

# Skeletonless Porosimeter Simulation

Jorge Rodríguez <sup>1</sup>, Irving Cruz <sup>2</sup>, Eduard Vergés <sup>2,3</sup>, Dolors Ayala <sup>2,3</sup>

<sup>1</sup>CEMVICC, Universidad de Carabobo, Valencia, Venezuela

<sup>2</sup>Universitat Politècnica de Catalunya, Barcelona

<sup>3</sup>Institut de Bioenginyeria de Catalunya, Barcelona

jrodrigu@uc.edu.ve, (icruz,everges,dolorsa)@lsi.upc.edu

## Abstract

We introduce a new approach to simulate a virtual mercury intrusion porosimetry (MIP) using neither skeleton computing nor seed-growing methods. Most of the existing methods to determine local pore sizes in a porous medium require to compute the skeleton of the pore space. However, the skeleton computation is a very time consuming process. Instead, our approach uses a particular spatial enumeration encoding of the porous media, a set of disjoint boxes, and an algorithm able to determine the set of boxes invaded by the mercury at each iteration without any need of a previous skeleton computation. The algorithm detects all the pores which must be filled for a given mercury intrusion pressure, which is related to a diameter by the Washburn equation. The presented method is able to detect narrow throats and one-dimensional transitions between pores in order to prevent incorrect full fluid invasion of the whole sample. The particular encoding used in this work is a new compact version of an existing model, the Ordered Union of Disjoint Boxes (OUDB). Finally, the pore size distribution of the porous medium and the corresponding pore graph can be obtained from the analyzed sample.

## 1 Introduction and related work

The geometrical and topological representation of the internal structure of samples as biomaterials, scaffolds and rocks is necessary for the evaluation of their physical properties. For instance, to compute the permeability of such materials, a graph can be devised of the porous space of the sample.

Classical experimental methods use mercury intrusion porosimetry (MIP) based on the capillarity law governing penetration of non-wetting liquid into porous regions. With this technique, mercury is intruded at different pressures (which are related to diameters by the Washburn equation) into the pore space. Higher pressures allow mercury to invade smaller cavities. Pore size distribution and permeability can be computed using this technique.

Nowadays non-destructive computer-based methods are being used to evaluate structural parameters. They use images captured with Computer Tomography (CT or  $\mu$ CT) to construct 3D virtual models of the samples and apply virtual processes. There exist approaches that simulate MIP [5] [6]. As each pressure corresponds to a diameter, these approaches consider pores as cylinders and devise a cylinder network. Then they apply the Poiseuille law to compute

the conductivity of all the cylinders and, from them and the network, the global permeability of the sample can be computed. Most approaches use directly the voxel model. In [8] alternative models EVM (extreme vertices model) and OUIDB (ordered union of disjoint boxes) are used to simulate MIP. However all these approaches rely on a previous skeleton computation of the pore space.

There are other approaches that analyze the pore space graph. Granulometry-based approaches apply successive openings [3] to compute pore size distribution while approaches based on sphere-filling heuristics [4] [7] obtain also a pore graph.

In this work we present a method to simulate MIP. The novelty of this method is that it does not need to previously compute the skeleton of the model as all the MIP-based methods do. Our approach simulates mercury intrusion as a labeling process of the connected regions of the pore space. Depending on the input pressure, the mercury entry points and the connectivity of the pore space, its different OUIDB-encoded regions are labeled as invaded or non-invaded. Moreover, this method uses a compact variant of the OUIDB partitioning, called CUDB (compact union of disjoint boxes), which can work in 3D as well as in 2D with a reduced number of elements.

Next section reviews the OUIDB models and presents the CUDB model. Section 3 explains the proposed method. Section 4 details how to compute the pore map from the result obtained with the virtual MIP method. Finally, Section 5 discusses the obtained results and Section 6 presents the conclusions .

## 2 OUIDB and CUDB models

### 2.1 The OUIDB model

Let  $P$  be an OPP and  $\Pi_c$  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 some intervals. Let us assume that this intersection changes at  $c = C_1, \dots, C_n$ . More formally,  $P \cap \Pi_{C_i - \delta} \neq P \cap \Pi_{C_i + \delta}, i = 1, \dots, n$ , where  $\delta$  is an arbitrarily small quantity. Then,  $C_i(P) = P \cap \Pi_{C_i}$  is called a *cut* of  $P$  and  $S_i(P) = P \cap \Pi_{C_s}, C_i < C_s < C_{i+1}$ , is called a *section* of  $P$ . Figure 1 (top, left) shows an OPP with its *cuts* and *sections* perpendicular to the X axis.

Since we work with bounded regions,  $S_0(P) = P \cap \Pi_{-\infty} = \emptyset$  and  $S_n(P) = P \cap \Pi_{+\infty} = \emptyset$ ,  $n$  being the total number of *cuts* along a given coordinate axis.

*Sections* and *cuts* are related by the two following expressions:

$$S_i(P) = S_{i-1}(P) \otimes C_i(P)$$

$$C_i(P) = S_{i-1}(P) \otimes S_i(P)$$

for  $i = 1 \dots n$ , where  $\otimes$  denotes the regularized XOR operation.

Therefore, an OPP can be represented with a sequence of orthogonal prisms represented by its section  $S_i$  (see Figure 1 (top, right)). Moreover, if we apply the same reasoning to each prism representative section, an OPP can be represented as a sequence of boxes.

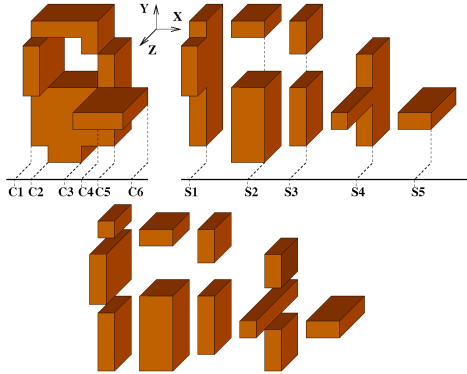


Figure 1: Top, left: an orthogonal pseudo-polyhedron with 6 cuts. top, right: its sequence of 5 prisms with the representative section (X direction). Bottom: Its XY-OUIDB representation

The OUIDB (Ordered Union of Disjoint Boxes) model represents OPP with such sequence of boxes. It is a particular partitioning and can be thought as a spatial enumeration model. The OUIDB is axis aligned as octrees and bintrees but the partition is done along the object geometry like BSP.

Depending on the axes order we choose to split the data, a 2D object can be decomposed into two different sets of boxes: X- and Y-OUIDB; and a 3D object can be decomposed into six different sets of boxes XY-; XZ-; ZX-; ZY-; YX- and YZ-OUIDB. See Figure 1, bottom.

For more details concerning OPP and OUIDB, see references [1] and [2].

## 2.2 Compact Union of Disjoint Boxes (CUDB)

In this work we have devised a new version of the OUIDB model which we call CUDB. The CUDB model is similar to the OUIDB. It is also a union of disjoint boxes but a more compact one as several contiguous boxes are merged into one in several parts of the model. The CUDB model is suitable for the virtual MIP method presented in this paper but it won't be suitable for processes that require an axis-ordered traversal of the boxes because the ordering of the boxes has been lost in the compactness process.

Let  $P$  be an OPP and let  $BB(P)$  be the bounding box of  $P$ . To obtain the OUIDB partitioning perpendicular to a main axis, say  $X$  for instance,  $BB(P)$  is split along the  $Y$ -axis at every *cut* of  $P$ . Thus,  $C_i$  splits all the geometry of  $P$  along its  $y$ -coordinate. Therefore, some local regions of  $P$ , where  $C_i$  has no relationship with it, are further divided unnecessarily (see Figure 2)

This constraint is mandatory to maintain an order among the resulting boxes. However, in order to subdivide just the pieces of  $P$  related with the *cut* which leads the splitting, this constraint can be relaxed. More formally, let A-OUIDB( $P$ ) be the OUIDB partitioning of an OPP  $P$ . Let  $b_i$  and  $b_j$  be two adjacent boxes in A-OUIDB( $P$ ) and let  $Sb_i$  and  $Sb_j$  be the transversal sections according to the A-axis of  $b_i$  and  $b_j$  respectively, then  $b_i$  and  $b_j$  can be compacted as only one box if  $Sb_i = Sb_j$ . Figure 2 shows some boxes with identical Y-sections which can be compacted. Figure 3 illustrates the OUIDB and

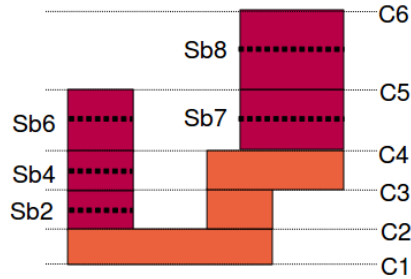


Figure 2: a)Y-OUDB of an OPP. Fine dotted lines denote splitting *cuts*. Thick dotted lines are the transversal sections of the darker boxes. Boxes with sections Sb2, Sb4 and Sb6, and boxes with sections Sb7 and Sb8 can be compacted.

CUDB models for the same OPP. Although the order among the boxes is lost, maintaining the neighborhood information is easy with a tiny storage effort.

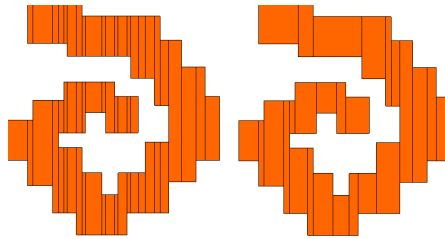


Figure 3: a)Y-OUDB of a 2D object (55 boxes). b)CUDB of the same object (26 boxes).

### 3 Skeletonless virtual porosimetry

The method presented in this paper receives as input the CUDB-encoded pore space of a sample. It consists of two different sorted sets of boxes for 2D samples and three different sorted sets of boxes for 3D samples. Then, an iterative process takes place, computing the invaded region  $R_i$  for each intruded pressure  $P_i$ ,  $P_i > P_{i+1}$ ,  $i = 1 \dots n$ . For each pressure  $P_i$  a representative diameter  $D_i$  is computed by the Washburn equation as follows:

$$D = -(4WC\gamma\cos\varphi)/P \quad (1)$$

where  $D$  is the diameter,  $P$  the intruded pressure,  $\gamma$  is the surface tension,  $\varphi$  is the contact angle and  $WC$  is the Washburn constant.

For each iteration, three main steps are applied over the CUDB-encoded pore space in order to simulate the mercury intrusion. First, all pore regions smaller than the current diameter  $D_i$  are discarded. Second, the throat detection step determines all the transitions, smaller than the current diameter, between adjacent pore regions in order to prevent the incorrect fluid flow along the pore space. Finally, the third step simulates the mercury intrusion, for the current

diameter, by detecting the invaded regions constrained by the marked narrow throats. These steps are explained in more detail in the following subsections.

### 3.1 Discarding small pore regions

The first step consists of the removal of all pore regions smaller than the current diameter  $D_i$ . To achieve this goal, the XY-CUDB, XZ-CUDB and YZ-OUDB encodings of the pore space are scanned and the corresponding discardings are carried out. In each main direction, we discard all those boxes with size smaller than the current diameter. Figure 4 illustrates the discarding process for a 2D example.

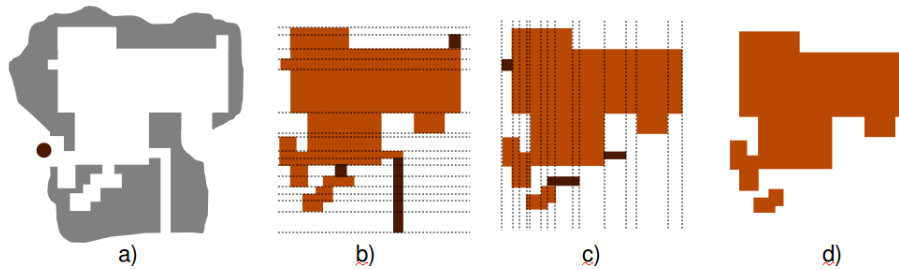


Figure 4: a) Segmented pore space (in white) and intruded diameter  $D_i$ (ball). b) Horizontal discarding: boxes with horizontal section smaller than  $D_i$  are discarded (light grey). c) Vertical discarding: boxes with vertical section smaller than  $D_i$  are discarded (light gray).d) Resulting pore region after discarding.

### 3.2 Detecting narrow throats

Once the discarding step has been accomplished, all regions smaller than the current diameter have been removed. However, there are still transitions, between adjacent pore regions, which are not defined by any section, so they are not detected by the previous discarding process (see figure 5(a)). When these transitions are smaller than the current diameter they are called narrow throats and they can be orthogonal or oblique oriented. Transitions larger than the current diameter are ignored for this intruded pressure. Figure 5 illustrates, in 2D, the narrow throats for a given diameter represented by the ball. As it shows, a fluid with the given diameter can not pass through the throat. The narrow throats prevent full mercury invasion in the whole pore space for the current intruded pressure, so detecting them is mandatory to simulated the correct fluid flow.

In order to detect the narrow throats, an exhaustive scanning of the remaining pore space after the discarding process, is required. Two different tests are applied to detect orthogonal and oblique throats respectively.

#### 3.2.1 Orthogonal throats detection

Two boxes share an orthogonal throat when they share a boundary element, an edge (in 2D) or a face (in 3D). In this case, two different situations can happen: one of the shared faces is a subset of the other one or anyone is a

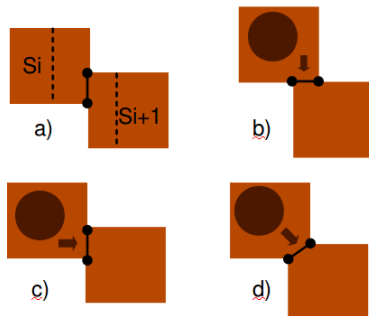


Figure 5: a) There is no section between  $S_i$  and  $S_{i+1}$  but there is a narrow throat between them. Vertical throat (b), horizontal throat (c) and oblique throat (d) for a given diameter (ball).

subset of the other. In the first case, the size of the contained face is the size of the throat. In this case, as both boxes survived the discarding process, both of them are large enough to contain the intruded fluid. Therefore, the throat, as large as the size of the smaller box, will always be larger than the current diameter. More interesting is the second case, when two boxes share a piece of their boundary, illustrated in Figure 5(b) and 5(c). Both adjacent boxes can be large enough to contain the current ball, however the ball could not pass from one box to the other one because the throat between them is smaller than the current diameter. In these cases, an orthogonal narrow throat has been detected. Orthogonal narrow throats are always orthogonal to the coordinate axes, so in order to identify them we must traverse the model in the two or three axes directions for 2D and 3D objects respectively and the following test must be applied:

Let  $S_i$  and  $S_{i+1}$  be the transversal sections of two adjacent boxes, according to the scan direction  $A$ , there exists a narrow throat in  $A$ -direction between them iff:

$$\begin{aligned}
 & (S_i \cap S_{i+1} \neq \emptyset) \wedge \\
 & (S_i \not\subseteq S_{i+1}) \wedge (S_{i+1} \not\subseteq S_i) \wedge \\
 & \text{Size}(S_i \cap S_{i+1}) < D
 \end{aligned} \tag{2}$$

where  $D$  is the diameter corresponding to the current intruded pressure.

When a narrow throat is detected, the model must be subdivided along the throat. To achieve this, a very simple method based on duplicating vertices is carried out. Figure 6 illustrates the subdivision strategy for orthogonal throats.

### 3.2.2 Oblique throats detection

Let  $S_{i-1}$ ,  $S_i$  and  $S_{i+1}$  be the transversal sections of three consecutive boxes, and let  $\bar{S}_i$  be  $S_i$  without its boundary, then an oblique throat between  $S_{i-1}$  and

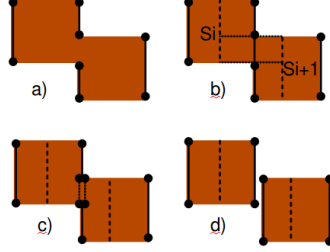


Figure 6: a) Original object, b) orthogonal throat detection, c) object with duplicated vertices along the throat, d) resulting subdivided object

$S_{i+1}$  exists if the following constraints are satisfied:

$$\begin{aligned}
 i) \quad & (\bar{S}_{i-1} \cap \bar{S}_{i+1}) \neq \emptyset \\
 ii) \quad & \bar{S}_{i-1} \not\subseteq \bar{S}_{i+1} \\
 iii) \quad & \bar{S}_{i+1} \not\subseteq \bar{S}_{i-1} \\
 iv) \quad & (\bar{S}_{i-1} \cap \bar{S}_{i+1}) \subset \bar{S}_i
 \end{aligned} \tag{3}$$

Figure 7 (a) illustrates the above conditions for a 2D example. However, the oblique throats are not restricted to three consecutive sections. According to the geometry of the object, an oblique throat can occur between two sections  $S_A$  and  $S_B$  which have more than one intermediate section between them, as shows Figure 7 (b). In these cases, conditions (i), (ii) and (iii) of Eq.(3) must be satisfied for  $S_A$  and  $S_B$  instead of  $S_{i-1}$  and  $S_{i+1}$  respectively, and condition (iv) can be rewritten as:

$$iv) \quad \forall i[A < i < B \Rightarrow (\bar{S}_A \cap \bar{S}_B) \subset \bar{S}_i] \tag{4}$$

It means that  $S_A \cap S_B$  is contained in every intermediate section between  $S_A$  and  $S_B$ .

Once all the oblique throats have been detected, the object is subdivided along each one of them. To achieve this subdivision, a unit-width orthogonal polygon (2D) or polyhedron (3D) is built for each oblique throat. This OPP is obtained by the 2D or 3D scan-conversion of the line segment (2D) or plane quadrilateral (3D) which represent the oblique throat. Therefore, the difference between the OPP-object and the OPP-throat produces the object subdivided along the throat. Figure 8 illustrates the subdivision of the object along an oblique throat.

### 3.3 Mercury intrusion simulation

The two previous steps give a CUBD model of the pore region larger than the current diameter. Moreover, the model has been subdivided along all its orthogonal and oblique narrow throats to prevent incorrect fluid passing through

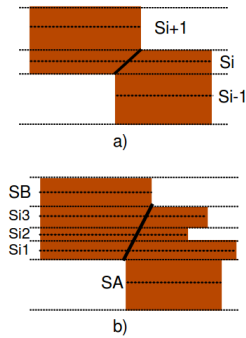


Figure 7: Oblique throats. a) Simplest case of oblique throat. b) Oblique throat with three intermediate sections.

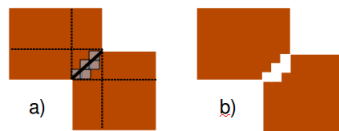


Figure 8: a) Scan conversion of the oblique throat. b) Difference between the object and the scan-converted line.

them. In our approach, reproducing mercury intrusion in the porous sample is a Connected Component Labeling-like method which consists of the two usual passes. The labeling pass labels all the elements and records the equivalences detected on-the-fly. The second pass, called renumbering, solves all the equivalences and renumbers the elements. This algorithm was already used with OUIDB models in [10] to detect disjoint pieces in an OPP. In the present work, a similar algorithm is applied with a different goal: simulating mercury intrusion by classifying each box as invaded or non-invaded using the CUDB model.

Initially, all the mercury entry points of the sample are listed. Then, the CUDB model is traversed and, on-the-fly, each box of the model is labeled with the same value of the already labeled neighbor box or with a new label when no neighbor has been labeled yet. If the current box  $b$  is a neighbor of any entry point, then  $b$  is labeled as invaded. When a box has two or more neighbors labeled with different values, a label equivalence is recorded. It means that two regions considered as disconnected ones are actually connected by the current box. All the equivalences are solved in a second traversal of the model, the renumbering pass. After the labeling and renumbering passes have been achieved, all boxes labeled as non-invaded are discarded and the invaded boxes define the CUDB model of the region filled with the mercury for the current mercury pressure.



## 4 Computing the pore map

Once the iterative process has finished, a set of  $n$  labeled CUDB-encoded regions,  $CUDB_{R_i}$ , have been produced. Each region  $R_i$  corresponds to the subset of the pore space invaded by the mercury for the intruded pressure  $P_i$ , so, according to the Washburn law, the following property is always satisfied:

$$\forall i [P_i < P_{i+1} \Rightarrow D_i > D_{i+1} \Rightarrow R_i \subseteq R_{i+1}] \quad (5)$$

Then, let  $\hat{R}_i$  be the region invaded by the mercury for the intruded pressure  $P_i$  exclusively, i.e. that any other lower pressure does not achieve to invade this region. For every  $P_i$ ,  $\hat{R}_i$  can be obtained, in terms of the CUDB model, as:

$$\begin{aligned} CUDB_{\hat{R}_1} &= CUDB_{R_1} \\ CUDB_{\hat{R}_i} &= CUDB_{R_i} - CUDB_{R_{i-1}} \\ 2 \leq i &\leq n \end{aligned} \quad (6)$$

where  $i < j \Rightarrow P_i < P_j \Rightarrow R_i \supseteq R_j$

Finally, the pore map,  $PM$ , where the entire pore space is shown subdivided in pore regions, each one represented by a set of boxes labeled with the same value, is obtained as:

$$CUDB_{PM} = \bigcup_I CUDB_{\hat{R}_i} \quad (7)$$

where each box belonging to  $PM$  has been labeled with the value related to the minimum pressure that allows the mercury invasion of this box.

## 5 Experimental results

In order to validate the performance of the presented method we have reproduced mercury intrusion with it and with an already validated skeleton-based method reported in [8]. Figure 9 shows a visual comparison over a 2D section (376 x 610) of a porous sample with multiple entry points. The top image shows the resulting pore map obtained with the skeleton-based virtual MIP and the bottom one shows the result with our proposal. The skeleton-based method has up to one-voxel error due to the medialness property of the medial axis for even distances in a discrete space. As our approach requires no skeleton, it avoids this approximation error. This can be observed in the slight color difference between images in Figure 9.

For a more objective test, the pore size distribution has been computed from both approaches (see Figure 10). The absolute volume histogram allows us to assert that the maximum difference between both approaches is less than 6%.

Finally, to illustrate the speedup achieved with the presented skeletonless MIP method, table 1 shows the processing times with both approaches.

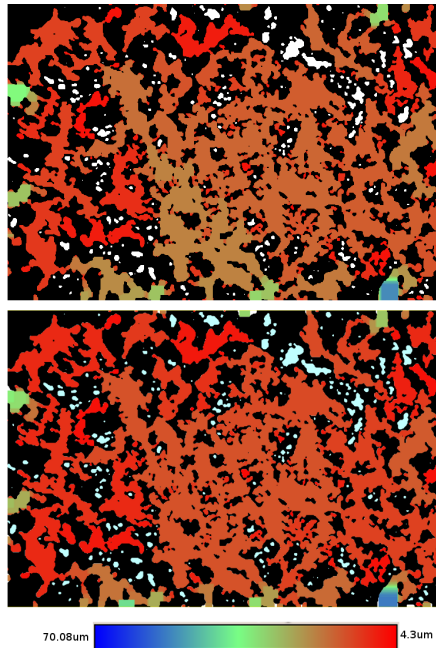


Figure 9: Pore map resulting after MIP simulation with the skeleton-based method (top) and with our skeletonless approach (bottom).

| Object    | Skeleton-based | Skeletonless |
|-----------|----------------|--------------|
| 2D sample | 112 sec.       | 5 sec.       |
| 3D sample | 1800 sec.      | 136 sec.     |

Table 1: Processing time for skeleton-based and skeletonless MIP approaches.

## 6 Conclusion

A new approach to simulate mercury intrusion in a porous medium has been proposed which doesn't need a skeleton computation preprocess. As this preprocess is very time-consuming, our approach achieves a noticeable reduction, up to 30 times less, for the mercury intrusion simulation. Moreover, the CUDB model has been introduced as a compact version of the OUDB partitioning which reduces the total of processed boxes. The results obtained with the skeletonless virtual MIP are very similar to those obtained with the skeleton-based method. As a future work, the pore size distribution, pore map and pore graph obtained with this method will be validated as an efficient approach to compute the permeability and other structural parameters of porous materials.

## 7 Acknowledgements

This work has been partially supported by the national project TIN2008-02903 and by the project 02-1/07 of the Institute for Bioengineering of Catalonia (IBEC). J. Rodríguez was supported by a grant from the University of Carabobo,

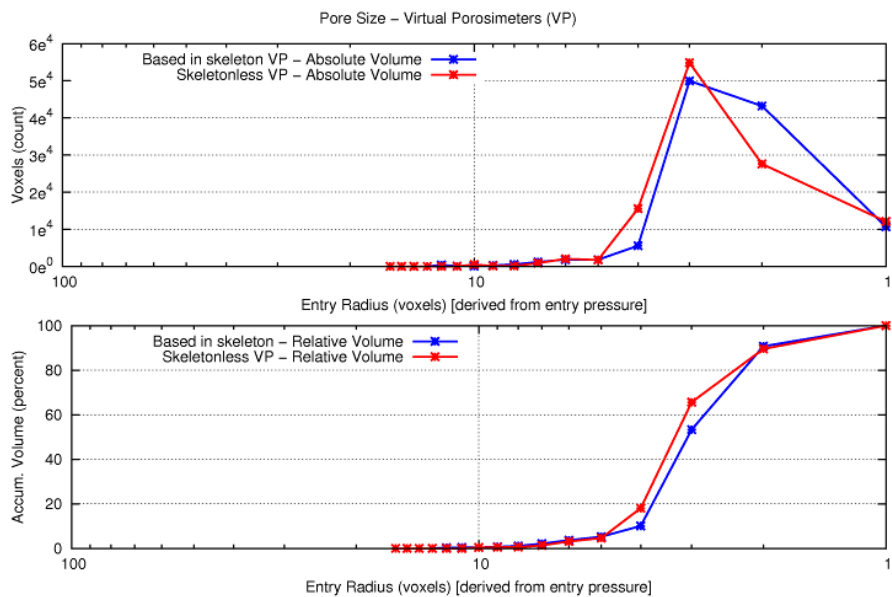


Figure 10: Absolute (top) and relative (bottom) pore size distribution histograms obtained with skeleton-based and skeletonless MIP methods.

Venezuela. I. Cruz was supported by a MAE-AECID grant.

## References

- [1] Aguilera A., Ayala D. *Converting orthogonal polyhedra from EVM to B-Rep and to alternative sum of volumes*, Computing Suppl, vol. 14. 2001. pp. 1-28.
- [2] Rodríguez J., Aguilera A., Ayala D. *EVM: A Complete Solid Model for Surface Rendering*. In Geometric Modeling for Sci. Visualization. Springer-Verlag. 2004. pp. 259-274
- [3] Cnudde, V., Cwirzen, A., Masschaele, B., Jacobs, P.J.S., *Porosity and microstructure characterization of building stones and concretes*, Engineering Geology, vol. 103. 2009. pp. 76-83
- [4] T. Van Cleynenbreugel. *Porous scaffolds for the replacement of large bone defects. a biomechanical design study*. PhD Thesis. Katholieke Universiteit Leuven. 2005
- [5] Delerue, J.F., Lomov, S.V., Parnas, R.S., Verpoest, I., Wevers, M.. *Pore network modeling of permeability for textile reinforcements*. Polymer Composites, vol. 24, 2003. pp. 344 - 357.
- [6] Vergés, E., Ayala, D., Grau, S., Tost, D., *Virtual Porosimeter*. Comp. Aided Design and Apps., vol. 5, 2008. pp. 557-564

- [7] Vergés, E., Ayala, D., Grau, S., Tost, D., *3D reconstruction and quantification of porous structures*. Computer & Graphics, vol. 32. 2008. pp. 438-444. A preliminary version can be found in Proc. CEIG'07, pp. 179-186
- [8] Vergés, E., Ayala, D. *Improved Virtual Porosimeter*. Proc. XXVI CASEIB. 2008. pp. 497-500
- [9] Vogel, H.J. et al. *Comparison of a Lattice-Boltzmann model, a Full-Morphology Model, and a Pore Network Model for Determining Capillary Pressure-Saturation Relationships*. Vadose Zone Journal, vol.4. 2005. pp.380-388.
- [10] Rodríguez J., Ayala D. *Fast neighborhood operations for images and volume data sets*. Computer & Graphics, vol. 27. 2003. pp. 931 -942.