CAGD's Top Ten: What to Watch

While painless for the guest editor, these introductions might cause readers some pain. Still, I don’t want to waste this opportunity to tell you about interesting developments in computer-aided geometric design by giving you reason to hastily skim past these opening paragraphs. So, I abstracted a “top ten” list of what’s happening in CAGD based on ideas from recent conferences, technical papers, and informal discussions with my colleagues. I do not mean the list to be definitive. Some items might overlap, and the length and level of detail vary considerably. I want to paint a panorama of CAGD as a backdrop for the detailed foreground highlights represented by the collection of articles in the two special CAGD issues of IEEE CG&A, September 1992 and January 1993.

This issue of CG&A contains the concluding part of the CAGD collection. The September 1992 issue offered articles on solid modeling, feature-based design, parametric surface interpolation, surface interrogation algorithms, nonuniform rational B-splines, interval Bezier curves, and Voronoi diagrams. This issue finishes with works on polar forms, cyclides, triangulations, scattered data modeling, surface-to-surface intersections, incremental boundary evaluation, and implicit curves and surfaces. I think the breadth and depth of topics provides excellent coverage of CAGD.

So, what’s happening in CAGD right now that bears watching?

1. 3D object representation from scattered data

Consider a cloud of empirical 3D data points provided by a scanner. Now think about the subsequent task of inferring the geometry and topology needed to obtain a mathematical representation of the surface(s) implied by this data. This problem proves difficult partly because the data contains no implicit neighborhood information and partly because it is subject to measurement error. Even if we inferred topological information or somehow got it in the form of a triangulation of a reduced set of points, the resulting parametric scattered data interpolation problem still challenges researchers.

The first techniques to address these problems appeared recently. However, we need to make them more robust and extend them to generate mathematical representations consistent with CAD/CAM systems now in use. The immense value of these algorithms for processing scanned data and 3D data reconstructed from the registration of multiple camera images ensures that this area will get a lot of attention in the future.

2. Nonuniform rational B-splines

A variety of parametric surface patches have been used in CAGD. We hear a lot about NURBS these days, and it is generally accepted that this class of parametric surface patches is rapidly becoming a de facto industry standard. Actually, NURBS have been around since the early days of computer graphics, since they result directly from applying B-splines in the context of homogeneous coordinates. One nice feature about NURBS curves is that we can represent conics exactly.

As NURBS become more accepted, we will need to include them as nodes in a constructive solid geometry (CSG) context for object description. For this to happen, we must address problems related to design, manipulation, and display. For example, in CSG we must answer the basic question: Given a point \((x, y, z)\), is this point in the object or not? Right now, we cannot easily answer this easily stated question for NURBS-bounded objects. Other basic operations needed to incorporate NURBS in a CSG framework include surface-to-surface intersection, surface offsets, and automatic mesh generation.

In addition to the normal control points associated with B-
spline functions, NURBS have additional parameters: the weights in the denominator, which the user must specify. In some situations, we need techniques for automatically selecting these values, thereby lifting this heavy burden from the naive user. Because users of modern workstations have come to expect realistic quality renderings, we must develop effective, robust algorithms for computing the ray intersections in ray tracing.

3. Scattered data modeling

One of CAGD’s major contributions has been the development of methods for extracting the relationships inferred from empirical or simulated data, a process sometimes called scattered data modeling. The 1980s saw rich and broad development in the case of bivariate scattered data interpolation techniques. However, most interesting cases today, such as finding the pressure on an aircraft wing or assessing mineral deposits in a geological feature, involve 3D domains and domains restricted to a 2D manifold in 3D space. For some methods, extension to 3D might merely involve the appropriate 3D redefinition of a distance function, but in general the elevation is not so trivial. For example, in 2D the number of triangles in a triangulation of a convex hull is fixed, while in 3D even a simple cube can be “triangulated” into 5 or 6 tetrahedra, which lends ambiguity to algorithms attempting to find an optimal triangulation.

Other research in this category includes higher (than three) dimensional domain data, restricted domain (surface on surface and manifold) data, and vector and constrained range data. It also covers methods for dealing with very large (millions or billions) point sets.

4. Interrogation

While strikingly realistic computer-generated images are possible today, we cannot ascertain the quality and technical smoothness of a surface from the conventional illumination models used, regardless of their accuracy or resolution. We might not detect anomalies in the surface until we examine a physical model, which is too late. Interrogation techniques can minimize the possibility of this inefficiency. Reflection lines, Gaussian curvature texture mapped onto a surface, and focal surfaces are examples of promising techniques that can reveal a surface’s problem areas at the design stage.

Further development of these and related techniques will help us solve the overall problem of surface design and description. Extending this concept to the case of volume data and volume renderings will also be valuable and interesting. In general, when trying to learn something by visual inspection through computer-generated images, the more tools the better.

5. Implicit versus parametric representation

The development of CAGD has seen two distinct approaches to representing surfaces in 3D space:

1. Parametric methods with a representation of the form \((x(u, v), y(u, v), z(u, v))\), which maps a 2D domain containing \((u, v)\) to 3D space.

2. Implicit methods that define a surface as a set of points \(\{x, y, z\) such that \(F(x, y, z) = 0\).

The parametric approach is associated with spline functions and has been quite successful for the general representation and design of free-form surfaces. The implicit approach is philosophically more closely related to the concepts of CSG and solid modeling.

Both approaches have long lists of pros and cons. One of the implicit approach’s greatest detractors is the lack of techniques for designing free-form surfaces. Over the years there have been some valiant attempts, but no real successes to date. Some of the early proponents of implicit methods have recently said that they feel it is a dead end. But shortly thereafter we saw some new techniques that look encouraging. The debate will continue, and rightfully so, since we all ultimately benefit.

A subproblem within the parametric camp is whether to use triangular or rectangular patches. Rectangles are relatively easy to deal with because they are just one variable process done twice (tensor products or Boolean sums). However, problems arise in trying to represent certain objects using only rectangular patches. In such cases, triangular patches prove useful. They arise naturally in other problems as well, but adding triangular patches to a CAD system complicates matters considerably. Allowing two types of patches requires two sets of procedures for dealing with them (rendering, offsets, intersections, and so forth). While we can argue that a rectangular patch is just two triangular patches, I doubt we will soon have a system based only on triangular patches given the strong momentum behind rectangular patches. This debate will also continue.

6. Data visualization and other uses

Data visualization uses computer-generated images to help scientists extract knowledge and understanding from experimental or simulated data. Properly prepared images using color, texture, transparency, and a myriad of other techniques can convey a tremendous amount of information about a data set quickly. CAGD can greatly aid the development of these techniques, such as those for representing the surfaces that separate 3D flows.

Understanding flows is fundamental to understanding many basic phenomena in science. Flows occur at all scales, from molecular beams to mesoscale oceanic flows to galactic winds. Researchers have already established the usefulness of topological methods for understanding 2D flows, so we can predict the value of topological methods based upon separating surfaces. CAGD researchers understand representing surfaces. However, already published algorithms on surfaces will probably not apply directly to the representation of separat-
As with many areas of science and engineering, CAGD has a need for and can benefit from the use of parallel and distributed computing. High dimensions and large data sets carry heavy computational burdens. In some areas of CAGD, the requirements of interaction demand speeds possible only through parallelism. Realistic renderings of 3D objects, interrogation methods, and other visualization techniques have this property. However, except for a few examples of special hardware for some basic CAGD algorithms, the CAGD-specific work in parallelism has gone at a pace and in a direction similar to other areas.

8. CAD/CAM technology

As CAD/CAM evolves within the mutual influence of existing manufacturing and engineering processes, it becomes more and more important to develop and use standards. One important area for this is in the distribution and exchange of geometric information. For example, the headlight manufacturer must have from the automobile manufacturer a usable form of the geometry for the cavity that will contain the headlight. To the extent that CAD/CAM technology depends on the techniques of CAGD, we must consider these standards from the standpoint of both theory and practice. On the other hand, in some cases, real-world CAD/CAM systems dictate the direction that further CAGD research will take. Feature-based design, n-sided patches, and blending techniques are some examples of this type of research.

9. Foundations

Several key concepts are unique to or originated in CAGD. We need to put these concepts on a firm mathematical foundation not only to promote CAGD's maturation, but to enable these concepts to influence other areas. I can more fully explain this premise using an example: geometric continuity.

The basic idea of geometric continuity originated with v-splines, curves that have individual component functions that are only $C^1$ yet have continuous curvature. They can be parameterized to be $C^2$ without changing the shape of the curve. So, from a geometrical point of view, they are really $C^2$; it just happened that the way the curve was represented made it not $C^2$. If a curve can be changed from $C^1$ to $C^2$ without changing its graph, then the mathematical definition of continuity is deficient in this context. Possibly geometric continuity, as developed in CAGD, is the proper concept, but it needs deeper underpinnings and mathematical footing.

Other fundamental topics of CAGD with the potential to become part of general scientific knowledge include aspects of triangulations and tessellations, geometric construction algorithms, characterization by optimization and organizing formulas such as blossoming. These are only samples from a much larger list.

10. Interface, Manufacturing and VR

To some people, a CAD system consists of a collection of procedures performing the various tasks needed to effectively design and represent a 3D object. Providing a user environment is brushed off as an interface problem, the responsibility of the implementing programmer. This indifference to the interface doesn't negate the importance of coherence between the various procedures. The sum is only greater than the parts if the system is put together with careful attention to how it will be used. Manipulating geometry in an interactive environment so as to effect the design of a 3D object is not a simple matter, but in these days—at the advent of virtual reality—it can be very exciting. Of course, just stating that virtual reality will take care of the interface issue sidesteps the interface issue as much as does ignoring its importance. However, once we progress past the hyperbole and gadgetry of virtual reality and these techniques (and associated hardware for 3D input and 3D display) become ingrained in the methods we use to solve problems, CAD/CAM within a VR environment will lead to effective collaborative design and production.

Some researchers have coined the acronym Vice (Visualization in Concurrent Engineering) to describe these futuristic scenarios, where geographically isolated designers, engineers, manufacturers, and economic consultants all interact with some product as it goes through a virtual evolution from idea to finished product. I am convinced that the strong research being done in CAGD today will bring these visions to fruition in the near future.

Acknowledgments

This project was supported and sponsored by the IEEE Computer Society Technical Committee on Computer Graphics. I am also happy to acknowledge the support of the Computer Science and Engineering Department at Arizona State University. Through this, Keith Voegtle has been a generous source of ideas and support. I thank him for this. In addition, I would like to thank the entire CAGD community. It would be difficult to find a single person who has not contributed in some way. Some people contributed papers, some served as referees, and many contributed in an advisory capacity. The names are too numerous to mention, but I gratefully acknowledge all of their contributions.

Gregory M. Nielson is a professor at Arizona State University, where he teaches and does research in computer graphics, computer-aided geometric design, and scientific visualization. He is one of the founders, and serves on the steering committee for, the IEEE-sponsored conference series on visualization. He is currently a director of the IEEE Computer Society Technical Committee on Computer Graphics. He serves on the editorial boards of several professional journals, including IEEE CG&A, and is a guest scientist at Lawrence Livermore National Laboratory. Nielson received his PhD from the University of Utah in 1970.

Readers may contact Nielson at Computer Science and Engineering, Arizona State University, Tempe, AZ 85287-5406. His e-mail address is nicelson@asuax.cac.asu.edu.