INTERPOLATION AS A TOOL FOR THE MODELLING OF THREE-DIMENSIONAL GEOSCIENTIFIC DATASETS

H. Ledoux*, C. Gold

GIS Research Centre, School of Computing, University of Glamorgan
Pontypridd CF37 1DL, Wales, UK.
hledoux@glam.ac.uk — christophergold@voronoi.com

KEY WORDS: Modelling, Three-dimensional, Algorithms, Visualisation.

ABSTRACT:

Although interpolation methods are commonly used in geographical information systems (GIS) for modelling two-dimensional fields, their three-dimensional counterparts are yet to be found in commercial software, and this despite the fact that more and more 3D datasets are available, especially in geosciences. In this paper, we discuss the generalisation to 3D of four weighted-average interpolation methods that are commonly used for 2D fields. We evaluate their properties when they are used with geoscientific data, and we also discuss implementation details. We restrict our analysis to methods that can be used in a context of interactive exploration of a dataset.

1 INTRODUCTION

Interpolation methods are an important part in a geographical information system (GIS) and have been used for years to model, among others, elevation data. They are crucial in the visualisation process (e.g. generation of contours lines), for the conversion of data from one format to another (e.g. from scattered points to raster), to have a better understanding of a dataset or simply to identify 'bad' samples. The result of interpolation—usually a surface that represents the real terrain—must be as accurate as possible because it often forms the basis for spatial analysis, for example runoff modelling or visibility analysis. Although interpolation helps in creating three-dimensional surfaces, it is intrinsically a two-dimensional operation for only the $x-y$ coordinates of each sample are used and the elevation is considered as an attribute.

With the new technologies available to collect information about the Earth, more and more three-dimensional data are collected. A typical dataset in geosciences has samples in three-dimensional space ($x-y-z$ coordinates) to which one or more attributes are attached. The attribute collected is usually a scalar, and typical examples are the salinity or the temperature of the water, the percentage of gold in the rock, or the humidity of the air. Samples are actually collected to study the spatial variability of an attribute over a certain extent in space; if a is the attribute studied, a continuous function $a = f(x, y, z)$ is defined, this is called a *field* (see Goodchild (1992) for more details). Because of the way they are collected, three-dimensional geoscientific datasets often have a highly anisotropic distribution. Geologic and oceanographic data are, for example, respectively gathered from boreholes and water columns: data are therefore usually abundant vertically but sparse horizontally. To model such datasets, three-dimensional interpolation methods that consider the specificities of the data must be used. While most of the interpolation methods used in GIS intuitively extend to 3D, it is not obvious that they preserve their properties or are appropriate for such datasets.

In this paper, we discuss the generalisation to three dimensions of some of the interpolation methods found in GIS or geoscientific modelling packages. We focus our attention on methods that can be applied in a context of *dynamic* or *interactive* modelling (Anselin, 1999; Bailey and Gatrell, 1995; Gold, 1993). This is an approach where the user does not only use standards operations (e.g. GIS spatial analysis or statistical methods) on a dataset, but actually interacts with it by manipulating and transforming it and then looks at the consequences of his manipulations. The emphasis is put on the interaction between the user and the dataset through modification and visualisation tools. A simple example of this approach is a spreadsheet tool (e.g. Microsoft Excel) where numerical data can be explored with the help of different statistical tests and charts (histograms, scatter points, pie charts, etc.). These charts are linked to the data, i.e. if some numbers in the dataset are changed, the charts are automatically updated. In the context of geoscientific modelling, one can think of an environment where the samples can be inserted, deleted or even moved at will by the user, and he gains insight about the spatial variability of the field studied by manipulating the dataset and observing the results. A key factor in interactive analysis is the speed at which each function is performed, to ensure that the user gets an (almost) instantaneous result from a query or an operation, so that he is not disturbed by long waits. We discuss in Section 2 the properties that an interpolation method for geoscientific data should have, and we present in Section 3 four methods. For each, we present and evaluate their properties in 3D space, and we also discuss implementation details to ensure that they are computationally efficient.

2 WHAT IS A GOOD INTERPOLATION METHOD?

Given a set of samples to which an attribute a is attached, spatial interpolation is the procedure used to estimate the value of the attribute at an unsampled location x. To achieve that, it creates a function f, called the interpolant, that tries to fit the samples as well as possible. Interpolation is based on *spatial autocorrelation*, that is the attribute of two points close together in space is more likely to be similar than that of two points far from each other. Watson (1992), in his authoritative book, lists the essential properties of an 'ideal' interpolation method for bivariate geoscientific datasets. These properties can realistically be present for trivariate datasets, and are as follows:

1. **exact**: the interpolant must 'honour' the data points, or 'pass through' them.

*Corresponding author.

2. **continuous**: a single and unique value must be obtained at each location. This is called a C^0 interpolant in mathematics.

3. **smooth**: it is desirable for some applications to have a function for which the first or second derivative is possible everywhere; such functions are respectively referred to as C^1 and C^2 interpolants.

4. **local**: the interpolation function uses only some neighbouring samples to estimate the value at a given location. This ensures that a sample with a gross error will not propagate its error to the whole interpolant.

5. **adaptability**, i.e. the function should give realistic results for anisotropic data distributions and/or for datasets where the data density varies greatly from one location to another.

In the context of dynamic/interactive modelling, we add to this list:

6. the method must be **computationally efficient**.

7. the method must require as little input as possible from the user, i.e. it should be **automatic** and not rely on user-defined parameters that require *a priori* knowledge of the dataset.

We describe in the next section four interpolation methods, commonly found in GIS and geoscientific software, that respect some of the criteria of an ideal method for an interactive system. The major omission from our list is Kriging (Matheron, 1971; Oliver and Webster, 1990), which, although possessing many of the ideal properties and being popular in many application domains because it produces an interpolant that minimises the error variance at each location, cannot be realistically be used in an dynamic environment. Indeed, the Kriging estimation is based on a function characterising the dependence between the attributes of any two samples that are at a given distance from each other. This function is obtained by studying the variance of the attributes according to the distance and the direction, and fitting a simple function. This is done manually by the user, and is a somewhat time-consuming process that is impossible to do every time the dataset is modified.

The four interpolation methods discussed are all weighted-average methods, which are methods that use only some sample points, to which weights ($^\circ$importance) are assigned. The interpolation function f of such methods have the following form:

$$f(x) = \frac{\sum_{i=1}^{k} w_i(x)\, a_i}{\sum_{i=1}^{k} w_i(x)} \tag{1}$$

where $w_i(x)$ is the weight of each neighbour p_i (with respect to the interpolation location x) used in the interpolation process, and a_i the attribute of p_i.

3 FROM TWO TO THREE DIMENSIONS

3.1 Nearest Neighbour

Nearest neighbour is a simple interpolation method: the value of an attribute at location x is simply assumed to be equal to the attribute of the nearest data point. Given a set S of data points in a d-dimensional space, if interpolation is performed with this method at many locations close to each other, the result is the Voronoi diagram (VD) of S, where all the points inside a Voronoi cell have the same value (see Figure 1 and Figure 2 for respectively the 2D and 3D examples). The Voronoi cell of a point $p \in S$, defined $\mathcal{V}_p$, is the set of points $x \in \mathbb{R}^d$ that are closer to p than to any other point in S. In 2D, $\mathcal{V}_p$ is a convex polygon, and in 3D it is a convex polyhedron. Observe that the size and the shape of a Voronoi cell is determined by the distribution of the points: when the data is dense the cells are smaller. The union of the Voronoi cells of all generating points $p \in S$ form VD(S). The VD actually creates a piecewise model, where the interpolation function inside each Voronoi cell is a constant function.

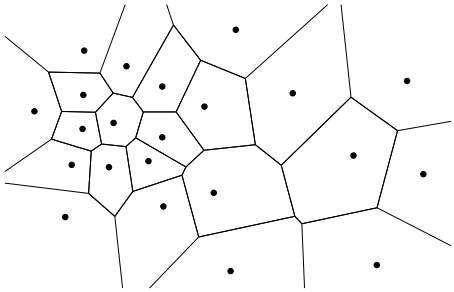


Figure 1: The VD for a set of points in the plane.

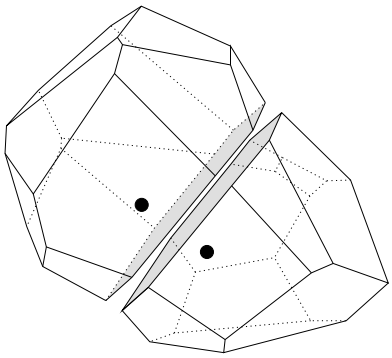
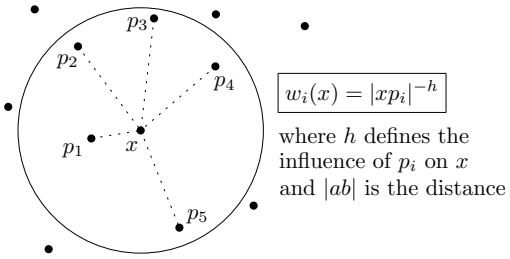


Figure 2: Two 3D Voronoi cells adjacent to each other.

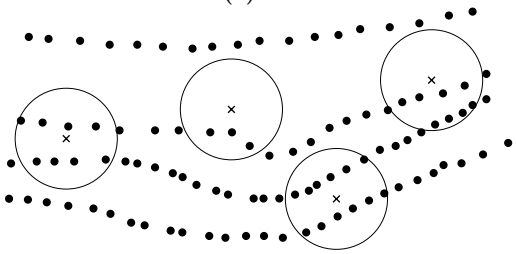
Although the method possesses many of the properties listed in Section 2 (it is exact, local and can handle anisotropic data distributions), it cannot be used with geoscientific data if realistic results are wanted because it fails lamentably properties 2 and 3. The interpolation function is indeed discontinuous at the border of cells. It can nevertheless be useful in some applications: it is for example often used in remote sensing to avoid averaging or blurring the resulting image, and it is also useful for nominal or ordinal data, e.g. rock types.

The implementation of the method sounds easy: simply find the closest data point and assign its value to the interpolation location. The difficulty lies in finding an efficient way to get the closest data point. The simplest way consists of measuring the distance for each of the n points in the dataset, but this is too slow for large datasets. To speed up this brute-force algorithm, auxiliary data structures that will spatially index the points must be used. They usually subdivide hierarchically the space into cells (usually squares or rectangles) and build a tree that uses $\mathcal{O}(n)$ space; examples of such trees are the KD-tree, the R-tree and the octree (see van Oosterom (1999) for a survey of two-dimensional structures: the three-dimensional methods are simple extensions). Thus, to find the nearest neighbour, it suffices to navigate in the tree and simply test the points in the adjacent cells. Locating the cell containing a test point can for example be done efficiently (in $\mathcal{O}(\log n)$ time) with a KD-tree.

Another solution consists of building the VD for the set of points and identifying inside which cell the interpolation point lies. In



(a)



(b)

Figure 3: **(a)** Inverse distance to a power interpolation. **(b)** Problems with anisotropic datasets.

fact, the Delaunay triangulation (DT)—which is the dual of the VD—can also be constructed as the two structures are equivalent, i.e. that the knowledge of one structure implies the knowledge of the other. Many algorithms exist to construct a three-dimensional VD/DT, see for example the incremental methods of Watson (1981) or Edelsbrunner and Shah (1996). More details about the latter method are available in Section 3.4. For a set of n points in $\mathbb{R}^3$ these algorithms construct the VD/DT in $\mathcal{O}(n^2)$, which is worst-case optimal since the complexity of the VD/DT for some distribution of points is quadratic. Finally, identifying the cell containing a query point is very efficient and is expected to take only $\mathcal{O}(n^{1/4})$ when the points are randomly distributed (see Mücke et al. (1999) for the details).

3.2 Distance-based Methods

Distance-based methods are probably the most known methods and they are widely used in many fields. As shown in Figure 3a, in two dimensions they often use a 'searching circle', whose radius is user-defined, to select the data points p_i involved in the interpolation at location x. The weight assigned to each is typically based on the square of the distance from x to p_i. Other weights can also be used, see Watson (1992) for a discussion. The size of the radius of the searching circle influences greatly the result of the interpolation: a very big radius means that the resulting surface will be smooth or 'flattened'; on the other hand, a radius that is too small might have dramatic consequences if for example no data points are inside the circle. A good knowledge of the dataset is thus required to select this parameter.

This method has many flaws when the data distribution is highly anisotropic or varies greatly in one dataset because a fixed-radius circle will not necessarily be appropriate everywhere in the dataset. Figure 3b shows one example where one circle, when used with a dataset extracted from contour lines, clearly gives erroneous results at some locations. The major problem with the method comes from the fact that the criterion, for both selecting data points and assigning them a weight, is one-dimensional and therefore does not take into account the spatial distribution of the data points close to the interpolation location. A criterion based on areas and volumes for respectively bivariate and trivariate data yields better results, as Sections 3.3 and 3.4 explain.

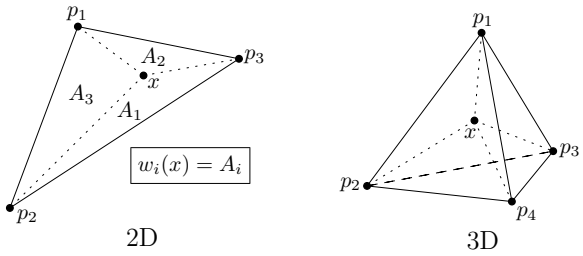


Figure 4: Barycentric coordinates.

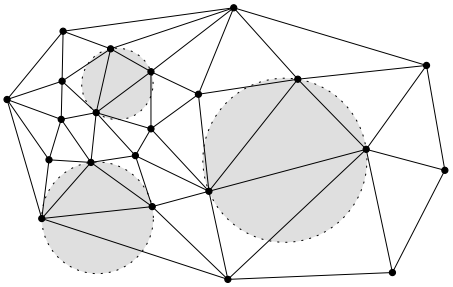


Figure 5: Delaunay triangulation of the same set of points as in Figure 1.

The generalisation of this method to three dimensions is straightforward: a searching sphere with a given radius is used. The same problems with the one-dimensionality of the method will be even worse because the search must be performed in one more dimension. The method has too many problems to be considered for geoscientific datasets: the interpolant is not guaranteed to be continuous, especially when the dataset has an anisotropic distribution, and the criterion has to be selected carefully by the user.

The implementation problems are also similar to the ones encountered with the previous method, and an auxiliary data structure must be used to avoid testing all the points in a dataset.

3.3 Linear Interpolation in Triangles

This method is popular for terrain modelling applications and is based on a triangulation of the data points. As is the case for the VD, a triangulation is a piecewise subdivision of the space covered by the data points, and in the context of interpolation a linear function is assigned to each piece (each triangle). Interpolating at location x means first finding inside which triangle x lies (see Mücke et al. (1999) for an efficient method), and then the height is estimated by linear interpolation on the plane defined by the three vertices forming the triangle. This can be efficiently implemented by using barycentric coordinates, which are local coordinates defined within a triangle (see Figure 4 for details). To obtain satisfactory results, this method is usually used in 2D with a Delaunay triangulation because, among all the possible triangulations of a set of points in the plane, it maximizes the minimum angle of each triangle. In other words it creates triangles that are as equilateral as possible (see Figure 5). A Delaunay triangulation is a triangulation for which the circle circumscribed to each triangle is empty, i.e. it does not contain in its interior any other point in the set. This ensures that the three vertices used in the interpolation process will most likely be close to and around the interpolation location.

The generalisation of this method to 3D is as follows: linear interpolation is performed within each tetrahedron of a 3D triangulation. Note that the barycentric coordinates can be generalised to linearly interpolate inside a tetrahedron. Finding 'good' tetrahedra is however more difficult than finding good triangles

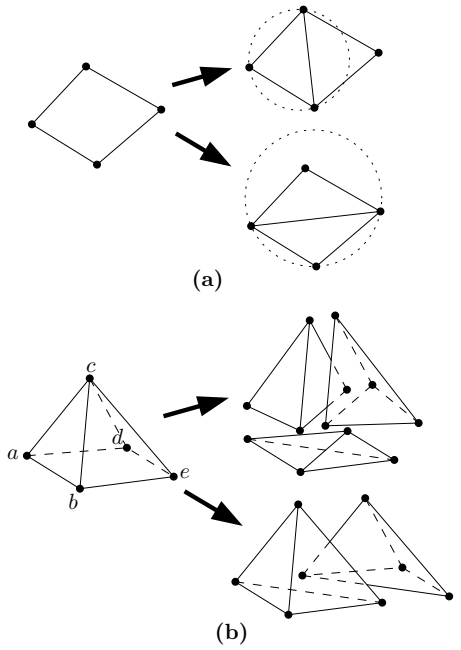


Figure 6: **(a)** A four-sided convex polygon can be triangulated by two different ways, but the Delaunay criterion guarantees that the triangles created will be as equilateral as possible. **(b)** In three dimensions, five vertices can be triangulated with either two or three tetrahedra. Although the triangulation at the bottom has two nicely shaped tetrahedra, they are not Delaunay (the point e is inside the sphere defined by the points a, b, c and d). The triangulation at the top is Delaunay, but contains one very thin tetrahedron spanned by the points a, b, d and e.

because the *max-min* property of Delaunay triangles does not generalise to 3D. A 3D DT can indeed contain some tetrahedra, called *slivers*, whose four vertices are almost coplanar (see Figure 6); interpolation within such tetrahedra obviously does not yield good results. Note that slivers do not have two-dimensional counterparts.

Despite the occasional presence of slivers, Delaunay tetrahedra still have a more desirable shape than arbitrary tetrahedra. Rajan (1991) shows that the smallest sphere containing a Delaunay tetrahedron is smaller than that of any other tetrahedron, i.e. that Delaunay tetrahedra tend to have a round shape. Also, Cheng et al. (2000) show that a triangulation where all the tetrahedra are well-shaped can be obtained as follows: start with the DT, assign some importance to some data points forming slivers, and modify locally some tetrahedra (with the help of *flips*, see next section) to obtain a *regular triangulation* of the set of points (Edelsbrunner and Shah, 1996).

As mentioned in Section 3.1, there exist different algorithms to construct and navigate efficiently in a 3D DT, and the linear interpolation itself is done in constant time since only four data points are involved. When used on a triangulation having well-shaped tetrahedra, the linear interpolation method possesses all the ingredients of an ideal method but one: the first (or second) derivative of the resulting function is not possible at the vertices, edges and faces of the triangulation. Akima (1978) solved this problem in two dimensions by using higher order functions in each piece of the triangulation, and the same could be done in three dimensions.

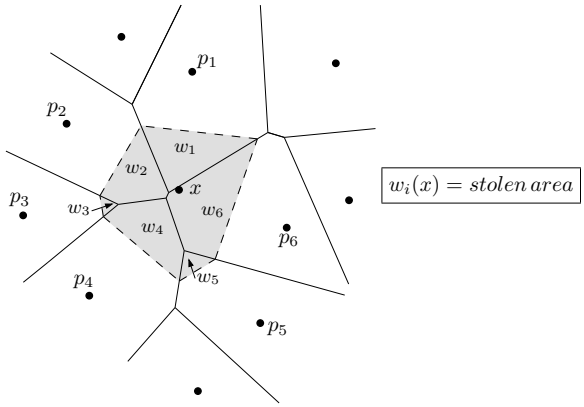


Figure 7: Natural neighbour coordinates in 2D for the interpolation point x. The shaded polygon is $\mathcal{V}_x$.

3.4 Natural Neighbour Interpolation

It has been shown by different researchers (Gold, 1989; Sambridge et al., 1995; Watson, 1992) that the natural neighbour interpolation method (Sibson, 1981) avoids most of the problems the other methods have with anisotropic datasets. This is a method based on the Voronoi diagram for both selecting the data points involved in the process, and assigning them a weight. It uses two VDs: one for the set of data points, and another one where a point x is inserted at the interpolation location. Let us describe the method for the two-dimensional case. The insertion of x locally modifies a VD: the Voronoi cell $\mathcal{V}_x$ of x 'steals' some parts of some Voronoi cells, as shown in Figure 7. This idea forms the basis of natural neighbour coordinates (Sibson, 1980, 1981), which define quantitatively the amount $\mathcal{V}_x$ steals from each of its natural neighbours. Let $\mathcal{D}$ be the VD(S), and $\mathcal{D}^+ = \mathcal{D} \cup \{x\}$. The Voronoi cell of a point p in $\mathcal{D}$ is defined by $\mathcal{V}_p$, and $\mathcal{V}_p^+$ is its cell in $\mathcal{D}^+$. The natural neighbour coordinate of x with respect to a point p_i is

$$w_i(x) = \frac{Area(\mathcal{V}_{p_i} \cap \mathcal{V}_x^+)}{Area(\mathcal{V}_x^+)} \tag{2}$$

where $Area(\mathcal{V}_{p_i})$ represents the area of $\mathcal{V}_{p_i}$. For any x, the value of $w_i(x)$ will always be between 0 and 1: 0 when p_i is not a natural neighbour of x, and 1 when x is exactly at the same location as p_i. A further important consideration is that the sum of the areas stolen from each of the k natural neighbours is equal to $Area(V_x^+)$. Therefore, the higher the value of $w_i(x)$ is, the stronger is the 'influence' of p_i on x. The natural neighbour coordinates are influenced by both the distance from x to p_i and the spatial distribution of the p_i around x. Note that the method is valid in any dimensions, for a Voronoi diagram exists in any dimensions. Hence, volumes are stolen in three dimensions, and hyper-volumes in higher dimensions.

Surprisingly, although many authors present the properties and advantages of the method, few discuss details concerning its implementation, and this is probably why its use is not widespread among the GIS community. The two-dimensional case is relatively easy to implement as efficient algorithms for constructing a VD/DT (Fortune, 1987; Guibas and Stolfi, 1985; Watson, 1981) and deleting a point from it (Devillers, 2002; Mostafavi et al., 2003) exist. In three dimensions, things get more complicated because the algorithms to modify a VD/DT and computing volumes are more complex (e.g. deleting a single vertex from a DT is for example rather tricky to implement, as Ledoux et al. (2005) show). To our knowledge, the only efficient algorithm is due to Boissonnat and Cazals (2002) and consists of mimick-

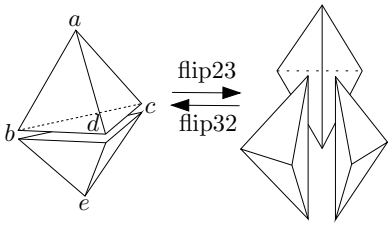


Figure 8: *flip23* and *flip32*.

ing the insertion of the interpolation point x to avoid deleting it afterwards. The stolen volume is obtained by a somewhat complicated method that 'orders' the vertices surrounding x and then decomposes the volume into tetrahedra. The implementation of this method is rather intricate.

We have therefore proposed a simple alternative that is valid in any dimensions (Ledoux and Gold, 2004). Our algorithm is based on the well-known *flipping* algorithm (Edelsbrunner and Shah, 1996) to insert a point in a d-dimensional Delaunay triangulation, and exploits the duality between a DT and the VD. Only minor modifications to Edelsbrunner and Shah's algorithm are needed to compute the natural neighbour coordinates. In brief, a flip is a local topological operation that modifies the configuration of adjacent tetrahedra in a triangulation (flips are actually possible in any dimensions, but for the sake of simplicity we only discuss the 3D case here). Consider a set $S = \{a, b, c, d, e\}$ of points in $\mathbb{R}^3$, as shown in Figure 8. There are two ways to triangulate S: either with two or three tetrahedra. In the first case, the two tetrahedra share a face bcd, and in the latter case the three tetrahedra all have a common edge ae. A *flip23* transforms a triangulation of two tetrahedra into another one containing three tetrahedra; a *flip32* is the inverse operation. Also, it has been shown that a single point can be inserted in a DT by applying a sequence of flips (Joe, 1991). Our idea was simply to 'remember' the sequence of flips used to insert a point, and reverse it to delete it. We also showed that only the volume of some faces of a Voronoi cell need to be computed in this context. The resulting algorithm is efficient (its time complexity is the same as the one for inserting a single point in a VD/DT), and we believe it to be considerably simpler to implement than other known methods, as only an incremental insertion algorithm based on flips, with some minor modifications, is needed.

Although computationally more intensive than the other methods, natural neighbour interpolation possesses all the properties enumerated in Section 2, except that the first derivative is undefined at the data points. To obtain a differentiable function everywhere, Sibson uses the weights defined in Equation 2 in a quadratic equation where the gradient at x is considered. To our knowledge, this method has not been used with success with real data and therefore we do not use it. Other ways to remove the discontinuities at the data points have been proposed: Watson (1992) explains different methods to estimate the gradient at x and how to incorporate it in Equation 1; and Gold (1989) proposes to modify the weight of each p_i with a simple hermitian polynomial so that, as x approaches p_i, the derivative of $f(x)$ approaches 0.

Modifying Equation 1 to obtain a continuous function can yield very good results in some cases, but with some datasets the resulting surface can contain unwanted effects. Different datasets require different methods and parameters, and, for this reason, modifications should be applied with great care.

4 DISCUSSION

We have shown that most of the weighted-average methods do generalise to 3D, but sometimes new problems appear, for example with triangulations in three-dimensions or with the computational efficiency. Also, the flaws present in some two-dimensional methods will often be amplified in three and higher dimensions. Natural neighbour interpolation seems to be the method that is the most appropriate for modelling geoscientific dataset because it is robust, entirely objective, and gives good results when the data distribution is highly irregular. We have already implemented in two and three dimensions the algorithm described in this paper and we hope our method will make it possible for the GIS community to take advantage of natural neighbour interpolation for modelling geoscientific data.

Acknowledgements

Part of this work was done when the first author was at the Hong Kong Polytechnic University, and we thank the support of the Hong Kong's Research Grants Council (Project PolyU 5068/00E).

REFERENCES

Akima, H., 1978. A method of bivariate interpolation and smooth surface fitting for irregularly distributed data points. *ACM Transactions on Mathematical Software*, 4(2):pp. 148–159.

Anselin, L., 1999. Interactive techniques and exploratory spatial data analysis. In P. Longley, M. Goodchild, D. Maguire, and D. Rhind, eds., *Geographical Information Systems*, chapter 17, pp. 253–266. John Wiley & Sons.

Bailey, T. C. and Gatrell, A. C., 1995. *Interactive spatial data analysis*. Wiley, New York.

Boissonnat, J.-D. and Cazals, F., 2002. Smooth surface reconstruction via natural neighbour interpolation of distance functions. *Computational Geometry—Theory and Applications*, 22:pp. 185–203.

Cheng, S.-W., Dey, T. K., Edelsbrunner, H., Facello, M. A., and Teng, S.-H., 2000. Sliver exudation. *Journal of the ACM*, 47(5):pp. 883–904.

Devillers, O., 2002. On deletion in Delaunay triangulations. *International Journal of Computational Geometry and Applications*, 12(3):pp. 193–205.

Edelsbrunner, H. and Shah, N. R., 1996. Incremental topological flipping works for regular triangulations. *Algorithmica*, 15:pp. 223–241.

Fortune, S., 1987. A sweepline algorithm for Voronoi diagrams. *Algorithmica*, 2:pp. 153–174.

Gold, C. M., 1989. Surface interpolation, spatial adjacency and GIS. In J. Raper, ed., *Three Dimensional Applications in Geographic Information Systems*, chapter 3, pp. 21–35. Taylor & Francis.

Gold, C. M., 1993. Forestry spatial decision support system classification, and the 'flight simulator' approach. In *Proceedings GIS'93: Eyes on the Future*, pp. 797–802. Vancouver, Canada.

Goodchild, M. F., 1992. Geographical data modeling. *Computers & Geosciences*, 18(4):pp. 410–408.

Guibas, L. J. and Stolfi, J., 1985. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Transactions on Graphics*, 4:pp. 74–123.

Joe, B., 1991. Construction of three-dimensional Delaunay triangulations using local transformations. *Computer Aided Geometric Design*, 8:pp. 123–142.

Ledoux, H. and Gold, C. M., 2004. An efficient natural neighbour interpolation algorithm for geoscientific modelling. In P. Fisher, ed., *Developments in Spatial Data Handling—11th International Symposium on Spatial Data Handling*, pp. 97–108. Springer.

Ledoux, H., Gold, C. M., and Baciu, G., 2005. Flipping to robustly delete a vertex in a Delaunay tetrahedralization. In *Proceedings International Conference on Computational Science and its Applications — ICCSA 2005*, LNCS 3480, pp. 737–747. Springer-Verlag, Singapore.

Matheron, G., 1971. The theory of regionalised variables and its applications. *Les Cahiers du Centre de Morphologie Mathématique de Fontainebleu*, 5.

Mostafavi, M. A., Gold, C. M., and Dakowicz, M., 2003. Delete and insert operations in Voronoi/Delaunay methods and applications. *Computers & Geosciences*, 29(4):pp. 523–530.

Mücke, E. P., Saias, I., and Zhu, B., 1999. Fast randomized point location without preprocessing in two- and three-dimensional Delaunay triangulations. *Computational Geometry—Theory and Applications*, 12:pp. 63–83.

Oliver, M. and Webster, R., 1990. Kriging: a method of interpolation for geographical information systems. *International Journal of Geographical Information Systems*, 4:pp. 313–332.

Rajan, V. T., 1991. Optimality of the Delaunay triangulation in $\mathbb{R}^d$. In *Proceedings 7th Annual Symposium on Computational Geometry*, pp. 357–363. ACM Press, North Conway, New Hampshire, USA.

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

Sibson, R., 1980. A vector identity for the Dirichlet tesselation. In *Mathematical Proceedings of the Cambridge Philosophical Society*, 87, pp. 151–155.

Sibson, R., 1981. A brief description of natural neighbour interpolation. In V. Barnett, ed., *Interpreting Multivariate Data*, pp. 21–36. Wiley, New York, USA.

van Oosterom, P., 1999. Spatial access methods. In P. Longley, M. Goodchild, D. Maguire, and D. Rhind, eds., *Geographical Information Systems*, volume 1, chapter 27, pp. 385–400. John Wiley & Sons.

Watson, D. F., 1981. Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. *Computer Journal*, 24(2):pp. 167–172.

Watson, D. F., 1992. *Contouring: A Guide to the Analysis and Display of Spatial Data*. Pergamon Press, Oxford, UK.