subdivided along each one of them to partition the pore space into its constituent pores. To achieve this subdivision, a unit-width orthogonal polygon 2D is built for each throat. This OPP is obtained by the 2D scan-conversion of the segment which represent the throat. Therefore, the difference between the OPP-object and the OPP-throat produces the object subdivided along the throat. Figure 13 illustrates the subdivision of the object along two oblique throats.

Once the pores have been correctly separated, their area is straightforward computed using the CUDB-encoded pore.

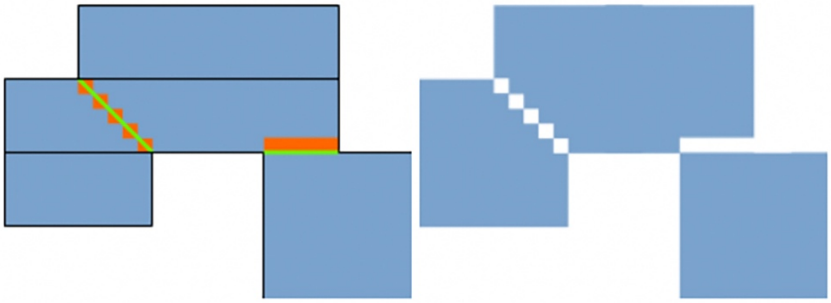


Figure 13. Left: Scan conversion of one orthogonal and one oblique throat. Right: Difference between the object and the scan-converted segments.
Source: Authors' own elaboration.

Results and discussion

The method presented here, like those based on prior computation of the skeleton, is a geometric approach that computes the expected solution.

Figure 11 shows a well-defined structure for which it is easy to determine the theoretical solution. This model has a size of 450 x 450 pixels and contains squares of 85, 58 and 30 pixels per side connected with small lines (throats) of size 2 and 4 pixels. Note in this figure that the method is invariant to rotation to detect the throats and to divide the pore space correctly. For more clarity, the pores after the subdivision along the throats have been colored.

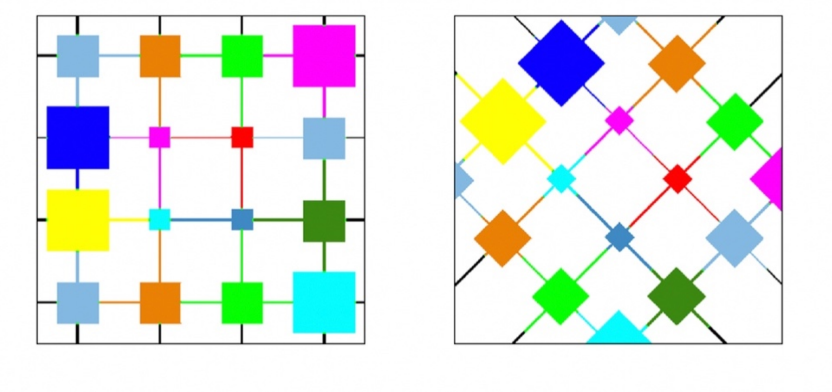
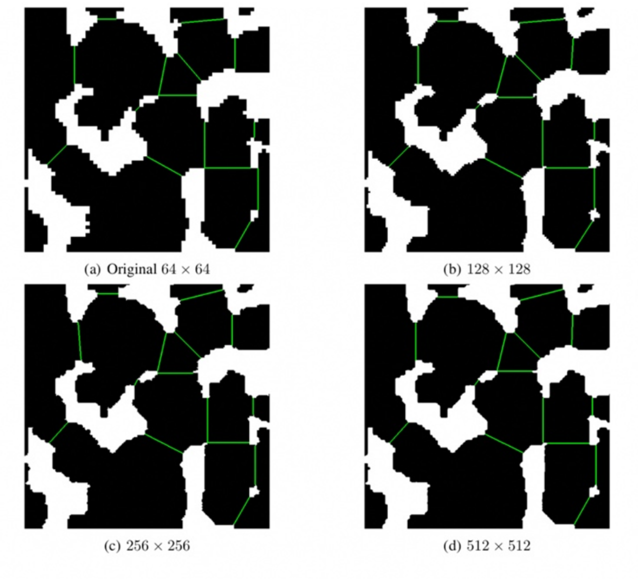


Figure 14. Model of squares. Left: Original 450 × 450. Right: Model rotated 45 clockwise.
Source: Authors' own elaboration.

The adopted approach has been compared with a skeleton-based approach. The model used to explain the partitioning process is the same used in Liang *et al.* (2000). In figure 9 (middle and right) both methods get a quite similar partitioning of the pore space.

Also, the effects of resolution on the method used here have been tested. Resizing the image does not affect the topology of the detected throats. Although the bigger the image, the more throats are generated, the method to detect the final throats allows to obtain an almost identical pore space partition. Figure 15 shows the minor effects with increasing resolution. Of course, the accuracy of measurements is expected to increase as resolution increases.



(a) Original 64 × 64

(b) 128 × 128

(c) 256 × 256

(d) 512 × 512

Figure 15. Effects of increasing the size of the image in the method used in this research.
Source: Authors' own elaboration.

Finally, sections of several real porous samples have been tested, as shown and described in table 1 below. Figures 16 to 19 depict the corresponding partitioned porous space and its pore-size distribution.

- Spheres: A synthetic model with spheres of a variable size.
- PLA: A biomaterial sample consisting of polylactic acid (PLA) with a 16 μm^3 voxel resolution.
- Soygel: Sample of hydroxyapatite with a soy-based foaming agent for bone prosthetics with a 16 μm3 voxel resolution.
- Stone: A stone sample consisting of sedimentary rock from a Lybian oil-bearing unit with a 4.4 μm^3 voxel resolution.

Table 1. Datasets. For each dataset: size of the sample in pixels, run time (in milliseconds) to partition the pore space.

Dataset	Size	Time (ms)
Spheres	450 x 540	96
PLA	314 x 237	56
Soygel	251 x 251	843
Stone	271 x 179	25

Source: Authors' own elaboration.

All the real samples have been scanned by Trabeculae® and segmented by thresholding and applying noise filtering. The corresponding programs have been written in C++ and tested on a PC Intel®Core i7-4600M CPU@2.90GHz with 7.6 GB RAM and running Linux.

Based on the analysis of the figures and the histograms, it can be concluded that the obtained pore size distribution is accurate enough. A more detailed discussion of the comparison of physical and in-silico experimentation is beyond the scope of this paper.

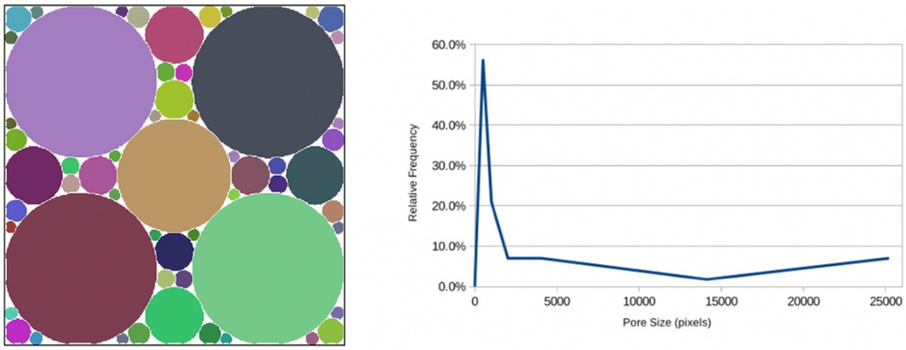


Figure 16. Spheres: A synthetic model. Left: Partitioned porous space. Right: Pore-size distribution.
Source: Author's own elaboration.

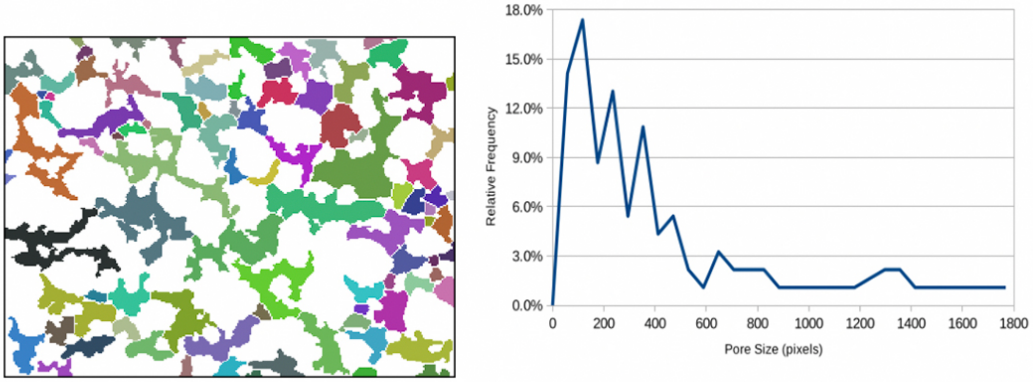


Figure 17. PLA: A biomaterial sample. Left: Partitioned porous space. Right: Pore-size distribution.
Source: Author's own elaboration.

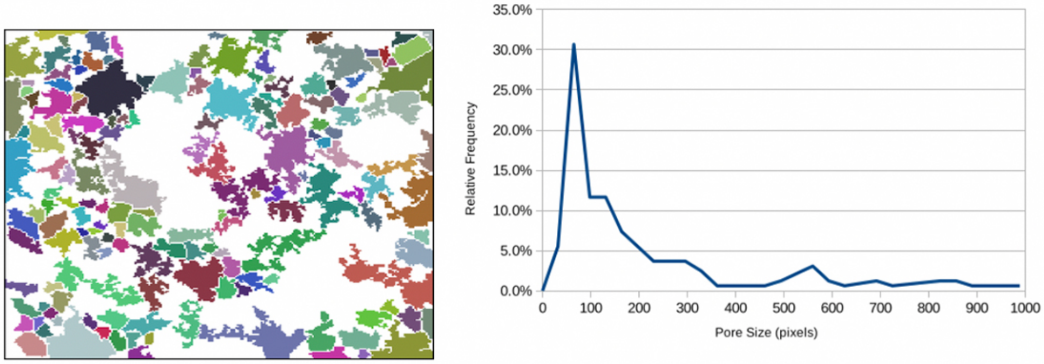


Figure 18. Soygel: Sample of hydroxyapatite. Left: Partitioned porous space. Right: Pore-size distribution.
Source: Author's own elaboration.

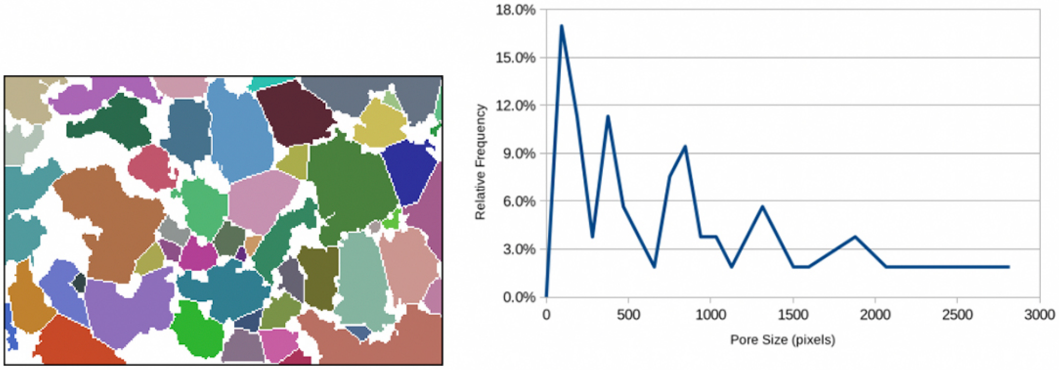


Figure 19. Stone: A stone sample. Left: Partitioned porous space. Right: Pore-size distribution.
Source: Author's own elaboration.

Conclusions

A new method to characterize the pore space has been proposed; it does not require prior computation of the skeleton. As this prior preprocessing is very time-consuming, this approach achieves a noticeable reduction in time. The key feature of the presented method is the process to determine the throats that make it possible to separate regions corresponding to different pores.

The in-silico approaches are based on the geometry of the analyzed samples, and the results obtained are the expected theoretical results, as shown in the results section.

As a future work, several topics must still be studied. At present, the authors are starting to study how to extend this method directly to 3D models and will make contact with biomaterials and geology research teams that provide new problems to study and other kinds of real samples. Further, new methods will be developed for other parameters, such as connectivity and anisotropy. Moreover, the authors of this research are also interested in applying time-varying techniques based on a CUDB model.

References

Andújar, C., Brunet, P., & Ayala, D. (2002). Topology-reducing surface simplification using a discrete solid representation. *ACM Transactions on Graphics*, *21*(2), 88–105. doi: https://doi.org/10.1145/508357.508359

Biedl, T., & Genç, B. (2011). Reconstructing orthogonal polyhedra from putative vertex sets. *Computational Geometry*, *44*(8), 409–417. doi: https://doi.org/10.1016/j.comgeo.2011.04.002

Biswas, A., Bhowmick, P., Sarkar, M., & Bhattacharya, B. B. (2012). A linear-time combinatorial algorithm to find the orthogonal hull of an object on the digital plane. *Information Sciences*, *216*, 176–195. doi: https://doi.org/10.1016/j.ins.2012.05.029

Bournez, O., Maler, O., & Pnueli, A. (1999). Orthogonal polyhedra: Representation and computation. *Hybrid Systems: Computation and Control*, *1569*, 46-60. doi: https://doi.org/10.1007/3-540-48983-5_8

Cleynenbreugel, T. V. (2005). *Porous scaffolds for the replacement of large bone defects: A biomechanical design study* (PhD thesis). Belgium: Katholieke Universiteit Leuven.

Cnudde, V., Cwirzen, A., Maddchaele, B., & Jacobs, P. J. S. (2009). Porosity and microstructure characterization of building stones and concretes. *Engineering geology*, *103*(3-4), 76–83. doi: https://doi.org/10.1016/j.enggeo.2008.06.014

Cruz-Matías, I., & Ayala, D. (2014). A new lossless orthogonal simplification method for 3D objects based on bounding structures. *Graphical Models*, *76*(4), 181–201. doi: https://doi.org/10.1016/j.gmod.2014.01.002

Cruz-Matías, I., & Ayala, D. (2017). Compact union of disjoint boxes: An efficient decomposition model for binary volumes. *Computación y Sistemas, 21*(2), 275-292. doi: http://dx.doi.org/10.13053/cys-21-2-2737

Dang, T., & Maler, O. (1998). Reachability analysis via face lifting. *Hybrid Systems: Computation and Control,* 96-109. doi: https://doi.org/10.1007/3-540-64358-3_34

Delerue, J. F., Lomov, S. V., Parnas, R. S., Verpoest, I., & Wevers, M. (2003). Pore network modeling of permeability for textile reinforcements. *Polymer Composites, 24*(3), 244-357. doi: https://doi.org/10.1002/pc.10034

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, *1*(1), 269-271. doi: https://doi.org/10.1007/BF01386390

Dillencourt, M., Samet, H., & Tamminen, M. (1992). A general approach to connected-component labeling for arbitrary image representations. *Journal of the ACM*, *39*(2), 253–280. doi: https://doi.org/10.1145/128749.128750

Eppstein, D., & Mumford, E. (2010). Steinitz theorems for orthogonal polyhedra. *In Proceedings of the 2010 annual symposium on Computational geometry*, 429-438. doi: https://doi.org/10.1145/1810959.1811030

Esperança, C., & Samet, H. (1998). Vertex representations and their applications in computer graphics. *The Visual Computer*, *14*(5-6), 240-256. doi: https://doi.org/10.1007/s003710050138

Garboczi, E. J., Bentz, D. P., & Martys, N. S. (1999). Digital images and computer modeling. In: P., Wong, T. (Ed.), *Methods in the physics of porous media*, volume 35, (pp. 1–41). Massachusetts, USA: Academic Press.

Giannitelli, S. M., Accoto, D., Trombetta, M., & Rainer, A. (2014). Current trends in the design of scaffolds for computer-aided tissue engineering. *Acta Biomaterialia*, *10*(2), 580-594. doi: https://doi.org/10.1016/j.actbio.2013.10.024

Golden, K. M., Eicken, H., Heaton, A. L., Miner, J., Pringle, D. J., & Zhu, J. (2007). Thermal evolution of permeability and microstructure in sea ice. *Geophysical Research Letters*, *34*(16), 1-6. doi: https://doi.org/10.1029/2007GL030447

Greß, A., & Klein, R. (2004). Efficient representation and extraction of 2-manifold isosurfaces using kd-trees. *Graphical Models*, *66*(6), 370–397. doi: https://doi.org/10.1016/j.gmod.2004.06.010

Grevera, G. J., Udupa, J. K., & Odhner, D. (2000). An order of magnitude faster isosurface rendering in software on a PC than using dedicated, general purpose rendering hardware. *IEEE Transactions Visualization and Computer Graphics*, *6*(4), 335-345. doi: https://doi.org/10.1109/2945.895878

Hilpert, M., & Miller, C. T. (2001). Pore-morphology-based simulation of drainage in totally wetting porous media. *Advances in Water Resources*, *24*(3-4), 243–255. doi: https://doi.org/10.1016/S0309-1708(00)00056-7

Ioannidis, M. A., Chatzis, I., & Kwiecien, M. J. (1999). Computer enhanced core analysis for petrophysical properties. *Journal of Canadian Petroleum Technology*, *38*(03). doi: https://doi.org/10.2118/99-03-01

Kim, B. H., Seo, J., & Shin, Y. G. (2001). Binary volume rendering using Slice-based Binary Shell. *The Visual Computer*, *17*(4), 243–257. doi: https://doi.org/10.1007/s003710100108

Klaver, J., Desbois, G., Urai, J. L., & Littke, R. (2012). BIB-SEM study of the pore space morphology in early mature Posidonia Shale from the Hils area, Germany. *International Journal of Coal Geology*, *103*(1), 12-25. doi: https://doi.org/10.1016/j.coal.2012.06.012

Lachaud, J., & Montanvert, A. (2000). Continuous analogs of digital boundaries: A topological approach to iso-surfaces. *Graphical Models*, *62*(3), 129–164. doi: https://doi.org/10.1006/gmod.2000.0522

LaMar, E., Hamann, B., & Joy, K. (1999). Multiresolution techniques for interactive texture-based volume visualization. *In Proceedings of the conference on Visualization '99: celebrating ten years*, (pp. 355-361). California, USA: IEEE Computer Society Press. doi: https://dl.acm.org/citation.cfm?id=319432

León-Mancilla, B. H., Araiza-Téllez, M. A., Flores-Flores, J. O., & Piña-Barba, M. C. (2016). Physico-chemical characterization of collagen scaffolds for tissue engineering. *Journal of Applied Research and Technology, 14*(1), 77-85. Doi: http://dx.doi.org/10.1016/j.jart.2016.01.001

Liang, Z., Ioannidis, M. A., & Chatzis, I. (2000). Geometric and topological analysis of three-dimensional porous media: Pore space partitioning based on morphological skeletonization. *Journal of Colloid and Interface Science*, *221*(1), 13–24. doi: https://doi.org/10.1006/jcis.1999.6559

Lindquist, W. B., & Venkatarangan, A. (1999). Investigating 3D geometry of porous media from high resolution images. *Physics and Chemistry of the Earth, Part A: Solid Earth and Geodesy, 24*(7), 593–599. doi: https://doi.org/10.1016/S1464-1895(99)00085-X

Martínez, J., Pla, N., & Vigo, M. (2013). Skeletal representations of orthogonal shapes. *Graphical Models, 75*(4), 189–207. doi: https://doi.org/10.1016/j.gmod.2013.03.005

Park, S. C., & Choi, B. K. (2001). Boundary extraction algorithm for cutting area detection. *Computer-Aided Design*, *33*(8), 571-579. doi: https://doi.org/10.1016/S0010-4485(00)00101-9

Quadros, W. R., Shimada, K., & Owen, S. J. (2004). 3D discrete skeleton generation by wave propagation on PR-octree for finite element mesh sizing. *In Proceedings of ACM Symposium on Solid Modeling and Applications*, (pp. 327–332). Genoa, Italy: Eurographics Association. doi: https://dl.acm.org/citation.cfm?id=1217927

Rodríguez, J. E., Cruz, I., Vergés, E., & Ayala, D. (2011). A connected-component-labeling-based approach to virtual porosimetry. *Graphical Models*, *73*(5), 296-310. doi: https://doi.org/10.1016/j.gmod.2011.06.001

Samet, H. (1990). *Applications of spatial data structures: Computer graphics, image processing, and GIS*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Schena, G., & Favretto, S. (2007). Pore space network characterization with sub-voxel definition. *Transport in Porous Media*, *70*(2), 181–190. doi: https://doi.org/10.1007/s11242-006-9094-z

Schroth, M. H., Istok, J. D., Ahearn, S. J., & Selker, J. S. (1996). Characterization of miller-similar silica sands for laboratory hydrologic studies. *Soil Science Society of America Journal*, *60*(5), 1331-1339. doi: https://doi.org/10.2136/sssaj1996.03615995006000050007x

Schulz, W., Becker, J., Wiegmann, A., Mukherjee, P. P., & Wang, C. (2007). Modeling of two-phase behavior in the gas diffusion medium of PEFCs via full morphology approach. *Journal of The Electrochemical Society*, *154*(4), B419-B426. doi: https://doi.org/10.1149/1.2472547

Silin, D. B., Jin, G., & Patzek, T. W. (2004). Robust determination of the pore space morphology in sedimentary rocks. *Society of Petroleum engineers*, pp. 69-70. doi: https://doi.org/10.2118/84296-MS

Stroeven, P., & Guo, Z. (2006). Modern routes to explore concrete's complex pore space. *Image Analysis Stereology*, *25*(2), 75-86. doi: https://doi.org/10.5566/ias.v25.p75-86

Theile, T., & Schneebeli, M. (2011). Algorithm to decompose three-dimensional complex structures at the necks: Tested on snow structures. *Image Processing, IET*, *5*(2), 132-140. doi: https://doi.org/10.1049/iet-ipr.2009.0410

Thurfjell, L., Bengtsson, E., & Nordin, B. (1995). A boundary approach to fast neighborhood operations on three-dimensional binary data. *Graphical Models and Image Processing*, *57*(1), 13–19. doi: https://doi.org/10.1006/gmip.1995.1002

Vanderhyde, J., & Szymczak, A. (2008). Topological simplification of isosurfaces in volumetric data using octrees. *Graphical Models*, *70*(1-2), 16–31. doi: https://doi.org/10.1016/j.gmod.2007.06.002

Vergés, E., Ayala, D., Grau, S., & Tost, D. (2008a). 3D reconstruction and quantification of porous structures. *Computers & Graphics, 32*(4), 438-444. doi: https://doi.org/10.1016/j.cag.2008.04.001

Vergés, E., Ayala, D., Grau, S., & Tost, D. (2008b). Virtual porosimeter. *Computer-Aided Design and Applications*, *5*(1-4), 557-564. doi: https://doi.org/10.3722/cadaps.2008.557-564

Vigo, M., Pla, N., Ayala, D., & Martínez, J. (2012). Efficient algorithms for boundary extraction of 2D and 3D orthogonal pseudomanifolds. *Graphical Models*, *74*(3), 61-74. doi: https://doi.org/10.1016/j.gmod.2012.03.004

Vogel, H. J., Tölke, J., Schulz, V. P., Krafczyk, M., & Roth, K. (2005). Comparison of a lattice-boltzmann model, a full-morphology model, and a pore network model for determining capillary pressure-saturation relationships. *Vadose Zone Journal*, *4*(2), 380-388. doi: https://doi.org/10.2136/vzj2004.0114

Wilhems, J., & Gelder, A. V. (1992). Octrees for faster isosurface generation. *ACM Transactions on Graphics*, *11*(3), 201-227. doi: https://doi.org/10.1145/130881.130882

Wong, W., Shih, F. Y., & Su, T. (2006). Thinning algorithms based on quadtree and octree representations. *Information Sciences*, *176*(10), 1379–1394. doi: https://doi.org/10.1016/j.ins.2005.04.001