Representation, fields

Hugo Ledoux
h.ledoux@tudelft.nl

A field is a model of the spatial variation of an attribute over a spatial domain, and it is used to represent continuous geographical phenomena such as the elevation of a terrain, the salinity of a body of water or the humidity of the air (phenomena that can change over time). Fields are more problematic than objects to store in a computer first because they are intangible, and second because they are continuous (while computers are discrete machine). This entry presents an overview of the different strategies to represent fields as found in GIS-related applications. These strategies take into account the fact that while geographical fields are continuous, we first have to collect samples to study them. Higher-dimensional representations, where for instance time is integrated as an extra dimension to the spatial ones, are also discussed.

1 Introduction

While we all have a certain idea of what 'space' is because we constantly interact with it, most of us would be hard pressed to give a definition. The nature of space has been a debate for a long time, and no clear and uncontroversial definition exists because different disciplines (eg mathematics, physics and geography) have different definitions, and these are context-related. However, we know that humans conceptualise space with two contrasting views: the *object* and the *field* views. The former considers space as being 'empty' and populated with discrete entities embedded in space and having their own properties. We usually use this view when we reason about the spatial arrangement of the buildings and streets in our neighbourhood, or about the pieces of furnitures in our house. The latter view considers space as being continuous, and every location in space has a certain property (there is *something* at every location); entities are formed by clusters of properties. We usually use this view for geographical phenomena such as the amount of rainfall in a given province, or the temperature of a body of water.

When we want to represent and store a certain part of space in a computer, ie to build a spatial data model and realise it with a data structure, the field-view approach is much more problematic than its counterpart. The causes of this are many. First, a field is intangible and is not part of our intuitive knowledge. It is easy for us to see and describe entities such as churches and rivers, but, although we can imagine a field, it is somewhat of an abstract concept. Second, fields are by definition continuous while computers are discrete machines. Third, there is much confusion among users because for years the GIS world has reduced each view to one of the two spatial models: "object = vector model" and "field = raster model". Goodchild [1992], among others,

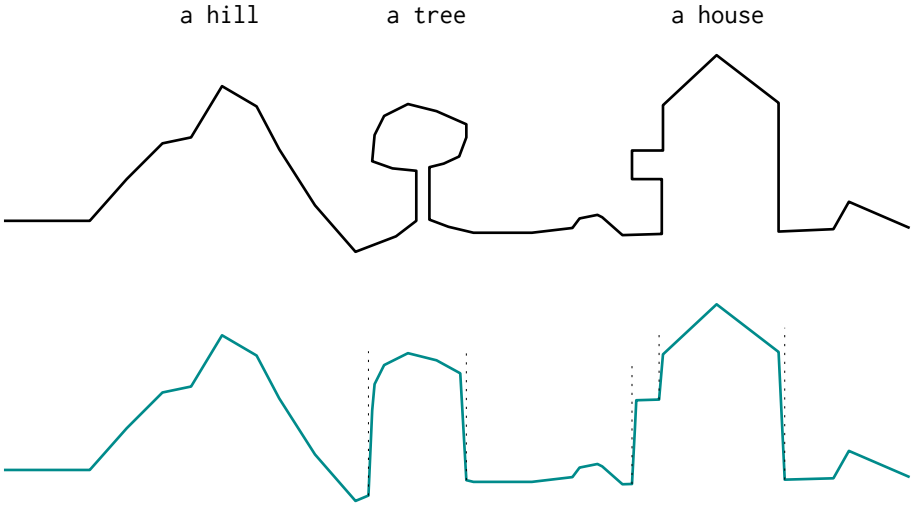

Figure 1: Top: a profile view of a terrain, with one tree and one house with a balcony. Bottom: the profile view if the elevation is a dependent variable of the (x, y); notice that the walls of the house and the sides of the tree are not fully vertical (dotted lines)

explains that this standpoint is simply false as both views can be stored with either spatial model. For example, the elevation of a terrain can be represented with a raster or with a set of triangles, and buildings in a city with polygons or with a binary raster.

This entry begins with the definition of a field, as usually used in GIScience. It should be noticed that fields as found in GIS-related disciplines are not restricted to two dimensions, and we consider three and higher dimensions (for spatial-temporal modelling). Different strategies to represent higher-dimensional fields in a computer are described in Section 3. These strategies take into account the fact that while geographical fields are continuous, we first have to collect discrete samples to study them. Section 4 presents different interpolation methods that can be used to reconstruct the continuity of a field from its set of samples. Section 5 presents different spatial models to *discretise* a field into a finite number of parts; these spatial models can be based on the samples or be arbitrarily subdividing the space. For each of the spatial models, potential data structures for an efficient implementation in a computer are briefly discussed.

2 What is a field?

The definition usually used in GIScience is borrowed and adapted from physics. Physicists in the 19th century developed the concept of a *force field* to model the gravitational force, where a force (a vector with an orientation and a length) has a value everywhere in space, and changes from location to location. For most GIS applications, the vector assigned to each point is replaced by a scalar value, and we obtain *scalar fields*; unless otherwise explicitly stated, in the following all fields are scalar.

A field is a model of the spatial variation of an attribute a over a spatial domain, we assume this domain to be $\mathbb{R}^d$, the d-dimensional Euclidean space. It is modelled by a function mapping one point p in $\mathbb{R}^d$ to the value of a, thus

$$a = f(p)$$

The function can theoretically have any number of independent variables (ie the spatial domain can have any dimensions), but in the context of geographical phenomena the function is usually bivariate (x, y) or trivariate (x, y, z).

Bivariate functions are commonly used in GIS to model the elevation of a terrain. However, it should be noticed that the dependent variable a (the elevation in this case) can have one and only value for each location (x, y). This restricts the real-world cases that can be modelled because, as shown in Figure 1, vertical surfaces (eg walls of a building if we model all man-made objects with the terrain to construct a digital surface model), overfolds (eg the balcony of a house) and caves are impossible to represent. Modelling the elevation (or other attributes) with a bivariate func-

tion is sometimes referred to as '2.5D GIS'; we model the function with a surface (ie topologically a two-dimensional object) but we embed this surface in $\mathbb{R}^3$.

Geographical phenomena tend to vary over time, and, as a consequence, we obtain *dynamic fields*. If we assume that time is continuous and linear, it is possible to incorporate it as an extra dimension, perpendicular to the spatial ones. It becomes a new independent variable, and if we already had a trivariate function (x, y, z) then it becomes a quadrivariate one (x, y, z, t). A dynamic field representing the salinity of a body of water over time is thus modelled in $\mathbb{R}^4$.

The value of the attribute a can be measured according to different scales, and depending on the scale used different types of fields will be obtained. We distinguish between two types of fields:

Continuous fields: the scale of measurement is a real number $\mathbb{R}$. Most fields studied in the geosciences fall into this category: temperature, precipitation or salinity are examples because they can be measured precisely. The measurements scales 'ratio' and 'interval' are of this type.

Discrete fields: the possible values of an attribute a are simply labels. An example is a map of Europe where each location contains the name of the country. The labels can also be ordered, eg a certain region can be categorised from 1 to 5 according to its suitability for producing wine: 1 being poor, and 5 excellent. These are respectively the scales 'nominal' and 'ordinal'. It should be observed that in the former case the only comparison possible between two values is whether they are the same or not; in the latter case operations such as 'greater than' and 'less than' can be made, but no arithmetic operations are possible. Discrete fields act also as objects because each regions can be modelled as a unique object.

Notice that the terms 'continuous' and 'discrete' can be misleading because in mathematics continuity has a slightly different meaning. Here they refer only to the scale of measurement, and not to the spatial continuity of a field. Indeed, both types of fields are spatially continuous, since they are represented by a function and there exists a single and unique value at every location in space; this is called a C^0 function in mathematics. For the representation of fields, it is however often desirable 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.

3 Strategies to represent fields in a computer

The representation of a field in a computer faces many problems. First, fields are continuous functions, and, by contrast, computers are discrete machines. Fields must therefore be *discretised*, ie broken into finite parts. Second, in practice it is usually impossible to measure continuous phenomena everywhere, and we have to resort to collecting samples at some finite locations and reconstructing fields from these samples. The discretisation task therefore begins at the acquisition phase, and is affected by the acquisition tools and techniques. This fact is aggravated for fields as found in GIS-related disciplines because, unlike disciplines like medicine or engineering, we seldom have direct access to the whole object of interest. Indeed, to collect samples in the ground we must dig holes or use other devices (eg ultrasound penetrating the ground); underwater samples are collected by instruments moved vertically under a boat, or by automated vehicles; and samples of the atmosphere are collected by devices attached to balloons or aircrafts. Moreover, because of the way they are collected, geoscientific datasets often have a highly sparse and anisotropic distribution: as shown in Figure 2, the distribution can be for instance dense vertically (with a sample every 2m in that real-world case) but extremely sparse horizontally (water columns are located at about 35km from each others). Even if a sensor is used to collect samples, the result (eg an image with pixels) is not a complete representation since each pixel usually averages the value of the studied phenomenon over the pixel area, or each pixel represents the value for one location inside the pixel. Third, another peculiar property of geoscientific datasets is that the samples they contain represent the spatial variability of a given attribute only at time t, and many geographical phenomena in geosciences change and evolve relatively quickly over time.

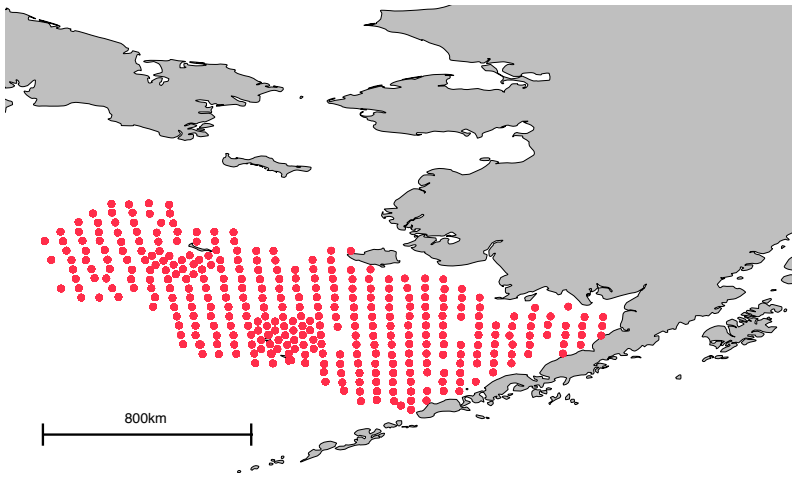

Figure 2: An oceanographic dataset in the Bering Sea in which samples are distributed along water columns. Each red point represents a (vertical) water column, where samples are collected every 2m.

3.1 What is needed to represent a field?

To represent a field in a computer, two things are needed:

1. a set of samples that were collected to study the field, eg these can be the elevation of a terrain obtained with airbone laserscanning or the humidity of the soil collected with a certain device. This set forms the 'ground truth' of the field, and its importance is such that it has been dubbed the *meta-field* by Kemp and Včkovski [1998].

2. a set of rules to obtain one and only one value for the attribute studied at any location, in other words, to reconstruct the continuity of the phenomenon studied from the discrete samples. This operation is referred to as spatial interpolation.

Section 4 presents the first strategy: storing the original samples with the parameters of the *global* spatial interpolation method that is best suited to the distribution of the samples and their accuracy. Notice that this strategy permits us to reconstruct the continuity of a field from the samples by calculating the value of the attribute, but that this value is not persistently stored in the computer.

Section 5 presents an alternative strategy: the spatial interpolation function used is *piecewise* (instead of being global). That is, the field is tessellated, or partitioned, into several pieces, and for each of these we assign an interpolation function describing the spatial variation. Piecewise models imply that a supporting data structure is constructed, and stored, to represent the tessellation (it is however possible to construct on-the-fly some well-known structures only when they are needed since they have formal rules). Some of these tessellations partition arbitrarily the space, while some are based on the set of samples that were collected.

3.2 Incomplete representations

In the GIS literature, different representations for two-dimensional fields are often listed. For instance, these six are common:

1. regularly spaced sample points;
2. irregularly spaced sample points;
3. contour lines;
4. rectangular cells (raster);
5. triangulated irregular networks (TIN);
6. planar partition with arbitrary polygons.

While these are by no means exhaustive, they represent how a typical practitioner would store her two-dimensional fields. Theoretically, these representations all generalise to three and higher dimensions (these are discussed in Section 5).

Although the first three representations are commonly used, they are *incomplete*: the set of rules to reconstruct the continuity of a field at unsampled locations is not explicitly given. These should therefore not be considered valid representations of a field. It should be noticed that while finding a spatial interpolation function suitable for the first case is simple (a few are discussed below), for the third case it is a complex task. Indeed, Dakowicz and Gold [2003] demonstrate that using simple rules (nearest-neighbour for instance) yields fields that are not realistic and have bad slope, which is in practice problematic for several applications. Obtaining good slope from contour lines is possible, but is in practice a complex process.

4 Samples + global interpolation rules

We consider as 'samples' any data that were collected to study a phenomenon. In practice, these data can take different forms:

1. a set of scattered points;
2. a set of lines. The most common case is a set of contour lines coming from a topographic map;
3. a set of scattered polygons (or polyhedra or polytopes) to which one value is attached. Although this case is theoretically possible, in practice this is very rare. It is nevertheless always possible to discretise each primitives into a set of points, and then perform interpolation.

Observe that it is also possible to collect directly a raster image, coming from remote sensing, where the value of each pixel represents the temperature of the sea for instance. The value is the average of the all the values within the pixel, and this case is considered as being already tessellated (the pixels) and is discussed in the next section.

Interpolation methods are rather difficult to categorise because they are based on different paradigms, and some methods fall into more than one category. No attempt is made in this entry to introduce categories; the reader is referred to the other entries in the book *Interpolation, inverse-distance weighting* and *Interpolation, Kriging*.

The following are a few examples of 'global' interpolation methods, ie the same function is applied everywhere to a subset of (regularly or irregularly spaced) sample points (and the function is not piecewise, as in Section 5). Observe that these methods can all be used in three and higher dimensions.

Nearest neighbour: the simplest interpolation method. The value of an attribute at location x is assumed to be equal to the attribute of the nearest sample point.

Inverse-distance weighting (IDW): as described in the chapter *Interpolation, inverse-distance weighting*, it requires different parameters to define which points are involved in the interpolation at a given location, and also the power must be defined. The parameter to define the points is usually a searching radius, which can be used in higher dimensions.

Kriging: as described in the chapter *Interpolation, Kriging*, while the modelling of a dataset is a difficult and time-consuming task, the output of the modelling process (a function characterising the dependence between the attributes of any two samples that are at a given distance from each other) can be simply stored as a string.

Notice that while it is theoretically possible to find a polynomial function of high degree for any set of real-world samples, in practice this is very rarely used because, as the degree of the polynomial increases, so can the error at unsampled locations. That is, the error at the samples is zero, but between two samples the function returns a value that can be far higher or lower than the two values of the samples (known as the Runge's phenomenon). Piecewise functions formed by many polynomial functions, known as splines, are usually preferred and used in GIS, see Mitasova and Mitas [1993] for instance.

Storing explicitly the interpolation method is efficient because in practice only a few parameters have to be stored. For instance, in the case of IDW, it suffices to store the search radius and the power used (two numbers). Finding the appropriate values for interpolation parameters is a difficult and time-consuming task, as the user must have a good understanding of the spatial distribution of the samples, and of the details of the method. A vivid example is Kriging, with which experienced users can obtain very good results, but which also leaves newcomers clueless with its many parameters and options. Using Kriging with the appropriate parameters leads to results having statistically minimum variance, however, simply using the default values for the parameters will most likely lead to unreliable results. Thus, it is better to let specialists perform the modelling of a dataset and find the best interpolation method (and the parameters to use), and store only the results of that modelling. The practitioners can then reconstruct the field without having to find the optimal parameters, provided that they have access to software to perform the interpolation.

5 Piecewise spatial models

In this section different spatial models needed to represent tessellations of space (this space can be in our case two-, three-, or four-dimensional) are presented and discussed. While the same spatial models can in theory be used in any dimensions, their implementations with a specific data structures for the three- and higher-dimensional cases are more complex than the two-dimensional case, and for this reason, potential data structures are also briefly discussed.

Once the d-dimensional space covered by a field is tessellated, the field function becomes *piecewise*: to each d-dimensional element of the tessellation—which is named a *cell* in the following—is assigned a function describing the spatial variation in its interior. This function is usually a simple mathematical function:

- constant function: the value of the modelled attribute is constant within one cell, for example to represent a discrete field;
- linear function;
- higher-order function.

In general, we classify the tessellations of space into three categories (as shown in Figure 3): *regular*, *hierarchical*, and *irregular*.

5.1 Regular Tessellations

All the cells have the same shape and size. The most common regular tessellation in GIS is by far the grid (or raster representation), in which the cells are squares in 2D (usually called *pixels*, a portmanteau of 'picture' and 'element', as an analogy to digital images). However, while they are not common in practice, other regular shapes are possible, such as hexagons or triangles. Regular tessellations generalise to higher dimensions: for instance, in 3D, instead of squares we have cubes (called *voxels*; 'volume' and 'element'), and *tesseracts* in 4D.

Observe that a regular tessellation often arbitrarily tessellates the space covered by the field without taking into consideration the objects embedded in it (the samples) or the nature of the phenomenon modelled. This is in contrast with irregular tessellation in which, most of the time, the cells constructed are constrained by the samples. Kemp [1993] states that "[a 2D grid] requires us to enforce a structure on reality rather than allowing reality to suggest a more appropriate structure for our analysis". In practice this means that, unless a regular tessellation is obtained from an image sensor (remote sensing or photogrammetry), we can assume that it was constructed from a set of samples by using spatial interpolation. Converting sample points to cells is not optimal because the original samples, which could be meaningful points such as the summits, valleys or ridges of a terrain, are not necessarily present in the resulting grid.

Concrete example: a 2D grid. A 2D grid, stored for instance with the GeoTIFF format, is thus a piecewise representation of a 2D field: a regular tessellation where each cell has a constant function. The value assigned to each cell is either an estimation previously obtained by spatial

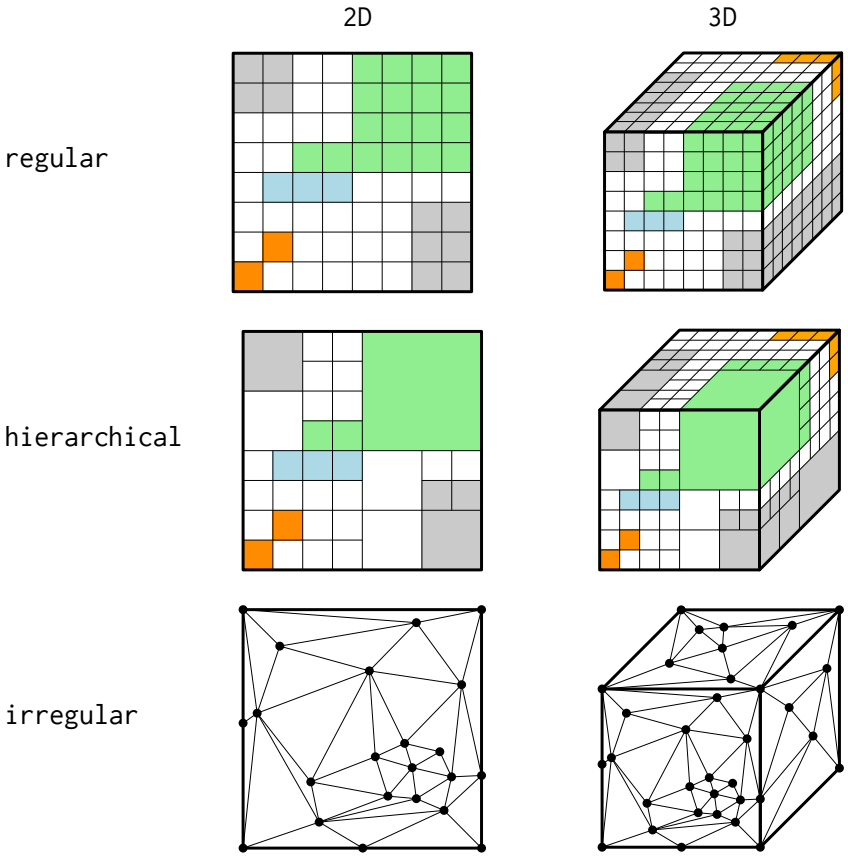

Figure 3: Type of tessellations and one example for each in two and three dimensions.

interpolation, or, in case of aerial imagery, the average values inside that cell. However, for a given grid, it is usually unclear if the value of a cell is for its centre, or for one of its vertices (and if it is the case, for which one?). Different formats have different rules, and converting a field represented with one format to another one (while retaining the same cell resolution and spatial extent) can shift the value from the centre to the top-left corner for instance.

Concrete example: a 3D/4D grid. As is the case for 2D grids, 3D/4D grids are piecewise representations, and usually a constant function is used for each cell. Several implementations of a voxel-based GIS exist, notably the open-source system GRASS[1] in which diverse operations such as interpolation and visualisation are possible. Also, most of the earlier attempts at building 3D GISs were made using voxels. Four-dimensional grids can be stored, among others, in the netCDF format[2]. It can actually be used for d-dimensional grids, with different spacing for different dimensions. They are binary and spatially structured, which means that parts of a dataset can be efficiently retrieved and processed.

While the function assigned to each cell is often a constant, other functions can be used. Notice that a linear function is impossible (unless, for instance, in 2D, the 4 vertices of the cells are coplanar), and instead *bilinear interpolation* is used. The method, which can be seen as an extension of linear interpolation, performs linear interpolation in one dimension (eg along the x axis), and then in the other dimension (y axis). The result is independent of the order in which the axes are used, and can be generalised to higher dimensions.

The wide popularity of regular tessellations in 2D and 3D GIS applications is probably due to simplicity and to the fact that they permit us to easily integrate 2D remote sensing images and fields. Indeed, a grid is naturally stored in a computer as an array (each cell is addressed by its position in the array, and only the value of the cell is stored), and thus the spatial relationships between cells are implicit. This is true for any dimensions, thus, contrary to other tessellations,

[1] Geographic Resources Analysis Support System (`http://grass.itc.it`).
[2] `http://www.unidata.ucar.edu/software/netcdf/`

grids are very easy to generalise to higher dimensions. The algorithms to analyse and manipulate (Boolean operations such as intersection or union) are also straightforwardly implemented in a computer. On the other hand, grids also suffer problems. First, the size of a grid can become massive for data at a fine resolution; this problem gets worse in higher dimensions. Second, grids scale badly and are not rotationally invariant, ie if the coordinate reference system used is changed, then the grid needs to be reconstructed to obtain regular cells whose boundaries are parallel to the axes of the reference system. To assign a new value to the transformed cells, spatial interpolation is needed, which is often performed not by re-using the original samples, but by using the values of the neighbouring cells. Unfortunately, each time a grid is transformed its information is degraded because not the original samples are used, but interpolated values.

5.2 Hierarchical tessellations

Hierarchical tessellations attempt to reduce the number of cells in a tessellation by merging the neighbouring cells having the same value (thus yielding cells of different sizes). While both regular and irregular tessellations can be hierarchical, in the context of the representation of fields, the former is more relevant and is often used in practice. Irregular hierarchical tessellations, usually triangulations, are used for obtaining multi-resolution models [De Floriani and Magillo, 2002], or act as a spatial index for efficiently accessing triangles. Their use as a support to construct a piecewise function is not necessary since, as shown in Section 5, irregular tessellations have cells of different sizes and shapes. Consequently, only hierarchical regular tessellations are discussed in the following.

A commonly used hierarchical structure in two dimensions is the *quadtree*, which is a generic term for a family of tessellations that recursively subdivide the plane into four quadrants. As is the case for grids, quadtrees are relatively easily implemented in a computer because they are trees in which each node has exactly four children, if any. Different variants of quadtrees exist, but the most relevant type in the context of this chapter is the *region quadtree*. As shown in Figure 3, it indexes and merges adjacent grid cells to save space by recursively subdividing the space into four squares of equal size, until every square contains one homogeneous region (based on the attribute of every grid cell, the colour in the figure). A region quadtree offers a more adaptive subdivision of space than a regular tessellation because smaller cells are present only when needed (to capture more details).

The region quadtree generalises to higher dimensions. For instance, in three dimensions, the space is recursively subdivided into eight octants, and thus the tree is called an *octree* (each node has eight children). In four dimensions, space is recursively subdivided into 16 cells (each one of them being a tesseract).

Examples of the use of regular hierarchical tessellations in GIS for representing fields are more numerous in 3D than in 2D, probably because the space issue (ie space used to store a field in a computer) is more critical. However, it is possible that a GIS stores and handles internally 2D grids as region quadtrees—the user does not need to know as no information is lost in the conversion, and important space and processing gains can be achieved.

The shortcomings of regular hierarchical tessellations are similar to those of regular tessellations: the rotation and scaling operations are difficult to handle. The main advantage of using them—saving memory space—is present only when there is spatial coherence between cells having the same attribute value, ie when they are clustered together. Indeed, the size of a quadtree is not dependent on the number of cells, but on their distribution. The quadtree of a 2D grid having no two adjacent cells with the same value contains the same number of cells as the grid, and its size would most likely be worse because of the overhead to manage the tree.

5.3 Irregular Tessellations

The cells of an irregular tessellation can be of any shape and size, and they usually 'follow'—or are constrained by—the samples points that were collected to study the field, albeit this is not a requirement. Subdividing the space based on the samples has the main advantage of producing a tessellation that is *adaptive* to the distribution of the samples (and thus to the complexity of the phenomenon studied). The subdivision is potentially better than that obtained with quad/octree (which subdivides arbitrarily the space without any considerations for the samples).

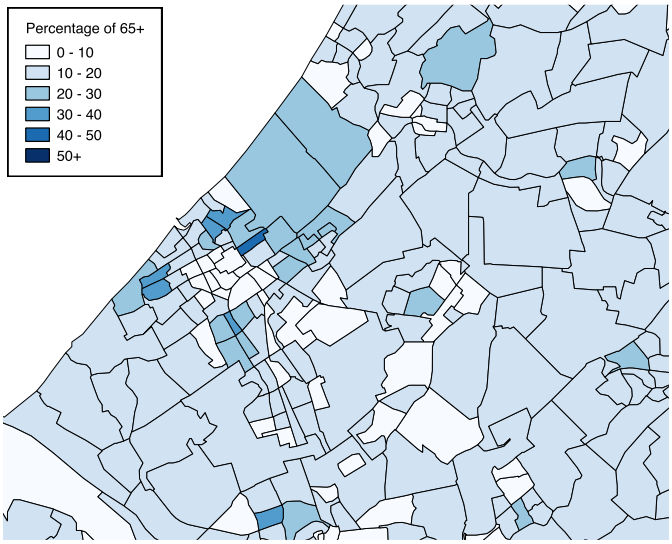

Figure 4: An example of a choropleth map (data from Statistics Netherlands).

The most known examples of the use of irregular tessellations in GIS for representing fields are arguably the choropleth map, the TIN and the Voronoi diagram.

5.3.1 Choropleth maps

A choropleth map subdivides the plane into arbitrary polygons, and a constant function is used within each polygon. They are mostly used to model and visualise discrete fields, such as densities, rates or proportions. Figure 4 shows one example where the percentage of people above 65 years of age per neighbourhood in a part of the Netherlands. Observe that since the field modelled is discrete, using other functions (linear or quadratic) would not make sense. Operations that are possible with continuous fields, eg interpolation, are also impossible.

The generalisation to higher dimensions of choropleth maps, ie of an irregular tessellation of space into cells of different shapes and size, is possible. In three dimensions, the cells are polyhedra, and these are used, among others, in geological applications where one type of rock is attached to each polyhedron. In four dimensions, the cells are polytopes, and can be conceptualised as polyhedra modelled over time.

The storage of a discrete field such as a choropleth is done with one of the data structures described in the chapter *Data structure, vector*, as this is simply a set of polygons to which one attribute value is attached. For the three- and four-dimensional cases, more advanced data structures have to be used. One example is the structure *generalised maps*, as first introduced in Lienhardt [1994]. It is capable of representing a wide class of objects in arbitrary dimensions. Practically, it also has the advantage of having been implemented in higher dimensions, and of being used in different systems in three dimensions (eg it is used in GOCAD[3] for geological modelling and in Moka[4] for geometric modelling).

5.3.2 Triangulated irregular networks (TINs)

As shown in Figure 5, a triangulated irregular network (TIN) refers to an irregular tessellation of a 2D field into non-overlapping triangles (whose vertices are formed by three sample points), and to the use of a linear interpolation function for each triangle. While in theory any 2D field can be modelled with a TIN, it is in practice mostly used for representing the elevation of an area because the lifting of each sample point to its elevation creates a surface, embedded in three dimensions, approximating the morphology of the terrain. The value of the attribute of a field at an unsampled location x is obtained by linearly interpolating on the plane passing through the

[3] http://www.gocad.org/
[4] http://moka-modeller.sourceforge.net/

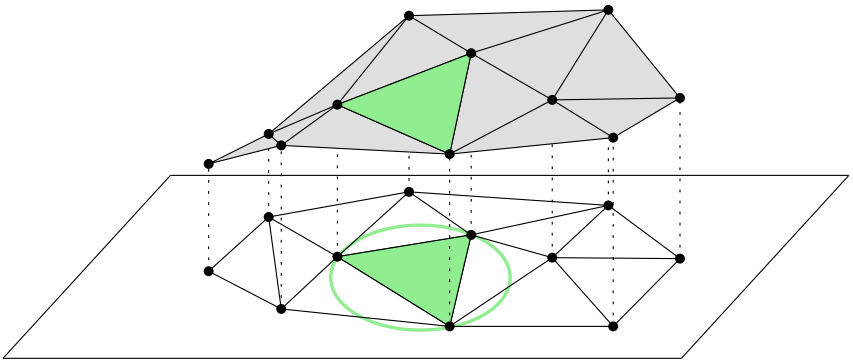

Figure 5: A TIN is obtained by lifting the triangles to their elevation. All the triangles are Delaunay, ie their circumcircle is empty of any other points in the plane. (The circumcircle for the green triangle is shown here.)

three vertices of the triangle containing x. TINs are the most popular alternatives to 2D grids for modelling elevation; both representations have advantages and disadvantages.

Given a set of points in the plane, there are several ways to construct a triangulation. A TIN usually refers—although this is not a strict requirement—to a Delaunay triangulation (DT) of the set S of sample points projected on the two-dimensional plane. A triangle respects the Delaunay criterion if the unique circle on which its three vertices lie does not contain any other point in the set S (see Figure 5). This also means that, among all the possible triangulations of a set S of points, the DT maximises the minimum angle (the *max-min* property) and minimises the maximum circumradii; in other words, it creates triangles that are as equilateral as possible. This property is useful in applications where interpolation is necessary (such as the reconstruction of a field) since with long and skinny triangles the estimation of the field attribute at location x by linear interpolation in the triangle will use samples that can be far away from x. In a DT the three vertices of the triangle containing x will most likely be close to, and around, x.

A TIN in which a linear interpolation function is used yields a C^0 piecewise representation, ie it is a continuous function but at the edges of the triangles the first derivative is not possible. Akima [1978] shows the advantages of using higher-order functions in each region of a TIN, to construct a C^1 or C^2 field. In other words, if elevation is the attribute modelled, the slope of the terrain will be 'good', which is important for several applications, for example flood modelling.

Storing a TIN can be done with any of the data structures described in the chapter *Data structure, vector* since a TIN is formed by a set of triangles. However, more efficient data structures, in terms of storage space and in terms of complexity, exist. The simplest of these, the triangle-based data structure, considers the triangle as being its atom and stores each triangle with three pointers to its vertices and three pointers to its adjacent triangle; it can be easily implemented in an array. Shewchuk [1997] reports the implementation of his program to compute the DT to be nearly twice as fast when this structure is used, instead of a generic structure for polygons. It is being used in several packages, notably by the open-source project CGAL[5]. It is also possible to see a TIN as a graph and to compress that graph, see for instance Blandford et al. [2005]. The data structure is more complex to construct, but can take five times less memory than the alternatives mentioned above. It should however be noticed that, from a practical point of view, if the triangulation used is Delaunay, it is easier to store *only* the sample points and reconstruct on the fly the DT when needed since the DT is unique for a given set of points (if no four or more points are cocircular). This is conceptually equivalent to the solution described in Section 4, and it is known that the DT of a set of points can be reconstructed very efficiently [Snoeyink and Kreveld, 1997].

The generalisation to three dimension of the Delaunay triangulation is the Delaunay tetrahedralization: each triangle becomes a tetrahedron that satisfies the *empty circumsphere* rule. However, obtaining nicely shaped tetrahedra that can be used for linear interpolation of a field is more difficult than finding good triangles because the max-min property of two-dimensional Delaunay triangles does not generalise to three dimensions (nor to higher dimensions). A three-dimensional DT can indeed contain some tetrahedra, called *slivers*, whose four vertices are al-

[5] http://www.cgal.org

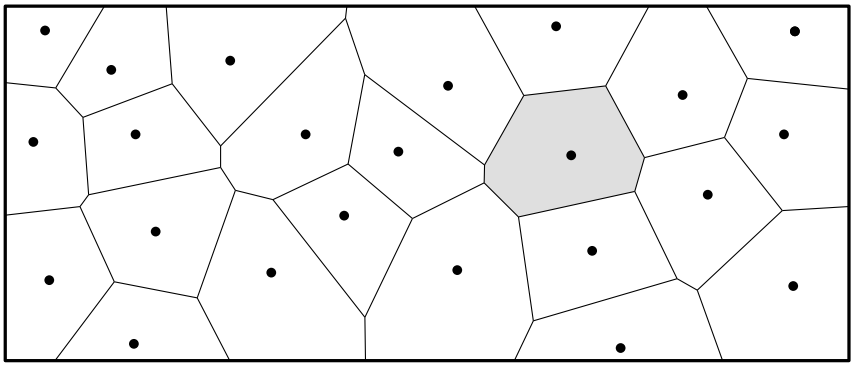

Figure 6: The two-dimensional Voronoi diagram. The cell in grey represents the space closer to the point in the middle of the grey cell than to any other point.

most coplanar, and thus have a volume of nearly zero. (Observe that slivers do not have two-dimensional counterparts.) For the reconstruction of a three-dimensional field with linear interpolation inside a tetrahedron, these tetrahedra are not optimal and should be removed. This is unfortunately a rather convoluted operation. However, it should be said that in most cases Delaunay tetrahedra have a more 'round' shape than arbitrary tetrahedra.

The DT generalises to four and higher-dimensions, the circumball around each *simplex* (the simplest element in a given dimension: vertex, edge, triangle, tetrahedron, etc) is empty of any other points. It can be constructed, among other ways, with the algorithm and the implementation described in Barber, Dobkin, and Huhdanpaa [1996].

Storing a 3D or 4D Delaunay triangulation is simpler than storing an higher-dimensional arbitrary subdivisions of space, and can be done with a straightforward generalisation of the triangle-based data structure: the atom is a d-dimensional simplex having $d + 1$ pointers to its vertices and $d + 1$ pointers to its neighbours. However, as is the case for the cases in 2D and 3D, the reconstruction on the fly is a simpler alternative in practice.

5.3.3 Voronoi diagram

Another example of an irregular tessellation is the Voronoi diagram (VD) of a set of points in a d-dimensional space, where each point is mapped in a one-to-one way to a cell representing the space closer to the point than to any other in the set. Figure 6 shows an example, and more details about the properties of the structure are discussed in the chapter *Data structures, Voronoi diagram*.

The VD is the geometric dual structure of the Delaunay triangulation (DT), and that in any dimensions. In brief, it means that both structures represent the same thing, but from a different viewpoint. The sample points collected to study a field are not on the boundary of the cells of a VD, but are rather the points generating each Voronoi cell (they are 'in the middle'). As is the case with the DT or other irregular tessellations, perhaps the main advantage of the VD for representing fields is that it permits us to preserve the original samples. The main advantage of the VD over the DT is that, for the three- and higher-dimensional cases, the shape of the cells is always 'good', ie even if slivers are present in the DT, the Voronoi cells will still be 'relatively spherical' (which is an advantage for interpolation).

If a constant function is assigned to each Voronoi cell, the VD permits us to elegantly represent discrete fields. Indeed, if sample points were collected to study a discrete field, it makes little sense to use spatial interpolation functions such as IDW; the VD offers a natural way to reconstruct the field. To know the value at a location x, one simply has to find the Voronoi cell containing x and query its attribute value. Observe that this is conceptually equivalent to storing the sample points and using a nearest neighbour function, as explained in Section 4.

To reconstruct a continuous field from a set of samples, more elaborate functions are needed since the VD creates discontinuities at the border of each cell. An interesting one is the natural neighbour interpolation method [Gold, 1989; Sibson, 1981] because it has been shown by different researchers to have many advantages over other methods when the distribution of samples is highly anisotropic: it behaves consistently whatever the distribution of the samples is, and it

is an automatic method that does not require user-defined parameters. The method is entirely based on the VD for both selecting the samples involved in the interpolation process, and to assign them an importance. The neighbours used in an interpolation are selected using the adjacency relationships of the VD, which results in the selection of neighbours that both surround and are close to x, the interpolation location. A field reconstructed with the natural neighbour function will be smooth and continuous everywhere (C^1), except at the samples themselves. It is possible to use that method in any dimensions, see Ledoux and Gold [2004].

Since the VD and the DT are dual structures, the knowledge of one implies the knowledge of the other. In other words, if one has only one structure, she can always extract the other one. Because it is easier, from an algorithmic and data structure point of view, to manage simplices over arbitrary polytopes (they have a constant number of vertices and neighbours), constructing and manipulating a VD by operating only on its dual structure is simpler and usually preferred. In the context of representing fields, there is therefore no need to store a VD. When the VD is needed, it is extracted on the fly from the DT. And since the DT is rarely stored in a computer as far as the modelling of fields is concerned, neither is the VD.

6 In brief

The choice of a spatial model to represent geographical phenomena is a fundamental issue of spatial data handling, and unfortunately one that is more than often overlooked by GIS practitioners. For a given task, practitioners tend to simply use the spatial models, and data structures, available in commercial GISs, without assessing the consequences of their choice. This is particularly problematic for the representation of fields as found in GIS-related disciplines because, regardless of a few notable models developed in academia, the only solutions available are grid-based models, and this is especially true in dimensions higher than two. The popularity of grids is probably due to their conceptual simplicity and to the fact that they are easily and naturally stored in computers. However, as it was demonstrated in this chapter, they have many shortcomings, both conceptually and technically. They tessellate arbitrarily the space without taking into consideration the objects embedded in that space (or the phenomenon being studied), which yields an unnatural discretisation. Moreover, grids need to be resampled each time the cartographic projection used is modified, and each resampling degrades the information stored in the file, and can lead to errors and misinterpretations during the analysis.

The alternative representations presented in this chapter are often more complex to construct and store in a computer, but practitioners should consider them because they would often permit them to *better* represent the phenomena they are studying. Irregular tessellations indeed permit us to obtain a subdivision that is based on the samples collected (which are preserved, by opposition to grids where they are 'lost'), and the cells constructed are adaptive to the spatial distribution of samples, which is crucial when dealing with highly anisotropic distributions such as the ones found in the geosciences.

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, p. 148–159.

Barber, C. B., D. P. Dobkin, and H. T. Huhdanpaa (1996). The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software* 22.4, p. 469–483.

Blandford, D. K., G. E. Blelloch, D. E. Cardoze, and C. Kadow (2005). Compact representations of simplicial meshes in two and three dimensions. *International Journal of Computational Geometry and Applications* 15.1, p. 3–24.

Dakowicz, M. and C. M. Gold (2003). Extracting meaningful slopes from terrain contours. *International Journal of Computational Geometry and Applications* 13.4, p. 339–357.

De Floriani, L. and P. Magillo (2002). Regular and irregular multi-resolution terrain models: A comparison. *Proceedings 10th ACM International Symposium on Advances in GIS*. McLean, Virginia, USA, p. 143–148.

Gold, C. M. (1989). Surface interpolation, spatial adjacency and GIS. *Three Dimensional Applications in Geographic Information Systems*. Edited by J. Raper. Taylor & Francis, p. 21–35.

Goodchild, M. F. (1992). Geographical data modeling. *Computers & Geosciences* 18.4, p. 401–408.
Kemp, K. K. (1993). *Environmental modeling with GIS: A strategy for dealing with spatial continuity*. Technical report 93-3. Santa Barbara, USA: National Center for Geographic Information and Analysis, University of California.
Kemp, K. K. and A. Včkovski (1998). Towards an ontology of fields. *Proceedings 3rd International Conference on GeoComputation*. Bristol, UK.
Ledoux, H. and C. M. Gold (2004). An efficient natural neighbour interpolation algorithm for geoscientific modelling. *Developments in Spatial Data Handling—11th International Symposium on Spatial Data Handling*. Edited by P. F. Fisher. Springer, p. 97–108.
Lienhardt, P. (1994). N-dimensional generalized combinatorial maps and cellular quasi-manifolds. *International Journal of Computational Geometry and Applications* 4.3, p. 275–324.
Mitasova, H. and L. Mitas (1993). Interpolation by regularized spline with tension: I. Theory and implementation. *Mathematical Geology* 25, p. 641–655.
Shewchuk, J. R. (1997). Delaunay Refinement Mesh Generation. PhD thesis. Pittsburg, USA: School of Computer Science, Carnegie Mellon University.
Sibson, R. (1981). A brief description of natural neighbour interpolation. *Interpreting Multivariate Data*. Edited by V. Barnett. New York, USA: Wiley, p. 21–36.
Snoeyink, J. and M. van Kreveld (1997). Linear-time reconstruction of Delaunay triangulations with applications. *Proceedings 5th Annual European Symposium on Algorithms*. Volume 1284. Lecture Notes in Computer Science. Graz, Austria, p. 459–471.

Further Reading

Couclelis, H. (1992). People manipulate objects (but cultivate fields): Beyond the raster-vector debate in GIS. *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*. Edited by A. U. Frank, I. Campari, and U. Formentini. Volume 639. Lecture Notes in Computer Science. Springer-Verlag, p. 65–77.
Couclelis, H. (1999). Space, time, geography. *Geographical Information Systems*. Edited by P. A. Longley, M. F. Goodchild, D. J. Maguire, and D. W. Rhind. Second. John Wiley & Sons, p. 29–38.
Fisher, P. F. (1997). The pixel: A snare and a delusion. *International Journal of Remote Sensing* 18.3, p. 679–685.
Kidner, D., M. Dorey, and D. Smith (1999). What's the point? Interpolation and extrapolation with a regular grid DEM. *Proceedings 4th International Conference on GeoComputation*. Mary Washington College Fredericksburg, Virginia, USA.
Kumler, M. P. (1994). An intensive comparison of triangulated irregular networks (TINs) and digital elevation models (DEMs). *Cartographica* 31.2.
Ledoux, H. (2008). FieldGML: An Alternative Representation for Fields. *Headway in Spatial Data Handling*. Edited by A. Ruas and C. M. Gold. Lecture Notes in Geoinformation and Cartography. Springer, p. 385–400.
Mitas, L. and H. Mitasova (1999). Spatial interpolation. *Geographical Information Systems*. Edited by P. A. Longley, M. F. Goodchild, D. J. Maguire, and D. W. Rhind. Second. John Wiley & Sons, p. 481–492.
Peuquet, D. J., B. Smith, and B. Brogaard (1999). *The ontology of fields: Report of a specialist meeting held under the auspices of the VARENIUS project*. Technical report. Santa Barbara, USA: National Center for Geographic Information and Analysis.
Samet, H. (1984). The quadtree and related hierarchical data structures. *ACM Computing Surveys* 16.2, p. 187–260.
Watson, D. F. (1992). *Contouring: A guide to the analysis and display of spatial data*. Oxford, UK: Pergamon Press.