Modelling Three-Dimensional Geoscientific Datasets with the Discrete Voronoi Diagram

Tom van der Putte and Hugo Ledoux

Abstract Geoscientific datasets are often formed by scattered samples in 3D space having highly anisotropic distribution. To model the continuity of the phenomena they represent (e.g. temperature of a body of water, or percentage of a chemical in the air) raster structures are in most cases used. To overcome the shortcomings of rasters the Voronoi diagram (VD) has been proposed as an alternative. However, while in theory the VD is a sound solution, its use in practice is hindered by the fact that it is complex to construct and to manipulate (removal of samples, interpolation, etc.), and spatial tools have to be built. We propose in this paper a 'middle' solution: the 3D *discrete Voronoi diagram* (DVD). We investigate the properties of the 3D DVD, we propose algorithms to construct and manipulate it, and we demonstrate its use in practice with a prototype that we have built. Our prototype uses existing tools for visualisation and further analysis of DVDs.

1 Introduction

The geoscientific disciplines such as oceanography, meteorology and geophysics are different from the other applications of 3D GIS because, instead of modelling man-made objects (e.g. houses, bridges, tunnels), we have to model the spatial distribution of continuous geographical phenomena in 3D space. Examples of these phenomena are the salinity of a body of water, the humidity of the air or the percentage of a certain chemical in the soil. The representation and analysis of such phenomena is complicated by the fact that the collection of samples is problematic: it is often impossible to measure everywhere these phenomena and as a results the

T. van der Putte (✉)
TNO, Geological Survey of the Netherlands Princetonlaan 6, Utrecht 3584 CB, The Netherlands
e-mail: tom.vanderputte@tno.nl

H. Ledoux
GIS Technology Group, Delft University of Technology, Delft 2628 BX, The Netherlands
e-mail: h.ledoux@tudelft.nl

datasets are formed by sparse and anisotropically distributed samples in 3D space. Three-dimensional continuous geo-information is usually modelled with raster structures, either directly with 3D grids (voxels) or with hierarchical grids such as octrees (Bak and Mill 1989; Jones 1989). This is due to the fact that raster structures are simple structures naturally stored and manipulated in a computer (with arrays of numbers) and hence a certain number of modelling tools are available. However, raster structures also have several disadvantages both in theory and in practice (Fisher 1997; Kemp 1993): (1) space is arbitrarily divided without taking into account the original samples; (2) the size of a 3D raster can become huge is a fine resolution is wanted; (3) rasters scale badly and are not rotationally invariant; (4) the original samples are 'lost'. To circumvent these problems, Ledoux (2006) and Ledoux and Gold (2008) have proposed the Voronoi diagram (VD) as an alternative and showed that it is useful not only for the representation of 3D phenomena, but also for their analysis. The main advantages are: (1) the tessellation of the 3D space obtained with the VD gives a clear and consistent definition of neighbourhood of the samples and adapts to the distribution of these; (2) the continuity of the 3D phenomena can be reconstructed with Voronoi-based interpolation methods; (3) the structure is locally modifiable, which permits us to interactively explore a dataset and manipulate it; (4) it enables several visualisation operations, as well as several spatial analysis operations.

While the 3D VD is conceptually superior to raster structures, it has drawbacks in practice. It is indeed rather difficult to construct it in a robust and efficient way (Field 1986; Sugihara and Inagaki 1995), and manipulation algorithms (movement and removal of points) are problematic to implement and as far as we know only exist for the VD of points in Euclidean space (Devillers and Teillaud 2003; Ledoux et al. 2005; Hoffmann 1989; Russel 2007). Another obstacle to its use by practitioners is that specialised tools have to be built.

In this paper we investigate the use of the 3D *discrete* Voronoi diagram (DVD) for the modelling of geoscientific datasets. The DVD is one natural variant of the 'normal' VD; several other variations are possible, see ? for an exhaustive list. It can be seen as a 'middle' solution between the VD and raster structures. Both models have pros and cons, and it is interesting to study which ones will be retained with the DVD. As Fig. 1a shows, given a set of points in space the VD divides the space into cells (called Voronoi cells) in such a way that every location within each cell is closest to the points that lies within that cell compared to all other points; the boundaries of the cells represent the locations that are equidistant from two or more points. The DVD of the same pointset is shown in Fig. 1b. With it the cells are formed by groups of pixels having the same ID (here a colour) and we can notice that they have a similar shape; but when we zoom-in on an edge between two cells, the tessellation becomes visible. The DVD is also possible in 3D: the cells are convex polyhedra formed by groups of voxels. We describe further in Sect. 2 the DVD and discuss related work. We also present in Sect. 3 new algorithms to construct and manipulate (e.g. removal of points) 3D DVDs. Observe that we are particularly interested in dynamic solutions where users can modify interactively DVDs (as in Anselin 1999) because we believe it to greatly help the exploration and
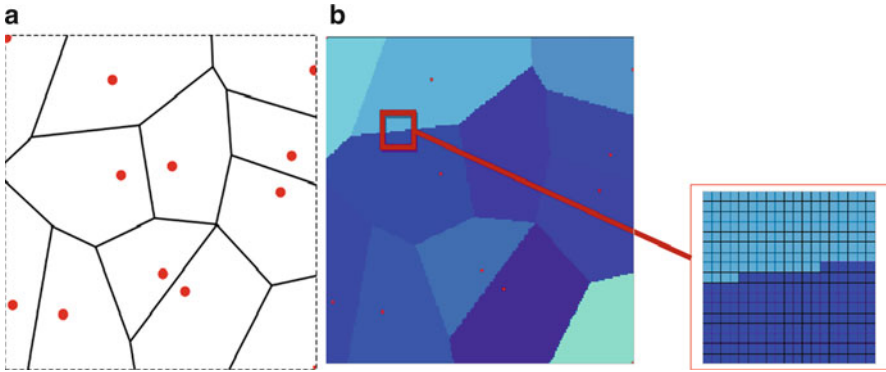
Fig. 1 (**a**) The VD for a set of points in 2D. (**b**) The DVD in 2D for the same set of points; the zooming in on a small section of the DVD shows the discrete nature of the structure

understanding of complex 3D datasets. We have implemented these algorithms and used them for the modelling of geoscientific datasets. We describe briefly in Sect. 4 the architecture we used, which is a mix of our own code and the use of the open-source tools for the handling, analysis and visualisation of the 3D DVDs.

2 Work Related to the 3D DVD

Given a set of points in 3D space (also called 'seeds'), the 3D VD divides the space into convex polyhedra that represent the 'closest' space around each seed. As Fig. 2 illustrates, for the 3D DVD of the seeds the space is tessellated into regular cells (cubes) that we call voxels. The Voronoi cells in discrete space are represented by a group voxels which share the same value (the ID of the seed generating the Voronoi cell). It should be noticed that while the Voronoi cells of the seeds on the boundary of the convex hull are in theory unbounded, for the DVD they have to be bounded arbitrarily (as the Fig. 2 shows).

Constructing the DVD for a set of seeds therefore boils down to finding the closest seed of every voxel and assigning that ID. When a voxel is situated at exactly the same distance from two or more seeds we have a 'tie' or a 'conflict situation'. Notice that in this paper we use Euclidean distances and not a different metric (e.g. the Manhanttan distance). The distance between two voxels is thus a Euclidean distance between the two centres of the voxels. However, using another metric would be relatively simple and could help building generalised Voronoi diagrams (Okabe et al. 1994); we discuss in Sect. 5 the possibilities.

We review in this section the algorithms and methods that have been proposed to construct a DVD. It should first be noticed that of all methods that we discuss only one was specifically developed for the 3D DVD, the rest were aimed towards the 2D VD; the generalisation of these is however technically possible.
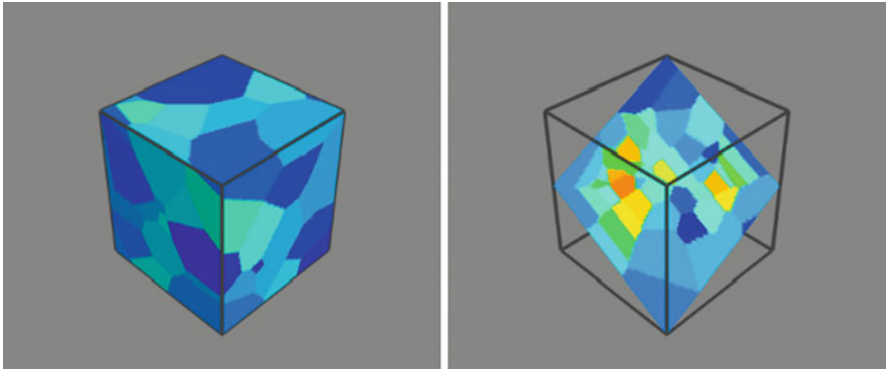
Fig. 2 *Left*: an example of a 3D DVD, only the boundary of the space is shown. *Right*: a slice from the DVD, which helps understanding the spatial distribution of the points in space
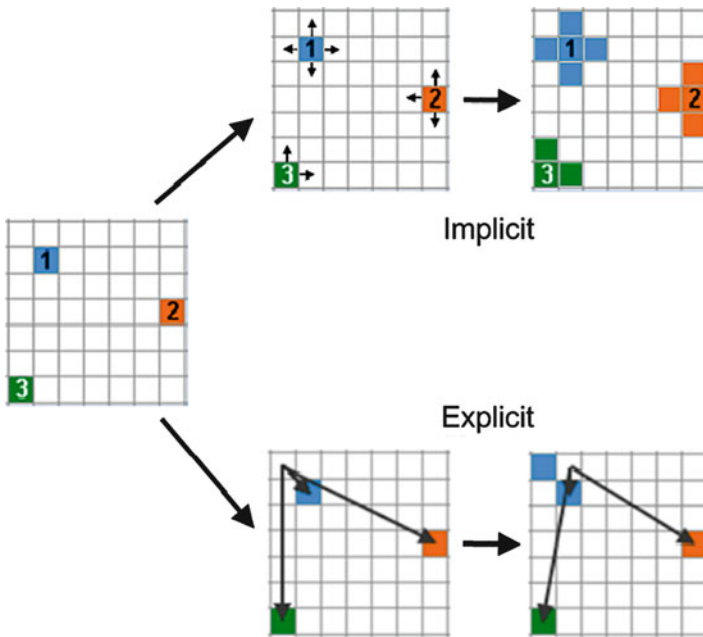

Fig. 3 The implicit and explicit methods to construct a DVD

To construct a 3D DVD, we distinguish between two different types of methods (as shown in Fig. 3):

- *Explicit methods* calculate the value to assign to each voxel by finding the closest seed in the input dataset.

- *Implicit methods* assign a voxel to a seed based on the value of the voxel next to them; several passes over a grid is needed. They are theoretically efficient since distance calculations are not required for every voxel. However, a control mechanism is imperative to ensure that voxels are assigned correctly.

2.1 Explicit Methods

A brute-force implementation of the explicit method with n seeds and m voxels would require for all m voxels to calculate the distance to each seed and assigning the ID of the closest, which is $O(n\,m)$ – with an arguably small tessellation of $100 \times 100 \times 100$ voxels with only 500 seeds, 500 millions distances have to be calculated. A major disadvantage of this method is when trying to add, move or remove seeds: it requires the entire VD to be rebuilt.

The brute-force algorithm can however be sped up with the use of auxiliary data structures, such as the *kd*-tree. Park et al. (2006) use it to create both the 2D and the 3D DVD. A *kd*-tree recursively partitions the space into a binary tree that can be used to efficiently solve the nearest neighbour problem. However, the efficiency of nearest neighbour queries requires a balanced tree and after points have been inserted or removed the tree becomes unbalanced and has to be re-balanced. But the tree structure and the DVD are not dynamically linked; in other words, when updating the tree, the DVD is not updated automatically. The main advantage of explicit methods is that they are not error-prone.

Several people have devised ways to create a DVD with the aid of the GPU (graphical processing unit) instead of the CPU, see among others Hoff et al. (1999) and Rong and Tan (2006, 2007). While these methods are very fast for the 2D case, they do not generalise directly to the 3D case, except Rong and Tan (2007) who propose a slice-by-slice approach. Another issue with GPU-based methods is that an *approximation* of the DVD is sought for visualisation purposes (i.e. they do not guarantee that a correct DVD is created in case of ties), and that could be problematic if this DVD is used for further analysis.

2.2 Implicit Methods

Schueller (2007) proposes a 2D method based on the principle of expanding circles around each seed to assign IDs (principle of dilation). It is theoretically more efficient than the brute-force explicit algorithm because only the pixels at the boundaries between two or more Voronoi cells have to be checked to determine the closest seed. In other words, as long as circles of two seeds do not touch, all the pixels within that circle are correctly assigned to the corresponding seed. To assess which pixels are neighbours of a claimed pixel P, he proposes different neighbour-hoods: the Moore and the Neumann neighbourhood (also called in 2D the 8- and the 4-neighbourhood), as shown in Fig. 4a. These are approximations of a circle and are
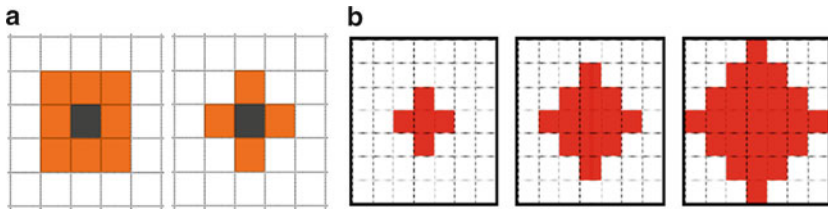
Fig. 4 (**a**) *Left*: the Moore neighbour. *Right*: the Neumann neighbourhood. (**b**) Three rounds of dilation for the Neumann neighbourhood

dilated around each seed; Fig. 4b shows the first three rounds of dilating a Neumann neighbourhood.

Li et al. (1999) use the dilation principle to create the DVD of different shapes (lines and polygons) but to make their implementation faster they use the Manhattann instead of the Euclidean distance.

Zhao et al. (2002) also use the dilation concept but in combination with a quadtree. They reason that when dilating seeds into Voronoi cells, it is only necessary to calculate distances between a pixel and two or more seeds for pixels near the boundaries of Voronoi cells. Pixels that are closer to the seeds might be assigned to a seed more effectively. This is done by creating a quadtree of the raster, so that whole areas of pixels that belong to a certain seed can be assigned that value in one operation.

3 Our Algorithms to Construct and Manipulate a 3D DVD

We describe in this section the algorithms that we have designed to construct a 3D DVD, to add/remove a single point to/from a 3D DVD, and to convert one DVD into a continuous raster with natural neighbour interpolation. The algorithms we propose are based on the dilation method of Schueller (2007); we generalised it to 3D and we made several modifications to the dilation component to make it more efficient. First observe that in 3D the Moore neighbourhood consists of a 3×3 voxels cube, and that the Neumann neighbourhood consists of a 6-voxel object, where all six faces of the central voxel are adjacent to one of the six voxels. These approximate a sphere in raster space, but, because they are not perfect spheres, when dilating them errors can occur. The challenge is ensuring that no errors occur while minimising the number of distances calculated. Figure 5 illustrates the problem with a 10×7 pixels in which two seeds have been inserted. These seeds are iteratively dilated with a Moore neighbourhood structuring element, resulting in expanding squares. The green voxel in the figure shows a situation that could introduce errors. In the last round, this has been claimed by the blue seed, but when looking purely at the Euclidean distance, the distance to the blue seed is larger than that of the orange. The way to overcome this is to allow the algorithm to reassign voxels that have already been claimed.
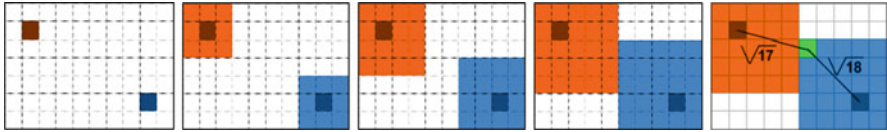
Fig. 5 Dilation of two seeds and the incorrect claiming of a pixel/voxel

Notice that because of space constraints we cannot give all the details of the algorithms but these are available – including pseudo-code – in the Master's thesis of the first author (van der Putte 2009). A proof of correctness for the construction, and its termination, is also available.

3.1 Construction of the 3D DVD

The following construction algorithm assumes that all seeds are known in advance and construct the DVD in one operation. The general idea of the algorithm is as follows. First all the voxels containing an input seed are assigned an ID, and the other ones are assigned NoData. Then a Neumann neighbour is used (we have used and tried others but the Neumann gave the best result, as it did for Schueller 2007) and placed over every voxel that has been given a ID value already, which actually 'stamps' the structuring element onto the original 3D grid. The algorithm continues until every voxel has been assigned an ID (the termination condition is that no voxel has been modified during one pass).

At each step of the algorithm the voxels with IDs are dilated, which means that they try to claim their neighbours. If the neighbour has the same ID value as the current voxel v, it is correctly assigned already; if a neighbour has NoData it is automatically assigned the ID of v; if an ID has been previously set, but not the same as that of v, then we have a conflict. The Euclidean distance to both claiming seeds are calculated and the minimum wins. If the two distances are exactly the same an arbitrary choice must be made, which can be based for instance on the lexicographic order of the IDs or first-come-first-served. Notice that the rule can influence significantly the resulting DVD, as Fig. 6 shows.

Observe that the only voxels that need to be dilated are in fact the voxels on the boundary of each dilating area; those 'inside' a dilating Voronoi cells have already been processed. This criterion can be narrowed down even further by stating that only the boundary voxels of Voronoi cells that are still dilating need to be processed at each step of the algorithm. This removes the need for dilating those voxels that are at boundaries that are stationary, either at the edges of the image or at the stable boundaries between areas. In effect, this means that only the voxels that – in the current pass – have been assigned a different ID value than they originally carried need to be considered for dilation. The improvement to the algorithm is done by keeping a list of these voxels, and since we use arrays we have direct access to them.
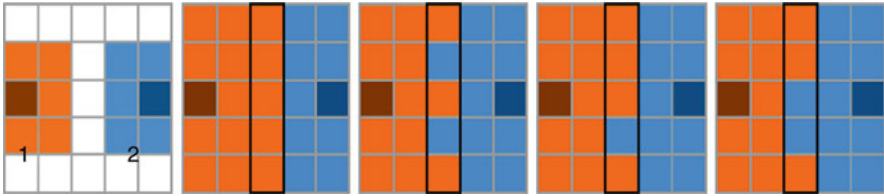
Fig. 6 Random assignment of two equidistant seeds. The pixels/voxels of the middle column could be assigned to either seeds, which would yield different DVDs
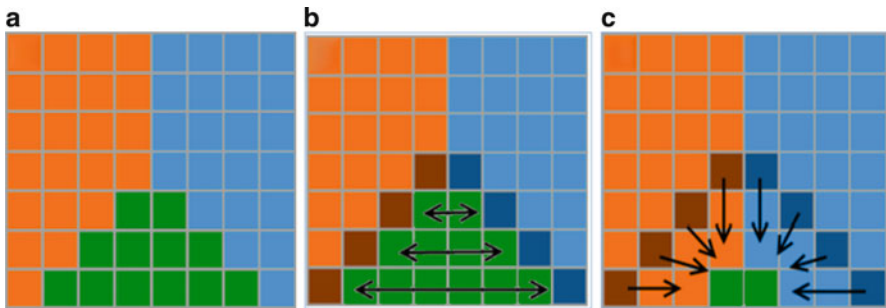

Fig. 7 The process of removing a seed and its Voronoi cell. (**a**) Three Voronoi cells. (**b**) Identifying the *green voxels* and boundary voxels. (**c**) The *darker coloured boundary voxels* are then redilated until all *green voxels* are overwritten

3.2 Insertion and Removal of Points

The same mechanism that enables a seed to propagate throughout a grid to all 'correct' voxels in the construction algorithm can be readily used to insert a new seed in a valid DVD. To do this, first the voxel representing the new seed has to be assigned the new ID, and the list of voxels to process contains only that seed. Then, the neighbours will be reassigned if they are closer to the new seed then the old ones. In this way, the new DVD will be finished in as many rounds as it takes to dilate the new seed. This method will not only allow for the insertion of one point, but can be used to simultaneously insert multiple new points. Building the adapted DVD will then take only as many rounds as are necessary for the slowest Voronoi cell to be dilated. Incremental construction of the DVD is thus possible without designing new algorithms.

In vector space, removing a point from a 3D VD is problematic when degenerate cases are present (Devillers and Teillaud 2003; Ledoux et al. 2005). By contrast, in raster environment it is relatively straightforward and there are no degenerate cases. The idea behind the algorithm is depicted in Fig. 7 and is as follows. First, all the

voxels having the ID to be deleted must be identified. We start at the seed to delete and expands until we reach the border of the cell, and we keep a list of the boundary voxels. Second, these voxels are processed with the same dilate function as is used for the incremental insertion; that will 're-dilate' the Voronoi cell surrounding the removed Voronoi cell, overwriting the voxels that are to be deleted, finally yielding a recreated Voronoi diagram (Fig. 7c).

3.3 Natural Neighbour Interpolation

One direct possible use of the DVD is to perform a natural neighbour interpolation on the input dataset. The natural neighbour interpolation is a method that can be performed with the VD or the Delaunay triangulation as a basis (Sibson 1981), and is used in several fields that make use of datasets containing scattered data such as engineering, computer sciences and geosciences (Gold 1989; Watson 1992; Sambridge et al. 1995).

Although the concepts behind natural neighbour interpolation are simple and easy to understand, its implementation is far from being straightforward, especially in higher dimensions (Owen 1992; Perez and Traversoni 1996; Ledoux 2006). The main reasons are that the method requires the computation of two VDs – one with and one without the interpolation point – and also the computation of volumes of Voronoi cells. While these operations are error-prone in vector space, we have seen that in raster space they are simple and degenerate cases do not arise. Moreover, the volumetric calculations that can be difficult to implement in vector space are replaced by simply counting voxels, and multiplying the number of voxels by their (known) volume, as Fig. 8 shows.
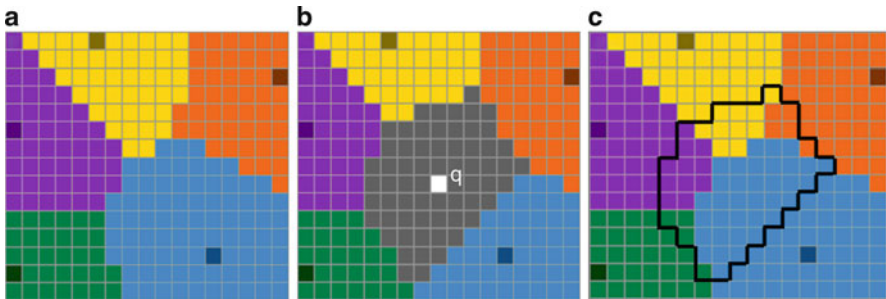

Fig. 8 Interpolation a DVD using natural neighbour interpolation. (**a**) The DVD. (**b**) The inserted point q, to be interpolated, and the corresponding new Voronoi cell (*grey area*). (**c**) The amount of voxels within the *black outline* corresponding to a specific seed, relative to the total amount of voxels within the outline, determines the weight for each of the corresponding seeds

4 Modelling Continuous Phenomena in Geoscience

To test our approach for the modelling of 3D geoscientific datasets, we have implemented a prototype whose architecture is described in Fig. 9. We have implemented all the algorithms described in the previous section with the Python language (using the well-known numpy package for n-dimensional arrays). To be able to analyse and visualise the 3D DVDs we use GRASS,[1] the only off-the-shelf GIS that permits us to handle 3D grids (as far as we know). Other GIS packages do handle 3D grids such as NetCDF, but they slice them to obtain 2D grids before carrying out analysis. For the visualisation of 3D grids, we found GRASS's capabilities limited and instead decided to use MayaVi2.[2] It is an implementation of the Visualization Toolkit (VTK)[3] and permits us to import different 3D formats (both vector and grids) and to perform several visualisation operations. While our solution does involve exporting and importing DVDs in different formats (we also wrote some conversion scripts), it has the benefits of using well-known tools and no new software has to be developed from scratch.

To show the possibilities the 3D DVD offers some examples of its usage are shown here. The dataset used is a geological dataset containing the concentration of a chemical substance; it contains 150 data points, each with (x, y, z, *attribute*). Normally a point dataset would be much larger, ranging from 10,000 to 100,000 points, however for this showcase a small dataset is used to be able to show details clearly. In Fig. 10 the distribution of the sample points in this test dataset is shown. It can be seen that most of the points are taken along lines parallel to the z-axis (blue), which indicates that the data is taken from for instance drilled wells. The anisotropy of the distribution of the sample points is also clear from this image. The 3D DVD of the dataset is shown in Fig. 2, and all the figures from this section were made with that dataset.

The 3D DVD, when combined with a GIS and a visualisation tool, can be used in different ways; what follows is an overview.

4.1 To Establish Neighbourhood Relations Between Unconnected Points

This is especially useful for anisotropically distributed datasets, because the VD permits us to find points that are both close to a location, and located 'around' that location. There is not much difference in the way the exact and the DVD handle these relations. Adjacency is in this case the key, which, for the exact version, is
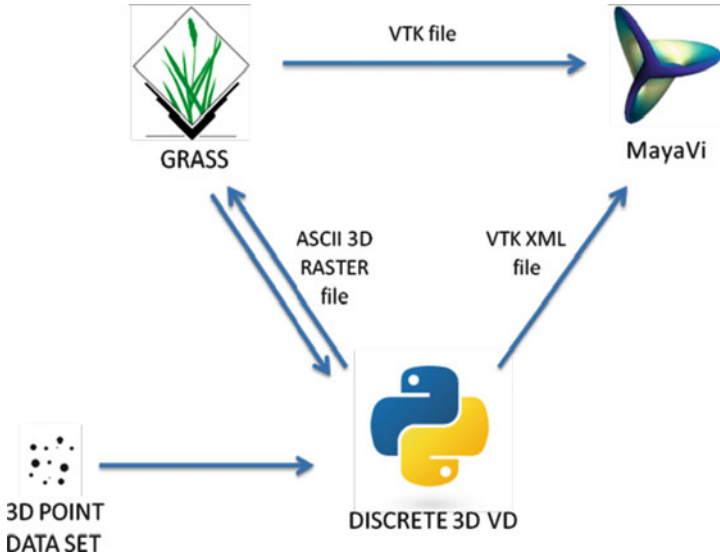
---

[1] http://www.grass.itc.it

[2] http://www.code.enthought.com/projects/mayavi/

[3] http://www.vtk.org

Fig. 9 Schematics showing the different software packages and the accompanying data formats

Fig. 10 Distribution of points in the test dataset. The anisotropy in distribution is clearly visible

usually explicitly stored. In the discrete representation of the 3D VD, although not implemented yet, storing adjacency would be easily added as a functionality – this would also permit us to derive the Delaunay tetrahedralization from a DVD.

4.2   To Determine the Area of Influence of Points

For the exact 3D VD, a Voronoi cell (which represents the area of influence) is described by a polyhedron, whereas in the DVD it is determined by the collection of voxels that carry the same ID. This means that it is relatively easy to determine the area of influence by simply finding all voxels that carry the same value. In this way sub-selections can easily be made and be used in combination with different spatial analysis operations, such as map algebra. Reclassification is one of these operations. Suppose a user wants to divide the attribute values into three classes (low, medium and high values). This can be easily done using 3D map algebra functionalities (Tomlin 1983; Mennis et al. 2005), as implemented in GRASS. Figure 11 shows a cut plane through a reclassified 3D DVD (the reclassification is based here on the value of the attribute).

4.3   To Support Numerous Visualisation and Spatial Analysis Operations

The exact 3D VD is difficult to visualise and analyse, and special tools have to be built (Ledoux and Gold 2008). By contrast, the 3D DVD can be used with off-the-shelf tools. Figure 12 shows that by slicing the data the distribution of points can be easily understood by the user. With tools such as MayaVi2, this plane can be controlled interactively in any directions. It must be noted here that the resulting slices do not represent a 2D VD.

As for the spatial analysis operations, it has already been mentioned that with the DVD Voronoi cells are easily identified, and since the raster format is used,
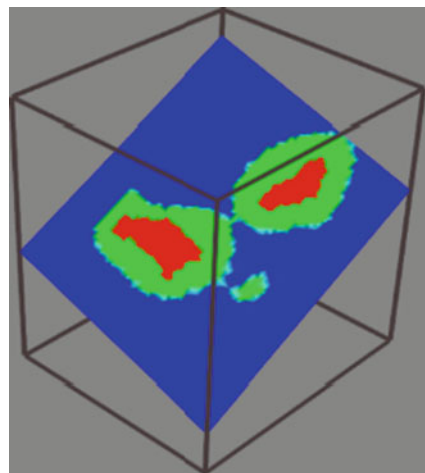

Fig. 11 A cut plane through a reclassified 3D DVD, showing a reclassification into three classes
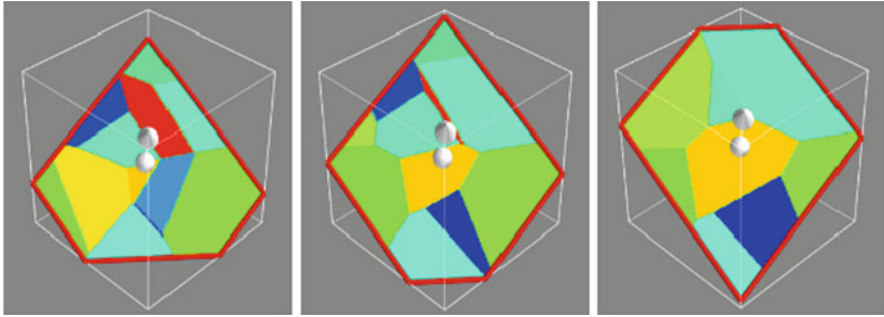
Fig. 12 Visualisation of a DVD by using a special slicing tool in MayaVi2, allowing for 3D visualization control. Three consecutive slices of one DVD are shown

performing spatial analyses on these structures is relatively easy. This is one of the points that the strength of the DVD may lie in, since spatial analysis operations for grids are simple, well-known and already implemented in GRASS. One example is the creation of a mask for a given grid if one wants to show only a particular section of the raster, based on the value of each voxel. Another example is the creation of neighbourhood (local) filters. To create these filters, a function is provided that enables the user to incorporate values of neighbouring voxels for each evaluated voxel. Using this function, it is possible create for instance raster representing the derivative of a continuous field (like the slope for grids of elevation).

4.4 As a Prerequisite for Natural Neighbour Interpolation

As explained in Sect. 3.3, the DVD is well-suited for recreating a 3D grid with the natural neighbour interpolation. Figure 13 shows some results of interpolation of the 3D DVD. Figure 13a shows one isosurface for the geological dataset, and Fig. 13b shows respectively the interpolated volume, a slice from it and the isosurface (same value as the one from the DVD). The isosurface created from the DVD tends to follow the edges of the boundary voxels of the Voronoi cells, whereas in the interpolated volume the isosurface is smooth. This is because the values between the boundary voxels of two Voronoi cells can be relatively far apart, resulting in very distinct isosurfaces. In the interpolated volume however, the difference in values is spread over a larger area, resulting in smoother isosurfaces.

5 Conclusions

We have shown that the 3D DVD provides a simple but effective tool for the modelling of 3D continuous datasets, and offers an alternative to using 3D grids or the 3D VD. It retains several of the advantages of the 3D VD, but has the advantages of
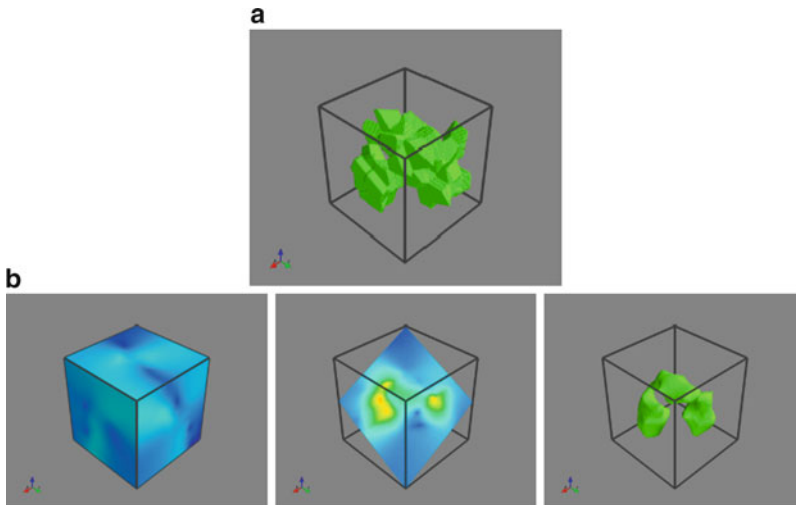
Fig. 13 (**a**) Isosurface created from the DVD. (**b**) The interpolated volume. *Left*: Its surface boundary. *Centre*: a slice from it. *Right*: an isosurface [same value as (**a**)]

being stored in a spatial model (grids) for which there are several tools available already. We have presented new algorithms to construct and dynamically modify 3D DVDs, these have the main advantage of being much less error-prone than the exact 3D VD and being conceptually simple (thus relatively simple to implement). The incremental construction, the dynamic deletion of seeds and the natural neighbour interpolation are all based on one operation that dilates spheres to update a DVD.

It should however be said that, as presented in this paper, the use of the 3D DVD for point-based geoscientific datasets is somewhat limited. We however believe it to be the first step for the construction and the modelling of other types of VDs, such as the generalised VD or the VD for other primitives. Indeed, the VD discussed in this paper is the ordinary VD, which assigns volumes to points based on the Euclidean distance. There are however many more types of VDs that are also used in different fields. Okabe et al. (1994) show how, with the help of 12 different generalised VDs, 35 different neighbourhood operations can be performed. The VDs they propose are generalised in space, in the assignment function (which determines to which seed a location is assigned to), and in the set of seeds that are used. Examples of these are the weighted VD and the high order VD. These can be very difficult to construct or modify in vector format in 2D (Gahegan and Lee 2000; Dong 2008), let alone in 3D. It would be interesting to see if the discrete versions of these generalised VDs can also be created with the algorithm we proposed in this paper. Our first analysis tells us that minor changes to the algorithm would permit us to indeed create weighted VDs, but we have not implemented them yet. The 3D VD for other primitives such as lines, surfaces and polyhedra is also an interesting extension of this project since, as far as we know, no known (robust) algorithms currently exist.

References

Anselin L (1999) Interactive techniques and exploratory spatial data analysis. In PA Longley, MF Goodchild, DJ Maguire, and DW Rhind, editors, *Geographical Information Systems*, second edition, pages 253–266. Wiley, New York.

Bak PRG and Mill AJB (1989) Three dimensional representation in a Geoscientific Resource Management System for the minerals industry. In J Raper, editor, *Three Dimensional Applications in Geographic Information Systems*, pages 155–182. Taylor & Francis, London.

Devillers O and Teillaud M (2003) Perturbations and vertex removal in a 3D Delaunay triangulation. In *Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 313–319. Baltimore, MD.

Dong P (2008) Generating and updating multiplicatively weighted Voronoi diagrams for point, line and polygon features in GIS. Computers and Geosciences, 34(4):411–421.

Field DA (1986) Implementing Watson's algorithm in three dimensions. In *Proceedings of the 2nd Annual Symposium on Computational Geometry*, pages 246–259. ACM Press, Yorktown Heights, New York.

Fisher PF (1997) The pixel: A snare and a delusion. International Journal of Remote Sensing, 18 (3):679–685.

Gahegan M and Lee I (2000) Data structures and algorithms to support interactive spatial analysis using dynamic Voronoi diagrams. Computers, Environment and Urban Systems, 24 (6):509–537.

Gold CM (1989) Surface interpolation, spatial adjacency and GIS. In J Raper, editor, *Three Dimensional Applications in Geographic Information Systems*, pages 21–35. Taylor & Francis, London.

Hoff KEI, Keyser J, Lin M, Manocha D, and Culver T (1999) Fast computation of generalized Voronoi diagrams using graphics hardware. In *SIGGRAPH '99: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pages 277–286. ACM Press, New York. ISBN 0-201-485605.

Hoffmann CM (1989) The problems of accuracy and robustness in geometric computation. IEEE Computer, 22:31–42.

Jones CB (1989) Data structures for three-dimensional spatial information systems in geology. International Journal of Geographical Information Systems, 3(1):15–31.

Kemp KK (1993) Environmental modeling with GIS: A strategy for dealing with spatial continuity. Technical Report 93-3, National Center for Geographic Information and Analysis, University of California, Santa Barbara.

Ledoux H (2006) *Modelling three-dimensional fields in geoscience with the Voronoi diagram and its dual*. Ph.D. thesis, School of Computing, University of Glamorgan, Pontypridd, Wales, UK.

Ledoux H and Gold CM (2008) Modelling three-dimensional geoscientific fields with the Voronoi diagram and its dual. International Journal of Geographical Information Science, 22 (5):547–574.

Ledoux H, Gold CM, and Baciu G (2005) Flipping to robustly delete a vertex in a Delaunay tetrahedralization. In *Proceedings of the International Conference on Computational Science and Its Applications – ICCSA 2005*, volume 3480 of *Lecture Notes in Computer Science*, pages 737–747. Springer, Singapore.

Li C, Chen J, and Li Z (1999) A raster-based method for computing Voronoi diagrams of spatial objects using dynamic distance transformation. International Journal of Geographical Information Science, 13(3):209–225.

Mennis J, Viger R, and Tomlin CD (2005) Cubic map algebra functions for spatiotemporal analysis. Cartography and Geographic Information Science, 32(1):17–32.

Okabe A, Boots B, and Sugihara K (1994) Nearest neighbourhood operations with generalized Voronoi diagrams: A review. International Journal of Geographical Information Systems, 8 (1):43–71.

Owen SJ (1992) *An implementation of natural neighbor interpolation in three dimensions*. Master's thesis, Department of Civil Engineering, Brigham Young University, Provo, UT.

Park SW, Linsen L, Kreylos O, Owens JD, and Hamann B (2006) Discrete Sibson interpolation. IEEE Transactions on Visualization and Computer Graphics, 12(2):243–253.

Perez C and Traversoni L (1996) Finite element simulation of shallow waters using natural neighbors techniques. In *Proceedings of the Computational Methods in Water Resources*, pages 239–245. Southampton, UK.

Rong G and Tan TS (2006) Jump flooding in GPU with applications to Voronoi diagram and distance transform. In *Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games*, pages 109–116. ACM Press, New York.

Rong G and Tan TS (2007) Variants of Jump Flooding Algorithm for Computing Discrete Voronoi Diagrams. In *Proceedings of the 4th International Symposium on Voronoi Diagrams in Science and Engineering*. IEEE Computer Society, Pontypridd, Wales, UK.

Russel D (2007) *Kinetic Data Structures in Practice*. Ph.D. thesis, Department of Computer Science, Stanford University, Stanford.

Sambridge M, Braun J, and McQueen H (1995) Geophysical parameterization and interpolation of irregular data using natural neighbours. Geophysical Journal International, 122:837–857.

Schueller A (2007) A nearest neighbor sweep circle algorithm for computing discrete Voronoi tessellations. Journal of Mathematical Analysis and Applications, 336(2):1018–1025.

Sibson R (1981) A brief description of natural neighbour interpolation. In V Barnett, editor, *Interpreting Multivariate Data*, pages 21–36. Wiley, New York.

Sugihara K and Inagaki H (1995) Why is the 3D Delaunay triangulation difficult to construct? Information Processing Letters, 54:275–280.

Tomlin CD (1983) A map algebra. In *Proceedings of the 1983 Harvard Computer Graphics Conference*, pages 127–150. Cambridge, MA.

van der Putte T (2009) *Using the discrete 3D Voronoi diagram for the modelling of 3D continuous information in geosciences*. Master's thesis, Geographical Information Management and Applications (GIMA), Delft University of Technology, Delft, The Netherlands.

Watson DF (1992) *Contouring: A guide to the analysis and display of spatial data*. Pergamon Press, Oxford, UK.

Zhao R, Li Z, Chen J, Gold CM, and Zhang Y (2002) A Hierarchical Raster Method for Computing Voronoi Diagrams Based on Quadtrees. In *ICCS '02: Proceedings of the International Conference on Computational Science*, pages 1004–1013. Springer, London, UK.