

MINIREVIEW

Current Trends in Geometric Modeling and Selected Computational Applications

Gerald Farin* and Bernd Hamann†^{1,2}

**Department of Computer Science and Engineering, Arizona State University, Tempe, Arizona 85287-5406; and †Department of Computer Science, University of California, Davis, California 95616-8562*

E-mail: *farin@asu.edu and †hamann@cs.ucdavis.edu

Received July 15, 1996; revised November 11, 1996

Most numerical simulations require techniques for the representation and manipulation of complex, three-dimensional geometries. This paper provides a short historical survey and an overview of state-of-the-art geometric modeling techniques and research issues, and discusses a few selected applications of geometric modeling in computational areas. © 1997 Academic Press

1. INTRODUCTION

This survey paper provides an overview of recent developments in *geometric modeling*, also referred to as *computer-aided geometric design* (CAGD), and selected applications in computational areas. Before the field of geometric modeling had become a separate, independent field (around 1970), it was viewed as part of *computational geometry*, a field then concerned with all aspects of geometry representation in computers [19]. Today, computational geometry is no longer concerned with geometric modeling techniques (analytically defined shapes) but primarily with algorithms regarding finite sets of “simple” geometrical objects such as points, lines, polygons, and polyhedra [42].

Numerical simulations require efficient techniques for manipulating complex,

¹ To whom correspondence should be addressed.

² Co-Director of the Center for Image Processing and Integrated Computing (CIPIC), University of California.

three-dimensional (3D) geometries. Numerical simulations are applied to an ever-growing number of fields: flow around aircraft and cars, ocean currents on a global scale, the Earth's atmospheric circulation, electromagnetic fields, robot simulation, etc. Geometric modeling affects several aspects of the simulation of physical phenomena. Furthermore, geometric modeling is the theoretical foundation of computer-aided design (CAD) and solid modeling systems used, for example, for aircraft and car body design. Mesh generation methods heavily rely on geometric modeling techniques for the definition of 3D grids. Geometric modeling is also becoming increasingly important for the development of visualization algorithms and systems.

This paper summarizes current trends in geometric modeling, emphasizing its applications in computational sciences. Section 2 provides a historical perspective and discusses some current research areas, while Section 3 focuses on applications of geometric modeling.

2. GEOMETRIC MODELING—A HISTORICAL PERSPECTIVE AND CURRENT RESEARCH ISSUES

2.1. *A Brief History of the Field*

Where and when was the field of computer-aided curve and surface design conceived? It seems that the answer must point to places in two countries, France and the United States. In France, P. de Casteljau started work on a curve and surface design system in 1959 (for Citroen), and a short time later, P. Bézier started work on his UNISURF system (for Renault). Also, in the early 1960s, J. Ferguson at Boeing worked on the implementation of cubic splines into a design system, and S. Coons developed his surface schemes at MIT. The subject was formally named "Computer Aided Geometric Design," with a start toward a scientific development, at a conference in Utah in 1974 [2].

While this is not the place to discuss the details of those curve and surface schemes, it is worthwhile to point out where they differ in their philosophies and what they have in common.

De Casteljau's work was not recognized because it was kept confidential by Citroen for a long time. As was discovered in about 1972, the approaches of Bézier and de Casteljau are mathematically equivalent. R. Forrest discovered this when translating Bézier's book [3] into English. W. Boehm, in Germany, also became aware of de Casteljau's work and publicized it in 1978.

Both methods use tensor product polynomials (i.e., rectangular polynomial patches) to describe a surface. The polynomial degrees were arbitrary, yet had to be limited to about eight by eight in practice. For more complicated shapes, composite patches were used, with provisions for first- and second-order continuity between adjacent patches.

J. Ferguson [18] devised a surface scheme that fits a surface between a network of C^2 cubic spline curves, all having equidistant knot spacing. A "Ferguson surface" is thus an array of bicubic patches, joined together such that first-order continuity is maintained between adjacent patches. The spline curves from the initial network were of higher-order continuity than the resulting surface. This was not so out of

necessity: it was the result of a poor choice of “twist” vectors, i.e., the mixed second partial derivatives. Ferguson set the corner twists of each bicubic patch to be the zero vector.

Quite a different type of surface was developed by S. Coons [9]. He solved the following problem, often encountered in practice: given four boundary curves, find a rectangular patch that has those curves as its boundaries. The Coons approach is distinct from the previous ones in that it allows the boundary curves to be of *any* form: polynomial, trigonometric, etc. Thus, a Coons patch is far more general than the polynomial or piecewise polynomial patches described so far. Of course, if the input curves for a Coons patch are polynomials, then the Coons patch is itself a polynomial surface.

These Coons patches, also called “bilinearly blended patches,” cannot be pieced together in order to form a smooth overall surface. One has to use the “bicubically blended” Coons patch, which will guarantee a C^1 overall surface. However, care must be taken in the correct definition of the twist vectors that are a necessary input for this surface type. If these twists cannot be defined in a consistent way, one has to use methods such as “Gregory’s square” [26].

Most of the above schemes were state-of-the-art during the 1960s. The 1970s saw the advent of B-spline curves and surfaces, introduced into curve and surface design by W. Gordon and R. Riesenfeld [25]. The theoretical foundations were laid earlier by Mansfield, de Boor, and Cox [11]. From then on, Bézier methods could be viewed as a special case of the more general B-spline method. Every piecewise polynomial curve may be expressed as a B-spline curve. Thus, also local methods (necessary for local shape modifications) may be expressed in this general framework.

While B-splines had the most impact on applications, other research trends formed, such as triangular patches or subdivision surfaces.

2.2. Current Research Areas

The inception of the first curve and surface design methods is now some 35 years back in history. Many new methods have been developed since (and many of them perished). As a general theme in a significant part of this activity, one may detect the introduction of more geometric methods into curve and surface design.

The first research on this topic was performed by two people, P. Bézier and S. Coons. Subsequent research was typically carried out by mathematicians with a solid background in numerical analysis. Later, researchers with a more geometric background were attracted into the field, and this trend is continuing. The influence of geometry grew just as the quality of the available graphics media increased. While early computer output used to be printouts, we now witness real-time display of extremely complex objects. As we communicate with computers via computer graphics, the need for geometric intuition rises. The development of new technology clearly influences our theoretical approaches.

Let us now discuss some of the methods that are currently being investigated:

Geometric continuity. As is to be expected in any science, once methods are developed, people try to generalize them. One such generalization was the concept of “geometric” or “visual” continuity, generalizing the mathematical concepts of

first- and second-order continuity. Research in this field had started as early as the late 1960s and early 1970s (by J. Manning 1974 in England and G. M. Nielson 1974 in the U.S.), but it took until the early 1980s for these concepts to gain enough momentum to qualify as a major research area; see the survey article by Herron [30]. What is interesting in the context of this survey article is that we see a departure from purely algebraic methods (using calculus) and a turn towards methods that are more geometric (using differential geometry).

Algebraic geometry. Another development that emphasizes geometry is the use of methods from algebraic geometry, first undertaken by T. Sederberg. Many methods that were “in” in the last century had been more or less forgotten; the arrival of the new discipline “curve and surface design” caused a revival of many of them. T. Sederberg’s Ph.D. thesis [45] started this development; see also [31].

In the context of algebraic geometry, surfaces are described in implicit form, which has the significant advantage in the field of solid modeling that it is easy to tell whether a given point is inside or outside an object that is in implicit form. For parametric surfaces, this decision involves heavy computing.

Coons and Gordon surfaces. Bézier and B-spline methods can be classified as methods that deal with one patch only or with a network of patches at once. The same generalization was applied to Coons patches: in the late 1960s, W. Gordon at General Motors developed a method that can be viewed as a way to treat many Coons patches simultaneously.

Bézier- and Coons-type schemes are the prime candidates to highlight the dichotomy between geometry and algebra in the field of curve and surface design: while the Bézier form emphasizes geometric intuition, the main research on Coons/Gordon surfaces relied on abstract algebraic concepts such as Boolean sums (although Coons’ original work was in fact soundly based on geometric ideas). In view of our main theme—increasing emphasis on geometry, decreasing emphasis on algebra—the following comes as no surprise: up to the early 1970s, Bézier and Coons surfaces had been given about equal weight in the research literature. B-spline and Gordon surfaces generalized both schemes in a similar way. Yet Gordon surfaces never gained the popularity that B-splines did (keep in mind, however, that Gordon also played a major role in the development of B-spline techniques). Consequently, not much CAGD research is happening today in the arena of Coons and Gordon surfaces.

In the FEM field, Coons patches (and their trivariate analogues) are referred to as *transfinite methods*. If grid points are supplied along the boundary of an area, the Coons method is well suited to create more points inside.

NURBS. Bézier and B-spline geometry is (piecewise) polynomial and thus unable to represent conics or quadrics exactly. This dilemma was first encountered at Boeing and is now resolved by the introduction of rational curves and surfaces, commonly referred to as NURBS (for nonuniform rational B-splines). The initial research goes back to Coons and Forrest, leading to K. Vesprille’s Ph.D. thesis [50]. NURBS are a proper “superset” of Bézier and B-spline geometry: their definition needs a set of *weights*; if these are all set to unity, they reduce to the

“standard” geometry. Although not all research questions have been settled, NURBS may not be regarded as a matured field, having led to several IGES standards and two books [16, 41].

Triangular patches. In his early work, de Casteljaou considered patches with a triangular domain even before he considered rectangular patches. However, today’s CAD systems utilize surfaces that are almost completely of the rectangular type. The main reason for this fact can be traced back to the first applications of surface design methods, outer car panels and airplane fuselages. Those surfaces possess an intrinsic rectangular structure, and thus early systems were built around rectangular patches. Later, when more complicated parts had to be modeled, the limitations of rectangular patches (and rectangular topology) became apparent—but it was easier to modify existing methods than to integrate completely new schemes. The same statement holds true for other nonrectangular patch types, so-called n -sided patches. Work on modeling complex topological structures is quite extensive and will probably gain more importance in the future.

Triangular patches, in spite of their potential for modeling complex parts, have not found their way into a significant number of CAD/CAM systems. In the area of mesh generation, we might see a different trend: most polynomial and rational polynomial patches have a close relation to finite elements, most notably the Clough–Tocher element [7, 15]. If we model geometry using (CAGD versions of) finite elements, should this not facilitate subsequent analysis?

Automatic versus interactive methods. The generalizations of B-spline methods both toward the rational version thereof and toward geometrically continuous schemes have one characteristic in common: they provide more degrees of freedom than a designer can utilize. This sounds positive—more flexible methods are certainly to be preferred over more rigid ones. However, most of those methods need interactive user input in order to exploit their full potential. Given that machine time is steadily becoming cheaper, however, *automatic* methods are called for. Automatic methods would also free a designer from making nonintuitive (nongeometric) decisions such as assigning weights to rational B-spline curves.

Research is necessary to provide automatic methods for the “shape parameters” in geometrically continuous schemes and for the weights in rational B-spline schemes.

Geometry processing. The term *geometry processing* summarizes all the algorithms that are applied to already existing geometric entities. Examples are surface intersection, filleting (or rolling-ball blending), and offset algorithms [32]. Little research has been undertaken in this area, thus giving rise to a multitude of mostly ad hoc algorithms.

The development of a coherent theory for geometry processing algorithms may help establish the field of geometric modeling as a scientific discipline in its own right. For instance, the problem of finding the intersection curve of two parametric surfaces is equivalent to that of finding the solution of a nonlinear system of three equations in four unknowns. In principle, numerical analysis has many methods that should be able to solve this problem—however, they don’t seem to be very

efficient. The successful algorithms that have been developed tend to use the known geometry of the problem, not just its definition as a set of some number of equations in some number of unknowns (for a collection of relevant articles, see [1]. This should be viewed as another instance of the growing utility of geometry in this field.

3. SELECTED APPLICATIONS OF GEOMETRIC MODELING

This section focuses on selected computational fields that clearly show the utilization and value of geometric modeling in various computational applications. We have decided to select a few applications rather than simply provide an exhaustive list of references. Obviously, our choice is subjective and influenced by our own experience. We believe that our selection highlights some very interesting, important, and evolving uses of geometric modeling principles.

3.1. *Geometric Modeling Techniques for CAD Data Approximation and Grid Generation*

For most applications in CAD, solid modeling, grid generation, and data exchange between multiple modeling systems, it is imperative that a complex, 3D geometry is correct in the sense that it does not contain discontinuities, e.g., overlapping surfaces, “gaps” between surfaces, and surface intersections. Unfortunately, such continuity problems arise frequently in most industrial applications and require significant time to be corrected.

We describe an interactive technique that has been developed to correct CAD data with discontinuities. The described technique was developed for the National Grid Project, a project conducted at the NSF Engineering Research Center for Computational Field Simulation at Mississippi State University [43]. The technique is based on the approximation of a complex geometry, potentially consisting of thousands of patches with “undesired” discontinuities, by B-spline surfaces. A user of the technique interactively specifies the boundary curves of “large” B-spline surfaces which will approximate the given geometry. Eventually, a new, continuous approximation of the given geometry is obtained that will consist partially of given surface patches and partially of B-spline approximations covering possible surface discontinuities. This B-spline approximation can then be used for data exchange, grid generation, etc.

The approximation technique is based on the constructing of local surface approximations—bilinearly blended Coons patches—which are defined by specifying the four boundary curves. These Coons patches are then discretized by $N \times N$ points which are projected onto the given geometry. The projections are interpolated, yielding the local surface approximations, bicubic B-spline surfaces. Overall, the computation of a single B-spline surface approximation consists of these three major steps:

- (i) Projection of $N \times N$ points of a local Coons patch approximation onto the given surfaces.
- (ii) Estimation of point projections whenever certain points of the Coons patch cannot be projected onto the given surfaces:

When performing (i), a projection might or might not be found for a particular point on the Coons patch. If one or more projections are found on the given surfaces within a small distance to the Coons patch, the one closest to the Coons patch is selected. If no projection is found, a projection estimate is computed by applying a scattered data interpolation scheme to known projections that could be found in a close neighborhood.

(iii) Interpolation of the projections and estimated projections resulting from (i) to (ii) using bicubic B-spline surfaces.

The points $\mathbf{x}_{i,j}$, $i, j = 1, \dots, N$, on a Coons patch are projected onto the given surfaces in normal direction $\mathbf{n}_{i,j}$. Line segments $\mathbf{l}_{i,j}$ are constructed passing through $\mathbf{x}_{i,j}$ with direction $\mathbf{n}_{i,j}$. Their intersections with the given surfaces define the “discrete projection” of the Coons patch. Projections must be estimated for all line segments without intersection with the given surfaces. Each projection point $\mathbf{p}_{i,j}$ that can be found on the original surfaces is a linear combination of the end points $\mathbf{a}_{i,j}$ and $\mathbf{b}_{i,j}$ of $\mathbf{l}_{i,j}$, i.e., $p_{i,j} = (1 - t_{i,j})\mathbf{a}_{i,j} + t_{i,j}\mathbf{b}_{i,j}$. Thus, projection estimates for all line segments without intersection can be obtained by approximating t values using a scattered data interpolation method. Hardy’s reciprocal multiquadric method is used for this purpose [22]. The equation system to be solved is given by

$$t_{i,j} = \sum_{I,J \in \{1, \dots, N\}} c_{I,J} ((u_{I,J} - u_{i,j})^2 + (v_{I,J} - v_{i,j})^2 + R)^{-0.5}, \quad (1)$$

$$i, j \in \{1, \dots, N\},$$

where all $t_{i,j}$ values and parameter tuples $(u_{i,j}, v_{i,j})$ (associated with $\mathbf{x}_{i,j}$) are considered for which a projection can be found. The value of $R > 0$ is chosen according to the point distribution on the Coons patch.

An error estimate can be computed for each local B-spline approximation. Original curves, e.g., surface boundary curves or trimming curves, can be preserved by this method. This application is a great example where various geometric modeling techniques are utilized for the solution of a very important and very common computational problem. The approximation technique is described in detail in [28, 33]. The geometric modeling methods required for this technique can be found in [17, 32]. Figure 1 shows an example: the original CAD data of a car body geometry with “holes” (upper two images) and its continuous approximation (lower two images).

Grid generation is concerned with the discretization of 3D geometries and some finite space surrounding them—a necessary preprocessing step for all numerical simulations of complex physics phenomena. In the context of flow simulation around an aircraft, for example, one is concerned with the generation of surface meshes and volume meshes for all surface patches and all so-called *blocks*, which are used to represent the regions of interest around the aircraft [23, 37, 49]. *Transfinite interpolation* (TFI) is one of the most common and powerful methods used in this context. TFI has its origins in geometric modeling, where it was originally used for the construction of surfaces from a set of boundary curves [8]. TFI was later generalized for the construction of higher-dimensional manifolds, e.g., 3D, hexahedral solids defined in terms of their six boundary surfaces [24]. Most grid generation systems use TFI for the computation of initial meshes of surface patches and blocks.

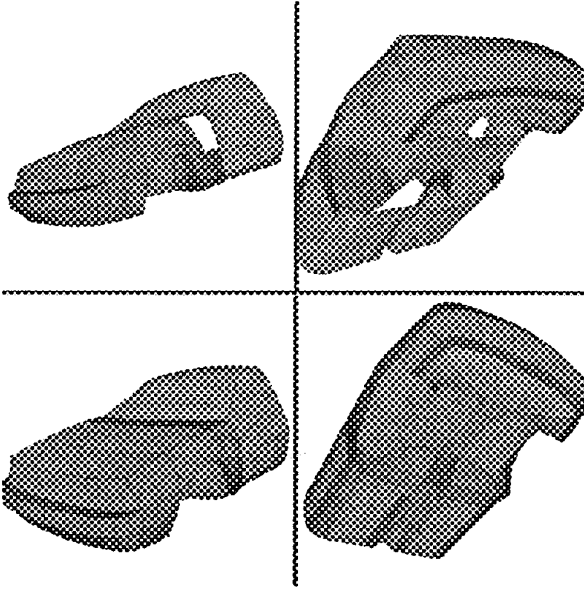


FIG. 1. Car body with “holes” (top) and its approximation (bottom) (courtesy of B. A. Jean and B. Hamann).

These initial meshes must usually be “smoothed” further for the grids to satisfy certain distribution and grid line smoothness conditions.

For pure geometric modeling applications, TFI is most often used for blending the boundary curves of a surface, thus defining the surface’s interior; e.g., considering the four parametric curves $\mathbf{c}_0(u)$, $\mathbf{c}_1(u)$, $\mathbf{d}_0(v)$, and $\mathbf{d}_1(v)$ ($\mathbf{c}_0(0) = \mathbf{d}_0(0)$, $\mathbf{c}_0(1) = \mathbf{d}_1(0)$, $\mathbf{c}_1(0) = \mathbf{d}_0(1)$, and $\mathbf{c}_1(1) = \mathbf{d}_1(1)$), defined for $u, v \in [0, 1]$, a four-sided surface can be defined by applying linear blending functions to the boundary curves (Farin, 1997). The resulting surface is

$$\begin{aligned} \mathbf{s}(u, v) = & (1 - u)\mathbf{d}_0(v) + u\mathbf{d}_1(v) + (1 - v)\mathbf{c}_0(u) + v\mathbf{c}_1(u) \\ & - ((1 - u)((1 - v)\mathbf{c}_0(0) + v\mathbf{c}_1(0)) \\ & + u((1 - v)\mathbf{c}_0(1) + v\mathbf{c}_1(1))). \end{aligned} \quad (2)$$

Higher-order blending functions must be used when concerned with derivative continuity across surface patch boundaries. The TFI paradigm is used to define and discretize the interior of blocks from their boundary faces. TFI is a prominent example for a geometric modeling technique that has directly affected the computational sciences, in this case numerical grid generation.

3.2. Alpha Shapes—A Promising Development in Computational Geometry

Recent developments in computational geometry indicate that various methods from this discipline of “discrete geometry” will have strong implications for the computational sciences. One interesting concept is the use of *alpha shapes* for the

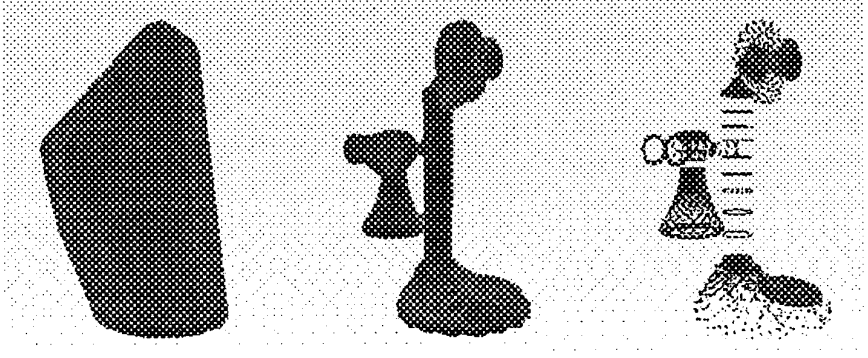


FIG. 2. Various alpha shapes of a digitized telephone; alpha decreasing from top to bottom and left to right (courtesy of H. Edelsbrunner and E. P. Mücke).

approximation of shapes in 3D (or even higher dimensions) derived purely from finite sets of scattered, unorganized points [14]. Unstructured, tetrahedral grid generation methods have, in the past, used techniques from computational geometry quite heavily, including the *Delaunay triangulation* [54]. Alpha shapes have the potential to particularly impact next-generation, automatic grid generation techniques.

Alpha shapes can be viewed as a generalization of the Delaunay triangulation. Ignoring certain degenerate cases, the Delaunay triangulation of a point set in three dimensions is characterized by the fact that the sphere passing through the four vertices of any tetrahedron does not contain any other point but the four vertices. The Delaunay triangulation defines a “complex” of edges, triangles, and tetrahedra. Given a specific alpha value, an edge in this “complex” belongs to the alpha shape if the radius of the smallest sphere passing through the edge’s end points is smaller than alpha. Similarly, a triangle (tetrahedron) in the “complex” belongs to the alpha shape if the radius of the smallest sphere passing through the triangle’s (tetrahedron’s) vertices is smaller than alpha. The Delaunay triangulation itself has an associated alpha value of infinity and gradually decreasing the alpha value toward zero leads to structures consisting of increasingly “isolated subcomplexes,” e.g., strings of edges, chains of triangles, groups of connected tetrahedra, and isolated points. Figure 2 shows a sequence of alpha shapes of a digitized telephone.

Emerging applications for alpha shapes are automatic mesh generation, cluster identification in 3D (or higher-dimensional) point sets, modeling complex molecular structures, and understanding the distribution of galaxies.

3.3. Multiresolution Methods for Computer Graphics and Visualization

Scientific visualization is the area in computer graphics dealing with the rendering of scientific data, e.g., medical data, fluid flow data, and material properties. The data sets to be studied are extremely large, often consisting of several million data points. Scientific data sets are generally not visualized directly, but one extracts meaningful, simple geometrical data sets (e.g., isosurfaces) that are rendered using

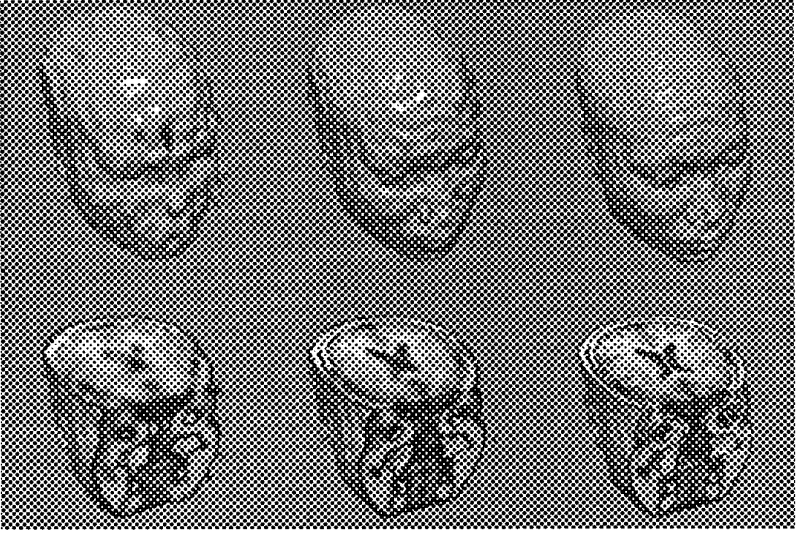


FIG. 3. Rendering MRI data set using 2,048, 8,192, and 32,768 wavelet basis functions (courtesy of S. Muraki).

standard computer graphics techniques. Essential visualization techniques and important research issues in scientific visualization are discussed in [27, 36, 40, 44].

Hierarchical geometric modeling techniques have been used for quite some time for the representation of curves and surfaces at various levels of detail [20, 21]. Recently, various *wavelet* and *multiresolution methods* have been developed that seem to be very promising in the context of compressing very large 3D scientific data sets as well as extracted 3D geometrical data [5, 10]. We will briefly outline two methods: a wavelet-based technique for the compression of volumetric data sets and a multiresolution method for very large surface triangulations.

First, let us consider a univariate, piecewise constant function $f(x) = f_i \Leftrightarrow x \in [i/2^N, (i+1)/2^N)$, $i = 0, 1, 2, \dots, 2^N - 1$, defined over the interval $[0, 1)$. This function can, on the coarsest level, be approximated by the single constant function $a(x) = 1/2^N \sum_{i=0}^{2^N-1} f_i$, which is the average of $f(x)$. One can now compute the error function $e(x) = f(x) - a(x)$ and approximate $e(x)$ in the same way, by two constant functions, one over the interval $[0, 1/2)$ and the other one over $[1/2, 1)$. This process is applied recursively, and it is terminated when all error functions are zero. This is the fundamental idea of all *dyadic multiresolution methods*. An overview of these methods, from a computer graphics point of view, is given in [47, 48].

Little work has been done in the area of applying the multiresolution paradigm to 3D, volumetric data sets—which is the most obvious application in the context of scientific visualization. The method described in [39] compresses 3D, volumetric scientific data given on rectilinear grids and applies the univariate wavelet paradigm to the 3D tensor product case. Few coefficients are required to describe regions characterized by constant or linearly varying phenomena, whereas many coefficients are necessary to capture detail in regions characterized by rapidly changing function values. This method has been successfully applied to volumetric data. Figure 3

shows an MRI (magnetic resonance imaging) data set rendered using an increasing number of wavelet functions (top, extracted isosurfaces; bottom, Gouraud-shaded cross sections).

Unfortunately, this approach does not apply to data on tetrahedral grids. Research in this direction—multi-resolution methods for tetrahedral grids—will most likely become an important research area. Some initial steps have been done in this regard by developing mesh decimation strategies for tetrahedral grids [29] and progressively refined volume triangulations [6].

Another related issue is the application of compression techniques to very large surface triangulations, which arise in nearly all surface-based visualization applications. Examples include the compression of multiple isosurfaces and the compression of multiple stream surfaces extracted from very large scalar or fluid flow data sets and represented by triangular meshes. Research has recently been done in the area of multiresolution methods for very general surface triangulations. A multiresolution method for a surface triangulation that is homeomorphic to a sphere is discussed in [38]. Related work and extensions are presented in [12, 13].

The underlying two basic principles of these methods are (i) the construction of a subdivision scheme for some initial, coarse surface triangulation (*base mesh*), e.g., splitting each triangle into four subtriangles, and (ii) the definition of wavelet(-like) basis functions for the various levels in the hierarchical mesh representation. A simple approach for modeling a surface homeomorphic to a sphere, for example, would be to initially approximate it by a polyhedron homeomorphic to an octahedron and recursively subdivide each triangular face into four subtriangles which more closely approximate the given shape. The computation of the coordinates of the new vertices resulting from subdivision is essentially determined by numerically approximating integrals of local wavelet functions and the difference between the given surface and its approximation by the previous-level triangular mesh.

The method in [38] considers only triangular meshes with *subdivision connectivity*, i.e., meshes that are obtained by recursively splitting a triangle into four subtriangles by splitting the edges (*1-to-4 splits*). In practice, one must construct hierarchical representations for very large triangulations that cannot directly be viewed as the result of some recursive subdivision process using 1-to-4 splits. The method presented in [13] “groups” triangles in a certain neighborhood of an arbitrary, large triangulation to form *tiles* consisting of certain triangles. These tiles define a Delaunay-like tessellation of the given set of triangles. The dual of this tessellation, some coarse triangulation obtained by connecting “center vertices” in neighboring tiles, is homeomorphic to the given, large triangulation and is used as the input to Lounsbery’s original algorithm. Lounsbery’s algorithm then produces finer triangulation levels approximating the given triangulation better and better. Figure 4 shows a geometry with three holes. The tiling of the given triangulation, the Delaunay-like triangulation (coarse base mesh), and two finer refinement levels are shown (from left to right).

With scientific data sets increasing rapidly in size, there is a need to investigate multiresolution strategies for 3D, time-varying data sets for all kinds of underlying grid topologies, including tetrahedral, hexahedral, and hybrid grids. For multiresolution visualization to become really powerful it is essential that efficient rendering



FIG. 4. Tiles on initial triangulation, base mesh, and two refined meshes (courtesy of M. Eck, A. D. DeRose, T. Duchamp, H. Hoppe, J. M. Lounsbery, and W. Stuetzle).

paradigms can be invented for very general volumetric multiresolution schemes considering all possible grid types, e.g., hierarchical “tetrahedrizations”; see the discussion of research issues in [44].

3.4. Motion Planning—Geometric Modeling for Robotics Applications

Another application of geometric modeling is motion design in the context of robot kinematics. Other areas that will most likely benefit from new motion design methods include animation (computer graphics), scientific visualization, and the animation of mechanical systems. We outline a path planning algorithm based on rational splines for robot motion control as described in [34, 35, 51, 52]. Rational motions are motions with only rational point trajectories. Mathematically, a rational motion can be described by a 4×4 matrix whose elements are polynomials. By converting the matrix representation to B-spline representation one obtains a rational motion. It is completely defined by the set of constant coefficient matrices, which can be visualized by so-called *affine control positions* of the moving object.

The main advantage of this approach is that each point trajectory can easily be described as a NURBS curve once the “control matrices” are known. The control points are the control positions of the moving point. More generally, any NURBS curve sweeps out a tensor product NURBS surface whose control point net is formed by the control positions of all control points of the moving curve. This property has previously been applied to the construction of sweep surfaces. This technique can also be used for the construction of certain motions which interpolate a given point trajectory such as a NURBS curve. One can also force the moving object to remain fixed with respect to the *Frenet–Serret motion* of the given curve [53].

NURBS curves allow the specification of weights for each control position. If all weights are positive, the motion satisfies the convex hull property with respect to its control positions. In other words, the area that is traced out by a moving object will lie inside the convex hull of its control positions. Therefore, when subdividing the motion into two or more segments, the union of the convex hulls has to converge

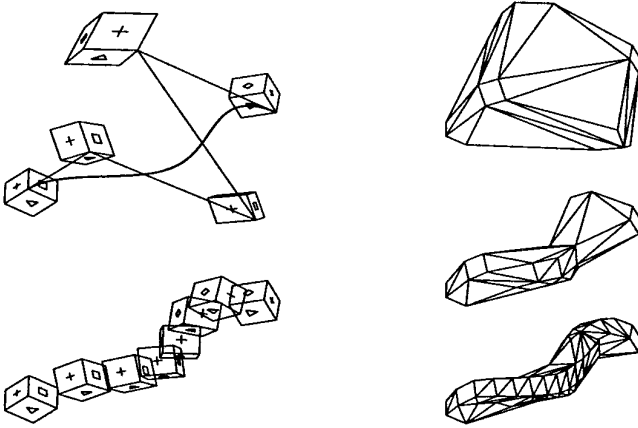


FIG. 5. Control positions, motion, and traced area using different refinement levels (courtesy of M. G. Wagner).

to the traced area. This property is very helpful for collision detection. If not all weights are positive, one can exchange the “traditional” convex hull property by its projective analog [35, 51]. Figure 5 shows a rational motion.

4. CONCLUSIONS

We have pointed out some recent, significant developments in geometric modeling and a few, selected applications of geometric modeling in computational sciences. May this survey help to foster the interdisciplinary nature of geometric modeling and increase the dialogue between computational scientists and experts in geometric modeling.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under Contract ASC-9210439 (Research Initiation Award) to the University of California, Davis, and Contract DM9123527 to Arizona State University. We are very much obliged to Matthias Eck, Herbert Edelsbrunner, Tony DeRose, Tom Duchamp, Hugues Hoppe, Brian Jean, Michael Lounsbery, Ernst Peter Mücke, Shigeru Muraki, Werner Stuetzle, and Michael Wagner for giving us permission to use some of their figures from [13, 14, 33, 39, 51].

REFERENCES

1. R. E. Barnhill (Ed.) *Geometry Processing for Design and Manufacturing* (SIAM, Philadelphia, 1992).
2. R. E. Barnhill and R. Riesenfeld, *Computer Aided Geometric Design* (Academic Press, Cambridge, MA, 1974).
3. P. Bézier, *Numerical Control: Mathematics and Applications* (Wiley, New York, 1972).
4. W. Boehm and H. Prautzsch, *Geometric Concepts for Geometric Design* (Peters, Wellesley, MA, 1994).
5. C. K. Chui, *An Introduction to Wavelets* (Academic Press, San Diego, CA, 1992).

6. P. Cignoni, L. De Floriani, C. Montani, E. Puppo, and R. Scopigno, Multiresolution modeling and visualization of volume data based on simplicial complexes, in *1994 Symposium on Volume Visualization*, edited by A. E. Kaufman and W. Krüger (IEEE Computer Society Press, Los Alamitos, CA, 1994), p. 19.
7. R. Clough and J. Tocher, Finite element stiffness matrices for analysis of plates in blending, in *Proc. Conference on Matrix Methods in Structural Analysis, Wright-Patterson AFB, OH*.
8. S. A. Coons, Surface patches and B-spline curves, in *Computer Aided Geometric Design*, edited by R. E. Barnhill and R. F. Riesenfeld (Academic Press, New York, 1974), p. 1.
9. S. A. Coons, Technical report, MIT, 1964, available as AD 663 504 from the National Technical Information Service, Springfield, VA 22161.
10. I. Daubechies, *Wavelets*, CBMS-NSF Series in Applied Mathematics (SIAM, Philadelphia, 1992).
11. C. de Boor, *A Practical Guide to Splines* (Springer-Verlag, New York, 1978).
12. A. D. DeRose, J. M. Lounsbery, and J. Warren, Technical Report 93-10-05b, Department of Computer Science and Engineering, University of Washington, Seattle, 1993.
13. M. Eck, A. D. DeRose, T. Duchamp, H. Hoppe, J. M. Lounsbery, and W. Stuetzle, Multiresolution analysis of arbitrary meshes, in *Proceedings of SIGGRAPH 1995* (ACM, New York, 1995), p. 173.
14. H. Edelsbrunner and E. P. Mücke, Three-dimensional alpha shapes, *ACM Trans. Graphics* **13**(1), (1994).
15. G. Farin, Triangular Bernstein-Bézier patches, *Comput. Aided Geom. Design* **3**(2), (1986).
16. G. Farin, *NURB Curves and Surfaces* (Peters, Wellesley, MA, 1995).
17. G. Farin, *Curves and Surfaces for Computer Aided Geometric Design* (Academic Press, Boston, 1997), 4th ed.
18. J. Ferguson, Multivariable curve interpolation, *J. Assoc. Comput. Mach.* **11**(2), 221 (1964).
19. A. R. Forrest, Computational geometry—Achievements and problems, in *Computer Aided Geometric Design*, edited by R. E. Barnhill and R. F. Riesenfeld (Academic Press, New York, 1974), p. 17.
20. D. R. Forsey and R. H. Bartels, Surface fitting with hierarchical splines, *ACM Trans. Graphics* **14**(2), (1995).
21. D. R. Forsey and R. H. Bartels, Hierarchical B-spline refinement, in *Proceedings of SIGGRAPH 1988* (ACM, New York, 1988), p. 205.
22. R. Franke, Scattered data interpolation: Tests of some methods, *Math. Comp.* **38**, (1982).
23. P. L. George, *Automatic Mesh Generation* (Wiley, New York, 1991).
24. W. J. Gordon, Blending-function methods of bivariate and multivariate interpolation and approximation, *SIAM J. Numer. Anal.* **8**(1), 158 (1971).
25. W. J. Gordon and R. Riesenfeld, B-spline curves and surfaces, in *Computer Aided Geometric Design*, edited by R. E. Barnhill and R. F. Riesenfeld (Academic Press, New York, 1974), p. 95.
26. J. A. Gregory, Smooth interpolation without twist constraints, in *Computer Aided Geometric Design*, edited by R. E. Barnhill and R. F. Riesenfeld (Academic Press, New York, 1974), p. 71.
27. H. Hagen, H. Müller, and G. M. Nielson (Eds.) *Focus on Scientific Visualization* (Springer-Verlag, New York, 1993).
28. B. Hamann, Construction of B-spline approximations for use in numerical grid generation, *Appl. Math. Comput.* **65**(1&2), 295 (1994).
29. B. Hamann and J. L. Chen, Data point selection for piecewise trilinear approximation, *Comput. Aided Geom. Design* **11**(5), 477 (1994).
30. G. Herron, Techniques for visual continuity, in *Geometric Modeling*, edited by G. Farin (SIAM, Philadelphia, 1978), p. 163.
31. C. M. Hoffmann, *Geometric & Solid Modeling* (Morgan Kaufmann, San Mateo, CA, 1989).
32. J. Hoschek and D. Lasser, *Fundamentals of Computer Aided Geometric Design* (AK Peters, Wellesley, MA, 1993).

33. B. A. Jean and B. Hamann, Interactive techniques for correcting CAD/CAM data, in *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, edited by N. P. Weatherill, P. R. Eiseman, J. Häuser, and J. F. Thompson (Pineridge Press, Swansea, U.K., 1994), p. 317.
34. B. Jüttler, Spatial rational motions and their application in computer aided geometric design, in *Mathematical Methods for Curves and Surfaces*, edited by M. Daehlen, T. Lyche, and L. L. Schumaker (Vanderbilt University Press, La Vergne, TN, 1995).
35. B. Jüttler and M. G. Wagner, Computer aided design of spatial rational B-spline motions, *ASME J. Mechanical Design* (in press).
36. A. E. Kaufman (Ed.) *Volume Visualization*, tutorial (IEEE Computer Society Press, Los Alamitos, CA, 1990).
37. P. M. Knupp and S. Steinberg, *Fundamentals of Grid Generation* (CRC Press, Boca Raton, FL, 1993).
38. J. M. Lounsbury, Multiresolution analysis for surfaces of arbitrary topological shape, dissertation, Department of Computer Science and Engineering, University of Washington, Seattle, WA, 1994.
39. S. Muraki, Volume data and wavelet transforms, *IEEE Comput. Graphics Appl.* **13**(4), 50 (1993).
40. G. M. Nielson, B. D. Shriver, and L. J. Rosenblum, (Eds.) *Visualization in Scientific Computing*, tutorial (IEEE Computer Society Press, Los Alamitos, CA, 1990).
41. L. Piegl and W. Tiller, *The NURBS Book* (Springer-Verlag, New York, 1995).
42. F. P. Preparata and M. I. Shamos, *Computational Geometry* (Springer-Verlag, New York, 1990).
43. M. G. Remotigue, The National Grid Project: Making dreams into reality, in *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, edited by N. P. Weatherill, P. R. Eiseman, J. Häuser, and J. F. Thompson (Pineridge Press, Swansea, U.K., 1994), p. 429.
44. L. J. Rosenblum, R. A. Earnshaw, J. L. Encarnação, H. Hagen, A. E. Kaufman, S. Klimenko, G. M. Nielson, F. Post, and D. Thalmann, (Eds.) *Scientific Visualization—Advances and Challenges* (IEEE Computer Society Press, Los Alamitos, CA, 1994).
45. T. Sederberg, Ph.D. dissertation, Department of Mechanical Engineering, Purdue University, 1983.
46. H.-P. Seidel, Symmetric recursive algorithms for surfaces: B-patches and the de Boor algorithm for polynomials over triangles, *Constr. Approx.* **7**, 257 (1991).
47. E. J. Stollnitz, A. D. DeRose, D. H. Salesin, Wavelets for computer graphics: A primer, 1, *IEEE Comput. Graphics Appl.* **15**(3), (1995).
48. E. J. Stollnitz, A. D. DeRose, and D. H. Salesin, Wavelets for computer graphics: A primer, 2, *IEEE Comput. Graphics Appl.* **15**(4), (1995).
49. J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, *Numerical Grid Generation* (North-Holland, New York, 1985).
50. K. Vesprille, Ph.D. dissertation, Syracuse University, 1975.
51. M. G. Wagner, Planar rational B-spline motions, *Comput. Aided Design* **27**(2), 129 (1995).
52. M. G. Wagner and H. Pottmann, Geometric motion design, in *Modelling and Planning for Sensor Based Intelligent Robot Systems*, edited by H. Bunke, T. Kanade, and H. Noltemeier (World Scientific, Singapore, 1995), p. 104.
53. M. G. Wagner and B. Ravani, Computer aided design of robot trajectories using rational Frenet-Serret motions, Department of Mechanical and Aeronautical Engineering, University of California, Davis, preprint, 1996.
54. N. P. Weatherill, O. Hassan, D. L. Marcum, and M. J. Marchant, Grid generation by the Delaunay triangulation, von Karman Institute for Fluid Dynamics, 1993–1994 Lecture Series, January, 1994.