DXSoil, a library for 3D image analysis in soil science

Jean-François Delerue*, Edith Perrier

UR Geodes, Centre IRD Île-de-France, 32, av. H. Varagnat, 93143 Bondy Cedex, France

Received 7 January 2002; received in revised form 25 March 2002; accepted 4 April 2002

Abstract

A comprehensive series of routines has been developed to extract structural and topological information from 3D images of porous media. The main application aims at feeding a pore network approach to simulate unsaturated hydraulic properties from soil core images. Beyond the application example, the successive algorithms presented in the paper allow, from any 3D object image, the extraction of the object skeleton, the measurement of the distribution of local apertures and the partition of the object into a set of interconnected subobjects according to different size classes. The resulting, specific library DXSoil is available under the OpenDX environment more generally dedicated to data exploration and visualization.

© 2002 Elsevier Science Ltd. All rights reserved.

Keywords: OpenDX; Porous media; Skeleton; Segmentation; Permeability

1. Introduction

Imaging has been increasingly used in soil science throughout the past decades to get information about soil composition and organization. Images are obtained at different scales, ranging from that of human vision using photography to smaller scales using different types of microscopic observation devices available in any soil science laboratory. Most experiments have been made on 2D images of soil thin sections. Measurements can be achieved now in 2D using classical image analysis software (Moran et al., 1990) to get global indicators such as soil porosity or solid space volume as well as local information about solid and void shape or size distributions (Ringrose-Voase and Bullock, 1984). Well-known biases occur, due to the 3D nature of the porous medium, especially when topological properties are considered, namely as regards the connectivity of the pore space that affects fluid dynamics (Vogel, 1997). New acquisition techniques have recently been developed to obtain 3D images, either through intensive serial cutting of resin-impregnated soil blocks (Moreau et al., 1999; Cousin et al., 1996) or by means of non-destructive methods based on different types of computed tomography (e.g. Timmerman et al., 1999). As increasing research about analysis and use of 3D imaging is carried out (Moreau et al., 1999; Cousin et al., 1996; Vogel and Roth, 2001), the need for not only geometrical but also functional information is still strong (Lindquist and Venkataraman, 1999), and no end-user software is, as far as we know, yet available to carry out automatic image analysis on such data.

In the context of our research program where we look for deterministic links between soil structural properties and soil hydrodynamics, we developed a library of computer routines that allows the user to analyze 3D soil images (and consequently 2D images). This library, called DXSoil, consists of C-language, specific programs added to the OpenDX software. OpenDX* is a free program dealing with 2D and 3D data exploration and visualization, whatever the field of application, and

*Code available from: http://www.bondy.ird.fr/~delerue/DXSoil/demo

*Corresponding author. Tel.: +33-148-025695; +33-148-473088.
E-mail address: jean-francois.delerue@bondy.ird.fr (J.-F. Delerue).

http://www.opendx.org/
running on many architectures and operating systems such as Windows or Unix. Beyond the useful visualization tools provided by OpenDX, the library DXSoil offers an easy and powerful tool to soil scientists to measure specific, structural and functional soil properties from a 3D soil image.

2. Concepts and objectives

We do not consider here image acquisition nor thresholding. DXSoil routines function using binary images representing the solid and void spatial distribution in a soil sample at a given scale and a given resolution. These routines can be used to characterize the “black” and “white” subspaces equally well, and they can be used on any type of image: from a general point of view, the library enables the partition of a 3D object in a set of interconnected subobjects of different sizes.

If the routines are applied to the void space object in a soil image, it is possible to extract a pore network—i.e. an abstract model of the geometry and topology of this more or less continuous 3D object—linked to fluid flow properties. The void space object in any porous medium can be actually divided into holes of different sizes called pores. The concept of pore is intuitive and well known in geosciences, but no rigorous definition has been actually given in the context of the complex geometry of a real 3D image of a porous medium sample: we will call pore a part of the void space surrounded by the solid space (let us say at least touching the solid-void boundary at two opposite points), with a homogeneous local aperture. We define the local aperture \( A(P) \) as a value associated with each point \( P \) in the pore space object \( O \), equal to the diameter \( d \) of the maximal ball \( B(c, d) \) included in the object and including the point:

\[
A(P) = \max\{d | P \in B(c, d) \cap B(c, d) = O\}.
\]

The pore size is a mean of local apertures calculated at each point in a pore. Pore size, and local aperture at any point are key notions as regards the description of a 3D object, namely as regards the estimation of functional properties such as fluids retention and conductivity through an object open to flow. Operational definitions of pores and apertures from complex 3D data are de facto given by the computer algorithms implemented in the DXSoil routines presented in this paper.

The main application deals with the calculation of hydraulic properties of an unsaturated soil core. From physically based modelling principles, the pore distribution as regards size and location is a major characteristic of soil structure which determines both the water content versus capillary pressure relationship in unsaturated media (where not all pores are filled), and the resulting hydraulic conductivity. A pore network approach can be considered either to represent a simplified porous medium in theoretical studies or be used to model real media by means of statistically equivalent structures (e.g. Vogel and Roth, 2001). Here we work directly on a pore network extracted from a 3D image, using the discrete pore network approach as an alternative to the much more complicated Navier–Stokes equations (e.g. Landau and Lifchitz, 1989) which can simulate fluid flow in a continuous void space with complicated boundary conditions.

Integrated laws are assumed at the pore scale in our model. Here, a local conductivity given by the Poiseuille law (e.g. de Marsily, 1986) depending on the pore size will be associated to each pore. Then the global conductivity of the sample is calculated by electrical conduction analogy, which consists in solving a set of linear equations over the pore network. In the unsaturated case, the pores which are water-filled are determined according to the Laplace law (e.g. de Marsily, 1986), that is as a function of the pore sizes, and the unsaturated hydraulic conductivity curve is calculated for the subnetwork which has water-filled pores. 3D information about the network connectivity is crucial to account for the way water can actually find a path from pore to pore and the DXSoil library gives access to that type of information.

Section 3 presents the base routines, which can be applied successively to extract subobjects of different sizes from any 3D object of a network. First there are: SoilSkeleton to extract the skeleton of the object, SoilFill, a filling algorithm useful to calculate accessibility to any given location, and SoilApertureMap which computes a local aperture map; Further routines SoilSeed and SoilSeedExpand implement growing and merging algorithms to create the partition of the object into subobjects of different sizes and SoilSeparatePore computes the subobject network, i.e. the graphic representation of the subobjects map. Finally SoilDynamic is devoted to the calculation of a global conductivity value of the network when local subobject conductivities are considered. Section 4 deals with our main application about the characterization of a pore space object, and the calculation of its conductivity as regards fluid flows. We present there macro routines which integrate the previous base routines for an easier use of the DXSoil library. In Section 5, we show results on an example of real soil data obtained by X-ray tomography. Some concluding remarks are given in Section 6.

3. Software tools

We now describe the different base routines offered by the DXSoil library. Despite the fact that the name of all routines begin with “Soil”, those routines have been
conceived for the general case where any 3D object has to be decomposed into elementary parts using the criteria based on local aperture. All the routines were inexistant in OpenDX and were added as a part of DXSoil. Most of the routines can be reused independently. We use gray scale images as a convenient way to associate a numerical value to each point (voxel) and to achieve numerical calculations by means of operations on images.

3.1. SoilSkeleton

This routine (Table 1) computes the skeleton of a binary image. The skeleton of an object, also called median line, is a set of points made of the centers of all maximum size balls included in the object. It is a line (Fig. 1B)—or surface in 3D—which divides the object in its center (Schmitt and Mattioli, 1994). Our implementation produces a skeleton where the gray value of each point on the skeleton represents the rounded value of the Euclidean distance from this point to the closest border of the object. This implementation of a skeleton assures that the skeleton is effectively centered on the object and gives the Euclidean distance to the border; however, we do not guarantee the connectivity in the skeleton. In our application, as the skeleton is not directly used to analyze structure, this point has no consequences on the result. The program first computes a Voronoi diagram using a segment-based discrete algorithm (Yu et al., 1998) and then prunes this diagram to retain only the points which belong to the skeleton (Delerue et al., 1999b). The skeleton can be computed either on the foreground object, either on the background or both on foreground and background.

3.2. SoilFill

This routine (Table 2) is the implementation of a filling algorithm and works on any gray scale image. An

<table>
<thead>
<tr>
<th>Name</th>
<th>Default value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orig_im</td>
<td>Input</td>
<td>Binary image</td>
</tr>
<tr>
<td>Distance</td>
<td>Input</td>
<td>2.5</td>
</tr>
<tr>
<td>Object</td>
<td>Input</td>
<td>0</td>
</tr>
<tr>
<td>Skel_im</td>
<td>Output</td>
<td>Gray scale image</td>
</tr>
</tbody>
</table>

Table 1

SoilSkeleton: orig_im is any binary image. Distance is a parameter used to control the pruning sequence. Default value is obtained empirically: it works well but can be tuned for better results. We indicate using object on which part of image we want to compute skeleton: A value of 0 indicates that skeleton is to be calculated for background object, 1 for foreground and 2 for both background and foreground. Each point of skeleton in skel_im is assigned distance to closed border.

Fig. 1. (A) Original image. (B) Skeleton computed by SoilSkeleton. (C) Local aperture map computed by SoilApertureMap. (D) Seed map computed by SoilSeed. (E) Subobject map computed by SoilSeedExpand with method 0. (F) Subobject map computed by SoilSeedExpand with method 1. (G) Subobject network computed by SoilSeparateObjects. Balls represent pores, and their diameters are proportional to pore aperture. (H) Local fluxes computed by SoilDynamic. Length of arrow is proportional to local flux intensity.

area of the input gray scale image—which may reduce to a single point—is first selected by the user to choose the origin of the filling process. The program spreads this
area to all neighboring points of the object in a recursive manner. The spreading process can be restricted to points whose values are included in a given class. The filling can so be done over the foreground object (value 1), or the background one (value 0) or over any subpart defined by a class of values. The result of the routine is a gray scale image where each point of the filled area has its original value replaced by a new value, selected by the user.

3.3. SoilApertureMap

This routine (Table 3) computes the local aperture map (Fig. 1). This routine is a sort of inverse function of the skeletonization, which takes the gray scale skeleton image computed in 3.1 as input. For each point P of the skeleton, it draws a ball of center P and of diameter equal to the value at the point P, that is, from the construction of the skeleton image, equal to the distance d from the point P to the border of the original object. Then each point in the ball is assigned this value d. When two balls overlap, one keeps the value of the ball of higher diameter as described in Delerue et al. (1999a). The result is an aperture map, i.e. an output image where each point of the object is associated with a local aperture value (Eq. (1)) defined by the algorithm and matching our intuitive definition of a pore.

Table 2
SoilFill: input_im is any gray scale image. Pt is a vector which represents origin of filling process. If all coordinates in Pt are positive, then Pt represents a point in image. However, if one coordinate in Pt is equal to −1, this means any coordinate: if Pt is equal to [2, 3, −1] (resp. [−1, 3, −1]) then all points from line x = 2, y = 3 (resp. plane y = 3) will be selected. Search of neighbors during spreading is restricted to points whose values belong to class of values [min, max]. In output_im every point in filled area is assigned value new selected by user.

<table>
<thead>
<tr>
<th>Name</th>
<th>Default value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input_im</td>
<td>Input</td>
<td>Gray scale image</td>
</tr>
<tr>
<td>Pt</td>
<td>Input</td>
<td>Vector</td>
</tr>
<tr>
<td>Min</td>
<td>Input</td>
<td>Integer</td>
</tr>
<tr>
<td>Max</td>
<td>Input</td>
<td>Integer</td>
</tr>
<tr>
<td>New</td>
<td>Input</td>
<td>Integer</td>
</tr>
<tr>
<td>Output_im</td>
<td>Output</td>
<td>Gray scale image</td>
</tr>
</tbody>
</table>

Table 3
SoilApertureMap: Routine computes aperture map apert_im from skeleton skel_im, that is from output of SoilSkeleton.

<table>
<thead>
<tr>
<th>Name</th>
<th>Default value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skel_image</td>
<td>Input</td>
<td>Gray scale image</td>
</tr>
<tr>
<td>Apert_im</td>
<td>Output</td>
<td>Gray scale image</td>
</tr>
</tbody>
</table>

3.4. Object extraction oriented routines

In this section, we present a set of three routines which aims at dividing the object into elementary subobjects, and then at producing a graph abstraction of those subobjects based on the recording of all neighboring links between them.

3.4.1. SoilSeed

This routine (Table 4) places seeds for the reconstruction of the elementary subobjects (Fig. 1D). It takes as input the skeleton image of the object produced by SoilSkeleton. Then it places non-overlapping maximum balls centered on the skeleton as shown in Delerue et al. (1999a). Only maximal balls of the same diameter may overlap each other. This routine is used to initialize the SoilSeedExpand routine in 3.4.2.

3.4.2. SoilSeedExpand

This routine (Table 5) takes the seeds produced by SoilSeed as input. Its aim is to extend each seed synchronously until all the original 3D object is occupied. Seeds spread in every direction until they touch another growing area or the border between the object and its background. The local speed of the extension process can either be fixed and equal on each point, or variable and proportional to a local value. In the latter case, the local speed for each point P of the front propagation is inversely proportional to the difference between the local aperture given by SoilApertureMap at point P and the value of the seed. Finally, the whole 3D object is partitioned into elementary subobjects defined by the resulting spread areas. The aperture or size of each subobject is calculated as a mean value of the local aperture at each point in the subobject; it is very close to the initial size value associated to each seed in the growing and merging process. The output of the routine is a map (Delerue et al., 1999a) (Fig. 1E and F) where each point of the object is assigned the value of the size of the subobject it belongs to.

3.4.3. SoilSeparateObjects

This routine (Table 6) takes the object map produced by SoilSeedExpand as input. It produces a representation of the object map as a network where each node

<table>
<thead>
<tr>
<th>Name</th>
<th>Default value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seed_im</td>
<td>Output</td>
<td>Gray scale image</td>
</tr>
</tbody>
</table>
Table 5
SoilSeedExpand: From image seed_im, output of SoilSeedExpand, growing process is defined using method to give speed growth: If method is equal to 0, then speed is fixed and equal to 1. If method is equal to 1, then local speed for each point P of front propagation is proportional in inverse ratio to difference between local aperture given by apert_im at point P and value of the seed. Also needed as input: original binary image orig_im and object, previously selected within that image using object. In routine output subobj_im, each voxel is assigned value of subobject to which it belongs.

<table>
<thead>
<tr>
<th>Name</th>
<th>Default value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seed_im</td>
<td>Input</td>
<td>Gray scale image</td>
</tr>
<tr>
<td>Apert_im</td>
<td>Input</td>
<td>Gray scale image</td>
</tr>
<tr>
<td>Orig_im</td>
<td>Input</td>
<td>Binary image</td>
</tr>
<tr>
<td>Object</td>
<td>Input 0</td>
<td>Integer</td>
</tr>
<tr>
<td>Method</td>
<td>Input 0</td>
<td>Integer</td>
</tr>
<tr>
<td>Subobj_im</td>
<td>Output</td>
<td>Gray scale image</td>
</tr>
</tbody>
</table>

Table 6
SoilSeparateObjects: subobj_im is output of SoilSeed; Output file geom.net contains following fields: size, positions and connections using DX graphical format; Main output is graph, file that contains same information using specific matrix-based representation, plus direction for each connection, calculated from macroscopic orientation given by inlet and outlet vectors from and to.

<table>
<thead>
<tr>
<th>Name</th>
<th>Default value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subobj_im</td>
<td>Input</td>
<td>Gray scale image</td>
</tr>
<tr>
<td>From</td>
<td>Input</td>
<td>Vector</td>
</tr>
<tr>
<td>To</td>
<td>Input</td>
<td>Vector</td>
</tr>
<tr>
<td>Geom_net</td>
<td>Output</td>
<td>Geometric field</td>
</tr>
<tr>
<td>Graph</td>
<td>Output</td>
<td>Internal representation</td>
</tr>
</tbody>
</table>

represents a subobject and each link a connection between the subobjects. The first output consists of a geometric representation (Fig. 1G) using a format which can be rendered by OpenDX and which contains the position of the subobjects centers, their size, and the position of the segments connecting them. The second output is a graph that uses an internal matrix-based format to store the same information—i.e. the network nodes and links—plus the orientation of each link when an inlet and an outlet have been selected by the user on the original object.

3.5. SoilDynamic

When the object can be considered as a network of conducting subobjects, SoilDynamic (Table 7) is a specific routine dedicated to the derivation of the equivalent conductivity of the whole object from the local data. Our specific application concerning a network of fluid conducting pores will be presented in the next section. SoilDynamic takes as an input file the internal graph representation of SoilSeparateObjects. For a given graph, local conductivities can be given to each node by selecting a function depending on the application field, the mean conductivity of the link between any two neighboring nodes is given by the user who selects a type of mean. Then one proceeds by analogy with a Kirchoff network of electrical resistances. One considers an arbitrary, global potential gradient, imposed between the inlet and outlet of the network. This determines the direction of each link in the input graph, and no-flux conditions are assumed in any other direction. The local potential values at each node are the unknowns in a system of linear equations describing local fluxes entering and leaving each node. Solving the system gives access to the actual value of the local potentials and local fluxes. Then the global, equivalent conductivity is calculated as the proportionality ratio between the total flux through the outlet and the global potential gradient.

The output of the routine consists of a scalar value, equal to the equivalent conductivity, and in a geometric representation it contains information about the network nodes and links, plus local potentials at each node and local fluxes through each link. Local potentials or fluxes can be displayed as in Fig. 1H.

4. Application to soil science

The above routines may be useful for the analysis of any 3D object. We now describe small programs to
demonstrate the use of the obtained routines when applied to the pore space of a porous medium and to fluid flow, namely in a soil and water context. The results are estimations of the pore size distribution, of the water retention curve, and of the hydraulic conductivity curve calculated from a 3D image of a pore space. We also display a representation of the pore network.

4.1. Pore size distribution

Our first application consists in computing the pore size distribution (PSD). It seems obvious that the PSD is the distribution of local aperture in the pore space of the soil image. The PSD is straightforwardly obtained by computing the histogram of the Aperture Map. (It could also be computed from the sizes of the pore subobjects after having applied a longer series of routines leading to the pore network extraction). As shown in Fig. 2, the soil image is first loaded using OpenDX’s routine Import. Then SoilSkeleton computes the skeleton of the pore space object in the image. The skeleton image is then given to SoilApertureMap. Then by applying the routine Histogram from OpenDX, we obtain the distribution of apertures. Plot and Image are two OpenDX routines used to render the result.

4.2. Retention curve

The retention curve describes the different values of pressure head associated with a given fluid content in a porous medium. Here, we simulate the intrusion of a non-wetting fluid (e.g. air entering a water-filled soil) into the pore space at different steps of pressure \( h \), as shown in Delerue et al. (1999b): the fluid reaches points \( P \) with smaller and smaller local apertures \( A(P) \) according to the Laplace law which gives the inverse proportionality \( h = \frac{z_1}{A(P)} \), where \( z \) is a known constant depending on the fluid. A first part of the program (Fig. 3) combines Import, SoilSkeleton and SoilApertureMap to compute the local aperture map. Then ForEachN gives successive aperture values and SoilFill, starting from the location where the fluid begins to invade the sample, fills all connected areas with a size greater than the one given by ForEachN. The result of SoilFill is the number of voxels which have been filled, i.e. the amount of fluid in the soil sample. By collecting for each aperture the amount of voxel filled in the pore space, we can draw the retention curve. As an example, the retention curve computed from image 6 is presented in Fig. 7A.

4.3. Conductivity curve

The conductivity curve describes the hydraulic conductivity of the porous medium for different pressure steps. At any water content, the hydraulic conductivity value is the ratio between any imposed pressure gradient and the resulting water flux. We use a classical approach in pore network modelling field—for detailed information we refer to Perrier et al. (1995)—and we work on the pore network that can be extracted from the data. This is done by the combination of Import, SoilSkeleton, SoilApertureMap, SoilSeed and SoilSeedExpand (Fig. 4). As for the retention curve (Section 4.2), a process of fluid intrusion is simulated using ForEachN and SoilFill. Then, Compute removes all pores from the pore map which have not been filled by SoilFill. The simplified pore map is then given to SoilSeparateObjects which produces a graph representation of the pore network of water-filled pores for the given pressure.
Finally, *SoilDynamic* computes the equivalent hydraulic conductivity. The default function to compute the local flows $q$ and local hydraulic conductivities $k_i$ is the Poiseuille law for a cylindric tube $i$:

$$q_i = \frac{\pi r^4 h_{in} - h_{out}}{8\mu L},$$

where $h_{in} - h_{out}$ is the local potential gradient, $\mu$ is a constant, $r$ is the pore size (width), $L$ is the pore length, and $k_i = \frac{\pi r^4}{8\mu}$; the mean conductivity value $k$ between two neighboring pores is the harmonic mean derived for serial cylinders $2/k = (1/k_1) + (1/k_2)$. As for the retention curve, by collecting for each pressure the equivalent hydraulic conductivity, we can draw the conductivity curve of the pore space. As an example, the hydraulic conductivity curve computed using the image presented in Fig. 6 is shown in Fig. 7B.

It is to be noted that the retention curve can be obtained also by this program (Fig. 4) as the amount of the pore space filled by the fluid as given by the second output of *SoilFill*.
Fig. 6. (A) Original 3D image, where pore space is colored red. (B) Skeleton of pore space computed by SoilSkeleton. (C) Seed map computed by SoilSeed. (D) Pore map computed by SoilSeedExpand using method 1. (E) Graph representation of pore network, with ball sizes proportional to pore sizes. (F) Representation of local flows generated by top-down potential gradient, where ball sizes are proportional to flow intensities and arrows show flow direction.
4.4. Network builder

The aim of this program (Fig. 5) is to display a graphic representation of both structural and dynamical properties of the pore space. We first need to obtain the pore map, which is done by combining Import, SoilSkeleton, SoilSeed, SoilApertureMap and SoilSeedExpand. Secondly, we need to compute the graphic representation of the pore network and to compute the dynamical properties, which is done, respectively, by SoilSeparateObjects and SoilDynamic. Let us note that several OpenDx routines—Mark, Showconnections, AutoGlyph, Compute, Collect, Image—are used to display the geometrical data produced by the DXSoil routines. A pore is represented by a ball, centered on the pore center, the diameter of which is proportional to the size of the pore. This is done by applying Mark to the first output of SoilSeparateObjects to select size as main data. Then AutoGlyph draws the balls. To draw the connections between the different pores, we use Showconnections. In the same way, to display the local fluxes of fluid, two Mark are used to select flux and vector as main data from the first output of SoilDynamic. Compute multiplies each unit vector with each flux intensity to make a flux vector, and all the flux vectors are drawn using AutoGlyph. Finally, Collect brings together the two glyph representations made by AutoGlyph and Image displays the final graphics.

5. Example: analysis of a tomographic soil image

In this part, we show the use of our library on a 3D soil image of macro-porous sandy loam soil, obtained by X-ray tomography (Timmerman et al., 1999). The original image was a gray scale image, where the value of each voxel represented the local X-ray attenuation through the material, which is proportional to the density of solid in the soil sample. The sample had been first binarized to separate pore space from solid space and we worked on the resulting image shown in Fig. 6A. Its size is 355 by 355 by 460 voxels, where each voxel represents a cube of 0.2 mm by 0.2 mm by 0.2 mm. The computed skeleton of that image is shown in Fig. 6B. The seed map is shown in Fig. 6C. Then the pore map is shown in Fig. 6D. In the skeleton, seed map and pore map representations, each color represents a different class of aperture. Then Fig. 6E shows the graph representation and Fig. 6F shows local flows for each pore. In Delerue et al. (1999a), the equivalent conductivity computed by SoilDynamic has been successfully compared with the hydraulic conductivity measured experimentally in the saturated soil. This program was run both on an UltraSparc 10 with 256 Mbyte of memory in 5 min and on a PentiumIII 450 Mhz with 256 Mbyte of memory under Linux in 4 min.

6. Conclusion

When 3D soil images are available, and increasingly such data are expected to be available, the DXSoil routines provide the means to calculate not only the pore or solid size distributions, fluid (water or mercury) invasion processes, but also the conductivity of the soil core at different water contents. As regards the limited definition of the available images, as well as regards the
lower bounds (e.g. the small pore sizes) and the upper bound (e.g. the high variability of the structural organization from one sample to another one), this scale issue is still an open research question, beyond the scope of this methodological paper. The presented method to determine pore objects and pore sizes within a 3D structure appears to be accurate, from an intuitive and visual point of view, and moreover it provides a definition of a concept widely used in soil science. Since the calculation of the conductivity value is the result of the whole series of DXSoil routines, the comparison between the predicted and measured saturated hydraulic conductivity on an example is promising. Whatever the results on further experiments, the value of the conductivity calculated using the DXSoil library is an accurate indicator of the sample connectivity, strongly linked to fluid behavior through a physically based network approach. Our goal is to provide new image analysis tools to quantify better the information contained in 3D images, first of all in soil science. The programs could also be used in other fields such as medical imaging where vessels and pulmonary vascular tree can be seen as network of “tubes” where a fluid (blood or oxygen) flows in a way linked to the shape of the network (Williams and Wolff, 1997).

Acknowledgements

This work was partly supported by a grant from the French Programme National Sol et Erosion program. We warmly thank Anthony Timmerman who let us use his sets of tomographic data.

References


