USING SPLINE FUNCTIONS TO REPRESENT DISTRIBUTED ATTRIBUTES

John R. Herring
Intergraph Corporation
Huntsville, Alabama 35894-0001 USA
e-mail: ingr!b17a!mobius!jrh@uunet.uu.net

ABSTRACT

All current GIS systems assign discrete, static attribute values to geometric objects (vector, pixel, or voxel). This is not how the world usually works. Physical objects of geographic importance are heterogeneous things. The width, depth, and flow-rate of a river, the porosity, density, and permeability of a rock body, the pressure, temperature, and velocity of the air or water, all of these things vary in complicated, sometimes chaotic, and convoluted ways; ways that affect our experience and ways that would effect our computer models, if we took them into account, and knew how to deal with them. Given this fact-of-life, the next generation of GIS systems must have a mechanism to model truly continuously variable attribute values. Spline functions gives us one such a way.

Spline functions have long been used in CAD/CAM to represent geometric forms, curves and surfaces, a use that they are well qualified to perform in GIS applications (see for example Auerbach (1990)). But splines are a much more general concept than a convenient way to store geometry; they are a way to efficiently approximate, to any degree of accuracy, any function. By shifting our paradigm, we can make the dimensions of the splines simultaneously represent both geometry and attribute distributions.

INTRODUCTION AND BACKGROUND

Any information system must be able to model the reality of its application (Casti (1989)). A database designer begins with a methodology (for example, entity, attribute, relation modeling), that at an abstract level, uses a model of reality onto which he will impose his data concepts by a series of data transformations, eventually mapping the highest level abstractions by stages to a concrete storage mechanism (Date (1983), Ullman (1988) and (1989), Codd (1990)). This resultant storage mechanism unfortunately puts restrictions back upon the scope of the original abstract model, often restricting the attributes of a data item to the fundamental data types of integer, floating point, character string and variants thereof. In addition, the data types are usually considered independent of the methods needed to manipulate them, leaving the application the requirements to supply not only ingenious storage work arounds, but also the edit, analysis, query and mechanisms needed. This compounds the fundamental database management system problems of data integrity and semantic data control (see Özsu (1991)). With the advent of abstract data types (ADT), this is no longer the case (Gorlen (1990)). Using ADT's, the database designer can encapsulate a complex data storage format with the methods for its creation, manipulation, analysis, query, and display. This process is beginning to make its way into commercially available relational data bases (RDB) such as Empress, Oracle, Informix, and Ingres, and is the foundation of the new object oriented technology, such as Versant (VERSANT Object Technology), and Ontos (Ontologic) (see Khoshafian (1990)). This paper investigates the use of spline functions as an ADT for the storage of both space varying and time varying attributes.

Spline functions (see Farin (1990), Bartels (1987), Faux (1979)), often used in computer aided design and manufacture (CAD/CAM), are actually part of a very old branch of mathematics, approximation theory. Basically, splines allow us to approximate any function by the specification of a set of control points in the range of the function (called "poles", not necessarily function values) which control a varying weighted average based upon a set of functions (called "weight functions"). In CAD/CAM applications, the poles are 3-D points, and the weight functions map a compact subset of a Euclidean space (of dimension 1, 2, or 3) to the unit interval $[0,1] = \{ x \in R \mid 0 \leq x \leq 1\}$. The resulting range of the spline is a geometric object (contained in the convex hull of the poles). This object is either a curve, surface, or solid depending upon the dimension of the domain space; 1, 2 or 3 respectively.

More simply put, this paper proposes that we take the CAD/CAM interpretation of a spline, and extend the dimensions of the both the domain (source) and range (target) space; so that a CAD/CAM 3D point (x, y, z) becomes a GIS N-D point (x, y, z, time, density, porosity, permeability, ...). This approach basically generalizes the use of geometry to represent geography into the use of geometry to represent any measurable quantity; an old, well known and understood concept that most people encounter in a first algebra course.

The remainder of this paper develops the theory of splines to support this concept and gives examples of the type applications most likely to useful in GIS applications. Anyone wishing to learn about splines for their own sake is directed to the references, especially Farin (1990) which presents a more geometric development than most, and Auerbach (1990) which is a good example of the use of splines in geographic visualization. The development presented in this paper will emphasize some particular aspects of splines in ways peculiar to their use in supporting the spatial and temporal distribution of attributes.

FUNCTIONAL REPRESENTATIONS

A functional geometric description used in CAD/CAM is a generalization of the algorithmic construction objects used in vector data sets (line, polyline, polygon, circles, ellipse, general conics, etc.). Each functional geometric object consists of a domain (source or parameter space) and a function mapping the domain into the range coordinates system, usually E^3, 3-dimensional Euclidean space. Domains are usually some subset of a Euclidean space, the most common of which are I (the closed interval from 0 to 1, [0,1]) or some unit cube I^n (the cartesian product of I with itself, n times).

Spline functions are defined as a variable weighted average, using weight functions, over the domain of some specified set of points in the target coordinate space (poles) (see Farin (1990), Rogers (1990), Bartels (1987) or Faux (1979)). Formally, for a three-dimensional data set, we have a (interval) spline as a function:

$$f : I^n \rightarrow E^3$$
$$f(t) = \sum_{i=1}^{n} w_i(t)\, P_i$$

where t is a vector in I^n, P_i are points in E^3 and $w_i()$ are functions from I^n into I^1 such that:

$$0 \leq w_i(t) \leq 1 \text{ for every } t = (t_1, \ldots, t_n) \in I^n$$
$$\sum_{i=1}^{n} w_i(t) = 1$$

The affect of any one of the poles P_i is felt only where the associated weight function w_i is non-zero (called the support of w_i).

Geometrically speaking, the weight functions are usually bell-shaped curves with a single maximum point (the parameter value of which is usually a "knot" associated to the pole), tapering off to 0 in all directions away from this central peak. Because of this, the poles of a spline are often near critical points of the spline, often the value of the spline as evaluated at the knot. A spline passes through a particular pole, in general, only if the associated weight function is 1 at its knot value (which implies that all other weights are zero).

On I^n, the most common splines are based upon the n'th tensor product of weight functions for I. Given collections of weights, w_i and W_j we can define a collection of $w \otimes W$ by $(w \otimes W)_{i,j}(u,v) = w_i(u)\, W_j(v)$. This set of functions can be used as weights since I is closed under multiplication and

$$\sum_{i,j} (w \otimes W)_{i,j}(u,v) = \sum_i \sum_j w_i(u)\, W_j(v) = \sum_i w_i(u) \sum_j W_j(v) = 1 \times 1 = 1$$

Splines built using such tensor product weight functions are tensor splines. Most commercially available packages use exclusively tensor splines for higher dimensional functions due to their ease of computation, see Faux (1979).

Generalizations of these standard cubes can involve the choice of a different interval to support either computational convenience or added geometric interpretations of the parameter; for example it is often computational advantageous to use time or arc length for curves (discussed below), see Farin (1990). Unless otherwise stated, we will assume that the parameter cubes I, I^2, ... , I^n can be based upon any intervals in Euclidean space, as needed to support interpretations.

Some earlier nontensor higher dimensional spline work used triangles in place of cubes producing what is called a simplical spline (see Farin (1990)). This type of splines has generated some interest in the GIS applications, specifically in contour preserving surface visualization using a simplical decomposition or triangulated irregular network (TIN) based upon a constrained Delaunay triangulation, in Auerbach (1990).

A special case of the spline function is the B-spline. B-splines use piecewise polynomial or rational functions for weight functions. Each weight function's support spans an interval defined by a set of knots (the number of which is the order of the spline). This gives the spline designer a "local control" that allows him to adjust pole values while only affecting the spline is a very restricted neighborhood of the pole's knot. Further, given a set of sample points (t_i, v_i) $t_i \in I$, $v_i \in E^n$, $1 \le i \le m$, there are closed form solutions to finding m poles for which the associated spline exactly fit the samples, or for finding least-square "best fit" splines with a fewer number of poles (see Bartels (1987)). All of this discussion can be generalize to TIN's and to general simplical complexes (see Farin 1990).

Derivatives

It should be noted here that, while not always precisely spline functions themselves, the various derivative of a spline have easily calculable forms. Given a differential form D (i. e. something like $\frac{\partial}{\partial t_k}$, $1 \le k \le n$), the value of the form applied to a spline can be expressed as (a result of simple calculus):

$$D\, f(t) = \sum_{i=1}^{n} [\, D\, w_i\, (t)\,]\, P_i$$

In all cases, this is not a spline function as it is written (the sum of the derivative weights would necessarily be a constant 0, since $\Sigma\, D\, w = D\, (\Sigma\, w) = D\, (1) = 0$), but calculations of the various D f() is not significantly harder than the calculation of the spline values themselves. Further, for particular classes of splines, such as Bezier splines, there is collection of poles that will represent D f() as a spline function of a different degree (see Faux (1979)).

Curves

Curves can be represented as one-dimensional splines:

$$c : I^1 \rightarrow E^3$$

The continuity and differentiability of the curve are determined by the smoothness of the weight functions. The various derivatives of the curve as a function have exactly what you might expect. c'(t) is a tangent vector to the curve, with magnitude equal to the velocity of "t" with respect to arc length (thinking of t as a time component). c"(t) is acceleration, with a component parallel to the curve (parallel to c'(t)) giving the acceleration of t with respect to arc length, and the remaining component vector normal to the curve pointing directly away from the center of curvature. The component of c'"(t) perpendicular to the c'(t), c"(t) plane is a binormal indicating the direction of the torsion (twisting) of the curve (tendency of the curve to leave a planar surface) (see Rogers (1990)).

Using different parameterizations, gives some other interesting physical interpretations to c'() and c"(). If c() is parameterized by the arc-length (usually written as "s") of the resultant curve, then the c'(s) is the unit tangent and c"(s) is in the normal with the length of c"(s), written as $\|c(s)\|$, being the local Gaussian curvature of the curve c(t), the inverse of the radius of curvature. The acceleration vector, c"(t), will always line in the plane of c'(s) and c"(s), so that we could write

$$c''(t) = a\,c'(s) + b\,c''(s)$$

where "a" is the magnitude of the force of acceleration along the curve, and "b" is the magnitude of the force of acceleration due to change of direction (a sort of steering force).

Surfaces

Surfaces can be represented as two-dimensional tensor splines:

$$s : I^2 \rightarrow E^3$$

The images of the domain lines in the surface give a spline grid of constant parameter values. The partial derivatives of a surface spline give us the tangents to the surfaces in the direction of the associated parameter curves. Another interpretation of a surface tensor spline is as a parameterized set of curve splines. Assuming that we have a surface spline, s(t, u), we can define

$$\forall\, u \in I, \quad c_u(t) = s(t,u).$$

Applying our knowledge of curves, we know that

$$c_u'(t) = \frac{\partial}{\partial t} s(t, u)$$

And, swapping the roles of u and t, we also have

$$c_t'(u) = \frac{\partial}{\partial u} s(t, u)$$

Interpreting this in terms of the geometry, we can say that the partial derivatives of the surface spline are tangent vectors to curves totally contained within the surface. Assuming that the surface spline function is well behaved, the two tangent vectors give us a spanning set for the plane tangent to the surface at the corresponding point.

Simplical splines are closely related to triangulated irregular networks (TIN). Based upon a triangulated domain, the most common methodology would be to use the underlying geographic surface as the spline's domain. The use of a generalized full 3D triangulation would allow the surface to fold back over itself by allowing multiple s values for a single (x,y).

Volumes

Solids (volumes) can be represented in two distinct manners. The most common representation is as a collection of surfaces which form the boundary of the solid. In terms of distributions of attributes, this formulation would be useless, since it does not distribute the parameter space into the interior of the volume. In the second technique, generalizing from the above, we can always consider solids as representable as three-dimensional splines:

$$v : I^3 \rightarrow E^3$$

Such volume splines must usually be almost everywhere one-to-one to prevent the function from collapsing multiple points from the parameter space into single points in the range space (the mathematical equivalent of "spindle, fold and mutilate"). The most common exception to this is where the boundary of the parameter space is collapsed to give us non-rectilinear ranges.

If we apply the same technique to the parameters of a volume as we did above to the surface, we can view the function as a parameterized set of surfaces or as a 2-parameter set of curves. The embedded surfaces are called a "foliation" of the volume.

The generalization of the TIN based spline uses simplices of dimension 3 (tetrahedrons), see Herring (1990). For geographic use, the underlying tetrahedron irregular network would normally be a simplical complex spanning the volume of interest.

Higher Order Geometries

All of the above geometric descriptions can be extended to a 4th or higher dimension entity using the same techniques. An interesting hybrid is to use temporal spline curves P(t) to describe the motion of the poles of a spline through time. For the tensor splines, this is simply going to a spline of one higher dimension. For a simplical spline, this forms the tensor product of the existing spline with the temporal curve, as opposed to forming a 4D simplical complex. Even for simplical splines, tensoring with time curves is probably the preferable technique, since this most closely matches the way in which one thinks of time and motion. For most applications where time can be treated as an independent dimension, this technique should be applicable without much difficulty.

DISTRIBUTIONS

Distributions of attribution can be addressed by spline and other functional representations in two basic manners. The first technique includes the definition of the attributes with the geometry in a single spline function, The second technique uses multiple splines over a single parameter space. Other approaches can be viewed as combinations, and multiples of these first two.

In the first approach, given an attribute or a set of "k" attributes, each of which is expressible as a real value, and each of which is a continuous function of space, we can generate spline functions whose first three range coordinates (dimension of target geometry) represent points, and whose k trailing range coordinate values are for the attributes along the spline:

$$f : I^n \rightarrow E^3 \text{ (geometry)} \times E^k \text{ (attributes)}$$

This generalizes to temporal variability through the use of a space time geometric component, giving us:

$$f : I^n \rightarrow E^3 \text{ (geometry)} \times E^1 \text{ (time)} \times E^k \text{ (attributes)}$$

In the second approach, the attributes are generated by separate spline functions, but sharing a common parameter domain. Thus, we have a set of functions, f_0, f_1, etc.. The first functions gives us a mapping from the parameter space to the geometry, and each additional function generates a single attribute or a set of related attributes. The value of an attribute "a" at a point is then given by an implicit equation:

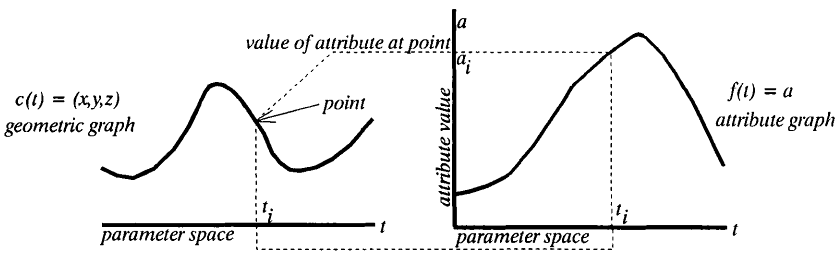
$$\forall\, t \in I^n \quad f\,(t) = \text{value of attribute a at the point } f_0(t)$$

In using splines or other functional geometric descriptions for the distribution of attribute values, we are creating a tensor sum model that makes no implementation distinction between geometry and other numerically measurable attributes (Herring (1990)).

Distributing Attributes Along a Line Feature

To distribute an attribute along a line feature, two pieces of information are needed. First, we need a parameterization of the line to use to associate spine values to positions on the line. Second, we need a set of sample values of the attribute along the line, or a mechanism to generate those values. Putting these pieces of information together, we now have a set of sample pairs consisting of the parameter values and attribute values:

$$S = \{(t_i, a_i) \;\; t_i \in I,\, a_i \text{ the attribute value at the point on the line associated to } t_i\}$$



The General Approach

We now have to choose a set of poles P_j and weights w_j:I→[0,1] ($1 \leq j \leq m$), that will generate a spline function

$$f = \sum_{j=1}^{m} w_j(t)\, P_j$$

such that:

$$\forall\, (t_i, a_i) \in S,\;\; f(t_i) \cong a$$

To associate a spline function to an existing line, we have to define how the parameter space is mapped to positions along the curve.

The spline case: If the line is already a spline (geometric) we can use this geometric spline's parameterization.

If our samples are at knots in the spline's parameter space, then we can augment the existing geometric poles with and additional dimension for the attribute, adjusting the pole-attribute values (a'_j), until the sample attribute values are achieved. This gives a combined geometric-attribute representation, as follows:

$$\forall\, t \in I, \quad f_0(t) = \sum_{i=1}^{m} w_i(t)\, (P_i, a'_i)$$

Assuming that we have n such geometrically correlated attributes, we have an extended spline function as follows:

$$\forall\, t \in I, \quad f_0(t) = \sum_{i=1}^{m} w_i(t)\, (P_i\,, a'_{1,i}, a'_{2,i}, \ldots, a'_{n,i})$$

Where P_i is the original geometric pole, and each $a'_{k,i}$ is an appropriately chosen value so that the k'th attribute value is achieved at the i'th knot.

If the attribute values are statistically independent of the shape of the geometry, or we do not have attribute values for the knots of the geometry spline, then the above method will not work. But using the same parameterization, we can define separate splines for each attribute or set of correlated attributes, using only the common parameter space to synchronize curve geometry and attribute distribution. Given a set of sample values (x_i,y_i,z_i,a_i), $1 \leq i \leq k$, of the attributes along the line feature, using the geometry spline $f_0()$, we solve for t_i such that:

$$f_0(f_i) = (x_i, y_i.z_i)$$

This gives us a set of spline functions samples (t_i, a_i), which we can use to generate a spline (using weights "W") that precisely fits the samples with k poles, or an approximation with fewer poles, giving us a spline $f_1()$ such that:

$$\forall\, t \in I,\; f_0(t) = \sum_{j=1}^{m} w_j(t)\, P_j$$

$$\text{and}\;\; f_1(t) = \sum_{i=1}^{n} W_i(t)\;\, a'_i$$

and the value of attribute "a" at $f_0(t)$ is $f_1(t)$.

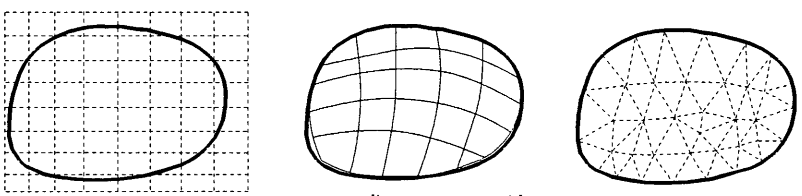
Multiple attributes can be handled either in single splines (as was done with the geometry spline above or as separate splines.

The Polyline and General Case: If the geometry of the line does not come with its own parameterization, then we can use any function, such as arc-length of a point from the line's beginning, as a distribution parameter. Using arc-length as a parameter defines the domain interval of the spline as $I = [0, L]$ where L is the total length of the line feature. Using this parameter, we are essentially in the second case from the section above. The new equation for the t_i is:

$$t_i = \text{distance along the line from start position to } (x_i, y_i, z_i)$$

Distributing Attributes Through an Area Feature, Across a Surface

The area, or surface distribution problem revolves around a restriction on the types of spline functions used for higher dimension. Most common software packages use tensor product splines. This places a restriction on the types of knot spacings that can be used. If k_i is the knot associated with $w_i()$ and k'_j is the knot associated to $w'_j()$, then the knot associated to $W_{i,j}() = w_i \otimes w'_j()$ is (k_i, k'_j). This means that the knots are geometrically dispersed in the parameter space in rows and columns (possibly nonuniformly spaced). There are three basic alternatives: 1) use a regular geometric grid as a spline parameter domain, 2) use a tensor spline geometric description of the area, or 3) use a simplical spline.



regular grid *tensor spline parameter grid* *simplicial spline poles (TIN)*

Pole Geometries for Alternative Distibution Types

If we chose to use a regular grid parameter space, we would create orthogonal profiles in a coordinate block large enough to encompass the entire area, associating the grid points internal to the area to interpolated attribute values, and external grid points to extrapolated attribute values. Here the basic problem is the "regularization" of the data to the grid points (discussed below).

In either of the other two alternatives, there are two problems. First, we have to disperse knots and geometric poles to describe the surface (x,y,z) or area feature (x,y). Second, we have to obtain attribute values for the points on the surface associated to each of the knot pole pairs. Having these, we can apply the algorithms described above to obtain attribute values for the poles that will give us the required distribution function.

Picking the Grid Points: If we already have a spline representation of the surface and the attribute values for the corresponding points on that spline surface, the simplest solution is to use the geometric knots (a direct analogy to the line cases).

If we do not have a spline surface and we wish to use a tensor spline, we can create a pseudo grid across the feature by digitizing two sets of profile lines, cross-hatching the area, using the intersections of these profiles to associate to a similarly set of orthogonal profiles in the chosen parameter space, creating a tensor spline surface that approximated the area feature. Using this method would not necessarily obtain a spline surface whose edge exactly matched the boundary of the boundary of the delineated area (splines can be made to fit a finite number of points, not usually an entire curve). The accuracy of the fitted surface would be a function of the complexity of the area boundary, and the order and number of poles of the chosen spline, but as long as the new surface covered the area feature, every point in the area would have an associated attribute value by the resultant spline distribution.

Alternately, if simplical splines can be used, a tessellation of the surface can be made using the Delaunay (or other) triangulation of the input attribute data samples.

Regularization of the data: In either or the grid techniques, it is probable that after getting a spline approximation of the area, the attribute values for the points on the spline surface will have to be approximated. Various such approximation techniques exist. Using the Delaunay triangulation of samples and either linear or "stolen area" interpolation (Gold (1989) and (1990)), simplical splines (Auerbach (1990)), kriging (Journel-78, David-76), and cokriging are good examples. The interpolation scheme may be chosen depending upon the particular application or depending on a priori assumptions about the data. Recall that the knots, or weights used for the geometric approximation need not be the ones used for the attribute approximation, as long as the parameter space is the same. In the simplical spline case, assuming the data points were chosen with care, little or no interpolation of Pole values should be necessary.

Distributing Attributes Through a Volume

The volume case is similar to the area case, except that a 3-dimensional approximating spline, a 3-dimensional regularization technique, or a 3-dimensional tetrahedron irregular network, as appropriated, are needed.

Vector Fields, Differential Equations and Trajectories

The use of splines to represent vector fields, and the ability to take derivatives of splines leads to their use to represent differential equations and systems of differential equations. For example, suppose that we have a spline representation of current flow in a hydrologic system. Thus, we have a function $F(u,v,w) \longrightarrow (x,y,z,dx,dy,dz)$ that maps a three dimensional parameter space into position and velocity. We can define a solution, or trajectory, to the differential equation:

$$(c,c') = F$$

as a function $c(t) \rightarrow (x,y,z)$ as one such that:

$$c(t) = \pi_{x,y,z} F(u,v,w) \Rightarrow c'(t) = \pi_{dx,dy,dz} F(u,v,w)$$

where "π" is the projection onto the subscripted coordinates.

A NOTE ON EXPERIMENTATION

Much of what is presented here can be classified as speculative, and in a normal situation, I would have waited for until more experimental results in specific applications could have been simultaneously reported. I choose not to delay for a variety of reasons. First, a great deal of work has gone into the various geometric aspects of spline curves and surfaces and, in a very real and meaningful way, this paper is simple a reinterpretation of those results. For example, Auerbach (1990) could be interpreted to show results on the distribution of a single attribute value over an area feature; its geometric representation (graph) resulting in a surface – contours representing isoclines. Secondly, a large part of this paper is a survey of some simple mathematical truths, viewed from an unusual perspective. Unlike physical science, most mathematical papers do not require experimental results to be valid. Third, and most important, is the potential scope of the applications of this sort of technology is broad enough to require multiple efforts to validate it. For example, the distribution of attributes along lines may solve the dynamic segmentation problem in road maintenance systems. The distribution of attributes in areas has applications in any field which needs to represent heterogeneous dispersions; forest or soil management, ecological applications such a predator prey simulations, etc.. Splines have the potential of solving some of the data volume problems associated to grid based map algebra systems. In 3 dimensions, spline distributions have a great deal of potential in representing heterogeneous aggregate both in geology and in engineered materials.

Given the potential of spline distributions and the track record of splines in the geometric applications, it seemed that the probability for successful experimentation in a wide variety of potential applications is very high.

SUMMARY AND IMPLICATION

Spline functions can be used to approximate a large variety of attribute distributions, through any standard geographic feature, to any accuracy or representation quality required. The implications of the methods outlined here are far reaching.

They can change the way we think of attribution. Attributes need not be thought of as static constants, but can be set to vary of both time and space. Attributes can include complex mathematical structures such as vector fields, set of trajectories for differential equations, etc.. Such attributes can be represented to any degree of accuracy required via the use of standard spline functions.

They can solve some long standing storage problems. Spline functions are known to be extremely efficient storage mechanisms, requiring as little as a tenth or a hundredth of the space as compared to vector representations of equal accuracy and quality of representation and visualization.

B-splines and NURBS (non-uniform rational b-splines), which are a standard in CAD applications and deliverable as standard software packages, meet the accuracy and representation requirements of these geographically and temporally distributed attributes. As a software engineering bonus, common geometric representations such as splines simplify system development.

Simplical splines solve some of the problems found in the standard tensor splines, and are a mechanism to visualize distributions from raw sample data. Theoretically, they should have many of the advantages of TIN based DTM's over grid representations.

ACKNOWLEDGMENT

I would like to thank Andrew Frank, Thomas Fisher, and Eric Bayer for their participation in the free form technical discussions over a period of year which lead directly to this paper. A good idea can have many fathers; a bad one is always an orphan.

REFERENCES

Auerbach, S. and Schaeerben (1990), "Surface Representations Reproducing Digitized Contour Lines", Mathematical Geology, Vol. 22, No. 6, pp 723-742.

Bartels, Richard H., John C. Beatty, Brian A. Barsky (1987), An Introduction to Splines for use in Computer Graphics and Geometric Modeling, Morgan Kaufmann Publishers, Inc., Los Altos, California

Casti, John L. (1989), Alternate Realities: Mathematical Models of Nature and Man, John Wiley & Sons; New York.

Codd, E. F. (1990) The Relational Model for Database Management: Version 2, Addison Wesley Publishing Company, Reading, Massachusetts.

Date, C. J. (1983), An Introduction to Database Systems, Vol. II (printed with corrections 1985), Addison Wesley Publishing Company, Reading Massachusetts.

Date, C. J. (1984), Relational Database: Selected Writings, Addison Wesley, Reading, Massachusetts.

Date, C. J. (1986), An Introduction to Database Systems, Vol. I, Fourth Edition, Addison Wesley Publishing Company, Reading Massachusetts.

David, M. (1976), "The Practice of Kriging", Advanced Geostatistics in the Mining Industry, M. Guarascio, et al. editors, D. Reidel Publishing Company, Dordrecht, Holland

Farin, G. (1990), Curves and Surfaces for Computer Aided Geometric Design, Second Edition, Academic Press, Boston, Massachusetts.

Faux, I. D., and Pratt M. J. (1979), Computational Geometry for Design and Manufacture, Ellis Horwood Limited, Chichester, UK.

Gold, Christopher M. (1989), Chapter 3 - Surface Interpolation, Spatial Adjacency and GIS, in Three Dimensional Applications in Geographic Information Systems, ed. J. Raper, Taylor and Francis Limited, London, pp. 21-35.

Gold, Christopher M. (1990), Spatial Statistics Based on Voroni Polygons, paper presented at the Atlantic Institute Seminar, August 1990, University of New Brunswick, Fredericton, New Brunswick, Canada, Y. C. Lee coordinator.

Gorlen, Keith E., Sanford M. Orlow, and Perry S. Plexico (1990), Data Abstraction and Object-Oriented Programming in C++, John Wiley and Sons, Inc., Chichester, UK.

Herring, John R. (1991), The Mathematical Modeling of Spatial and Non-Spatial Information in Geographic Information Systems, David Mark and Andrew Frank, ed., Cognitive and Linguistic Aspects of Geographic Space, Proceedings of a NATO Advanced Study Institute, (to appear).

Journel, A. and C. Huijbregts (1978), Mining Geostatistics, Academic Press, London, England.

Khoshafian, Setrag and Razmik Abnous (1990), Object Orientation: Concepts, Languages, Databases, User Interfaces, John Wiley and Sons, Inc., New York.

Munkres, James R. (1963), Elementary Differential Topology, Princeton University Press, Princeton, New Jersey.

Özsu, M. Tamer and Patrick Valduriez (1991), Principles of Distributed Database Systems, Prentice Hall, Englewood Cliffs, New Jersey.

Rogers, David F. and J. Alan Adams (1990), Mathematical Elements for Computer Graphics, Second Edition, McGraw Hill Publishing Company, New York.

Stonebreaker, M., ed. (1986), The INGRES Papers: Anatomy of a Relational Database System, Addison Wesley, Reading, Massachusetts.

Ullman, J. D. (1988), Principles of Database and Knowledge-Base Systems, Vol I, Computer Science Press, Rockville, Maryland.

Ullman, J. D. (1989), Principles of Database and Knowledge-Base Systems, Vol II, Computer Science Press, Rockville, Maryland.