The Visual Computer

The mathematics of computer graphics

Rae A. Earnshaw

University of Leeds,
LEEDS LS2 9JT, United Kingdom

Until relatively recently, researchers in computer graphics paid scant attention to the numerics of their computations. Computation was used as a simple tool to evaluate algorithms or transform data into some appropriate pictoral representation. Thus standard computer graphics texts have little to say about numerical methods, just as earlier numerical analysis textbooks had little to say about computer graphics. This is now changing, for the important reasons outlined in this paper.

Key words: Numerical methods – Euclidean geometry – Picture generation – Computer graphics – Image processing – Interpolation – Fractal mathematics

1 Computer graphics and numerical methods

1.1 Introduction

To date there has been relatively little interaction between computer graphics and numerical methods, except in standard 'mathematical' areas. A number of important considerations are currently reversing this trend. Geometric computations must be performed more accurately (in some sense) since the power of rendering and presentation now enables model construction to be more clearly seen. Computer-aided animation and image synthesis can consume large amounts of compute power – such computations need to be more carefully constructed. Computer graphics and image processing techniques are drawing closer together as a result of the developments in parallel processing architectures, silicon-compilation and execution, transputers, and other engines. Thus traditional aspects of image processing such as sampling, aliasing, Fourier transforms, convolution and basic systems theory are developing their correlates in the computer graphics field. And finally, further areas of overlap and intersection are in the areas of differential and algebraic geometry, fractal mathematics, curve definition, dynamics, and shape deformation, for reasons to do with the exploitation of these techniques (often originating in more 'classical' fields) in specialised areas of computer graphics.

This synergism will produce an added rigour to the definition and execution of graphics processes – whether in hardware or software – and enable the coupling of numerics and pictures to take place to their mutual benefit. This in turn will enable the next generation of graphics processors to be designed on a rigorous and consistent basis. When this is coupled with the application of formal methods to algorithm specification and execution, pipeline transformations, graphics language and interface design, and parallelism, our understanding of the processes of picture production will be greatly enhanced.

This paper reviews the aspects of mathematics and numerical analysis of relevance to computer graphics and the anticipated developments in the future.

1.2 Survey and background

Historically there has always been an overlap between computer graphics and numerical methods

The Visual Computer (1987) 3:115–124

and geometry due to requirements in areas such as the following:

(i) Manipulation of matrices representing points, lines etc.
(ii) Calculation of Euclidean distances e.g. unit normals and perpendiculars from points on to lines
(iii) Calculation of line intersections and lines with planes
(iv) Calculation of points on curves and surfaces in order to represent them on some display device
(v) Mapping pictures on to display screens that are effectively approximate, e.g. discrete grid or raster
(vi) Prevention of the accumulation of round-off error when approximating pictures on incremental devices

In addition, a knowledge of numerical properties and methods has been essential in the following areas:

(i) Choice of the type of curve or surface to best represent the physical properties of the system or object being represented
(ii) Choice of curve or surface that is invariant when the defining points are transformed or rotated (essential for the preservation of shape)
(iii) Methods for calculating expressions and functions so that intermediate values do not cause overflow or underflow on the particular hardware being used
(iv) Reformulation of algorithms to produce faster execution speeds
(v) Producing reliable, and accurate geometric computations to ensure consistency under all circumstances

Further interesting areas are those concerned with artefacts (e.g. jaggies or staircasing) and are often due to a combination of factors. Well-established techniques exist for smoothing out such anomalies.

Standard mathematical techniques for representing points, lines, curves and surfaces, and manipulating them, are not the province of this paper. They are well covered in standard text books such as "Mathematical Elements for Computer Graphics" (Rogers and Adams 1976).

The following is a convenient summary of the operations that must be performed for basic modelling and picture production given by Blinn (1984):

Representation. The exact form and use of the numerical parameters which define the object must be specified.

Modelling. It must be possible to generate the appropriate mathematical parameters from some conceptualisation of the desired shape in the mind of the designer.

Transformation. These are the basic geometrical transformations (scaling, rotation, translation and perspective) which are represented by the homogeneous 4 by 4 matrix. This operation is usually performed by deriving new mathematical parameters for the transformed surface from those of the old surface and the contents of the transformation matrix.

Boundary checking. A surface defined solely by functions may potentially stretch to infinity. Real objects are typically modelled as pieces of such surfaces sliced off at various boundaries. These boundaries then form space curves. Determination of whether a point is inside or outside the boundary must be performable.

Intersections. It must be possible to find the intersection of the surface with other surfaces, lines, and planes.

Surface normals. It must be possible to calculate the surface normal vector at any desired point. This is useful in two contexts: it serves to define the silhouette edge of the surface, and it is also a prime constituent of the intensity calculation when shaded pictures are to be drawn.

2 Mathematical models and representations

As a branch of mathematics dealing with shape and spatial relations, geometry has been the focus of attention with regard to the handling of 3D objects in a computer. However, one of the basic problems in 3D graphics is the representation of objects in a way which makes the analysis and rendering of them feasible in a finite time. In addition, it would be particularly useful to have some generalised model into which different shapes and components could be fitted, since these would be more tractable to elegant mathematical representa-

tion and manipulation. Informationally complete representations would enable any well-defined geometrical property of any represented object, or set of objects, to be calculated automatically. Requicha and Tilove (1978), Requicha (1980), and Requicha and Voelcker (1982, 1983) summarise some of the early approaches to solid modelling. Aspects of modelling and computational geometry of current interest are domain extensions for modellers; handling dimensioning, tolerances, and geometrical constraints; algorithms for rapid editing; null object and interference detection; conversion from one representation into another; analysing the complexity of geometric algorithms; and handling precision problems in numerical geometry.

Representing objects in the form of hierarchical structures is surveyed by Tilove (1981, 1984), Meagher (1982), Kedem and Ellis (1984), and Samet (1984). Further advances are noted in Samet (1985), Samet and Tamminen (1985), Nelson and Samet (1986), and Muuss (1987). Hybrid models are needed to effectively cater for all the requirements of modelling and representation. In addition, as the hardware of display architecture evolves, data structures and algorithms have to be mapped in such a way as to fully exploit the capabilities of the new developments.

A related approach to the modelling and representation of shape is presented by Brady (1981, 1984a, b) from the standpoint of computer vision. This applies more to vision recognition problems than to computer graphics, but automated robots and assembly plants need to know how to deal with shape as part of the process of recognition and manipulation.

As an example of the interdependency of representation, model, and picture, van Overveld (1987) considers the following example. When a 3D object is rendered into pixel space, the method chosen depends on the mathematical representation of the object to be rendered. If this is a list of polygons, then processing consists of raster scan conversion for each polygon in order to locate all the pixels of interest. If it is represented as a CSG tree, then this is interpreted and the appropriate CSG primitives rendered. However, a boundary representation of the same object (to highlight joints for example) involves a different strategy requiring the calculation of the intersection of the CSG primitives and utilising a line or curve drawing algorithm to select the pixels to be highlighted. But there is no guarantee that these pixels coincide with the locations of the joins as they appear in the picture generated initially. Franklin (1986) notes the same problem when different algorithms are being used in the same picture – producing missing and spurious pixels, the tell-tale signs of anomalous behaviour. What is needed is a more consistent and unified approach based on a rigorous model, with uniform operations.

A further example given by Forrest (1985) highlights the complex scenario in evaluating the seemingly simple case of the intersection of two lines. Careful attention to numerical detail is required and also a consistent ordering of geometric operations within practical systems. Mixing of different types of coordinate system can also be a non-trivial situation; mapping from one into the other requires detailed consideration. For example, the intersection between two line segments whose end points are on an integer grid may not be representable in the floating point number system, since the rounding or truncation to a unique floating point number has to be done consistently.

Some possible solutions are standardised floating point arithmetic; algebraic and symbolic manipulation; interval arithmetic (e.g. Mudur and Koparker 1984); and the use of rational arithmetic. However, the latter cannot cater for a general set of points.

Considerations in the areas of production automation and manufacturing have made it clear that it is no longer sufficient to use exact models to represent objects. Tolerance needs to be formulated and represented in some way. Process planning – the automation of robots and machine tools – needs more than just a description of stationary objects – we need to know how these objects move in space, and where they are at any given point in time (Cameron 1984). This involves spatial updating processes. In specialised application areas (e.g. sheep shearing machines and robots), flexible models have to be represented.

Some further problems of interest are handling the complexity introduced by assemblies of objects; the representation and interpretation of object properties such as mass, surface texture, and appearance; and handling the processes of motion, machining, and assembly of parts. One of the ultimate goals is to be able to automate the creation, analysis, transmission and management of all product definitions, process definitions, and associated business data.

3 Euclidean geometry

More general considerations of numerical computations in a graphical or geometrical context are given by Duff (1984), including order of convergence, designing look-up tables, function evaluation, intersection calculations, and spline interpolation and approximation. An interesting example in calculating Euclidean distance is cited from Moler and Morrison (1983). An algorithm for calculating $sqrt(a*a+b*b)$ is needed which is fast, robust, and does not cause overflow or underflow when calculating the intermediate values of $a*a$ or $b*b$. The method by Moler and Morrison does not suffer from these problems, provided the result is in range. It has cubic convergence and may even be faster than $sqrt(a*a+b*b)$. The following is a *C* implementation:

```
double hypot (p, q)
double p, q;
double r, s;
if (p<0) p=-p
if (q<0) q=-q
if (p<q) (r=p; p=q; q=r;)
if (p==0) return 0;
for (; ;) (
        r=q/p
        r*=r;
        s=r+4
        if (s==4) return p;
        r/=s;
        p+=2*r*p;
        q*=r;
        )
```

The result is accurate to 6.5 digits after two iterations, to 20 digits after three, and 62 digits after four. Thus normal use would specify the extent of the iteration and omit the test. Dubrelle (1983) analyses the algorithm geometrically and outlines a set of generalisations with arbitrarily large order of convergence. Duff (1984) estimates that calculating Euclidean distance accounts for 90% of the square roots in computer graphics applications. In addition, most illumination models need the unit normal to each visible surface to be computed at each pixel.

A further example, the CORDIC rotation algorithm from Volder (1959) and described in Turkowski (1982), can be used to calculate rotations in 2D, rectangular to polar and polar to rectangular conversion, Euclidean point-point, point-line and point-line segment distance, circular and hyperbolic trigonometric functions, exponentials, logarithms, and square roots.

Locating the zeros of non-liner equations is required for producing the ray-traced rendering of a 3D surface. Blinn (1982) developed a hybrid Newton-Raphson/false position iterative method to locate the right root, with quadratic convergence whenever Newton-Raphson would give the right root.

Hanrahan (1983) uses multivariate polynomial functions for ray-tracing algebraic surfaces, including planes, quadric surfaces and tori.

An introduction to curve interpolation and approximation with particular reference to computer graphics is given in Brodlie (1985) and Earnshaw (1985). There are many survey works on the uses and applications of the different kinds of splines (Bartels et al. 1984; Barnhill and Riesenfeld 1974; Barsky 1987).

4 Sampling, convolution and fourier transforms

Methods and techniques from image processing have increasing relevance for computer graphics as the two areas draw closer together through the use of analogous procedures in the areas of sampling, parallel processing, real-time image production, recognition of graphics and text, interactive CAD and robotics, AI and computer vision, and theoretical foundations.

Aliasing arises from inadequate sampling of the continuous environment (the picture) with the discrete raster grid (the display). This causes jaggedness along lines (also known as 'staircasing'). Similarly in the time domain, temporal aliasing arises from objects moving quickly with respect to the camera causing strobing in animated sequences of objects (Porter 1984). The basic problem in the former is scanline to scanline changes, and in the second frame to frame changes.

Two theorems relating to convolution and multiplication in the time and frequency domain elaborate on the problem of aliasing (Kajiya 1984a, b). When an input signal is sampled, its values at an equally spaced set of points are taken as representative of the complete signal. Sampling an input signal $s(t)$ at discrete intervals can be represented

as a multiplication by a train of delta signals:

$$s(t)=\sum_{n=-\infty}^{\infty}\delta(t-nT_s)$$

where T_s is the sampling interval.
The Fourier transform of a train of delta signals is a similar sequence with different spacing:

$$s(w)=\sum_{n=-\infty}^{\infty}\delta(w-n\Omega_s)$$

where Ω_s is the sampling rate.
Thus sampling an input signal can be modelled as a convolution in the time domain. The convolution is a sequence of delta signals separated by Ω_s, and results in copies of the input spectrum being inserted into the frequency domain, each bein centred at the position of the corresponding delta signal. This may give rise to regions where these copies overlap, producing aliasing. In the spatial domain this produces jaggies; in the frequency domain it produces a set of frequencies which wrap around to produce different frequencies. For no aliasing to occur the original spectrum should have no frequencies beyond a small range around zero (Shannon sampling theorem).
This if $x(t)$ has Fourier transform $X(w)$ such that $X(w)=0$ for $|w|>\Omega_s/2$, then $x(t)$ can be determined by

$$x=x(nT)$$

where T is the sampling interval. This uses the convolution and Fourier transform properties. To recover the original signal, all extra copies are suppressed by convolving the output with the kernel: –

$$h(t)=\sin(\Omega_s\, t/2)/(\Omega_s/2).$$

This has as its Fourier transform a rectangle function which is non-zero only in the region about the original spectrum.

$$H(w)=\begin{cases}1 & \text{if } |w|<\Omega_2/2\\ 0 & \text{otherwise.}\end{cases}$$

Further material is contained in Kajiya (1984), Oppenheim and Shafer (1975) and Pratt (1978).

5 Further topics

5.1 Deformation of primitives

Barr (1981, 1984a, b) outlines hierarchical solid modelling operations for twisting, bending, tapering, compression, and expansion of geometric objects. The position vectors and normal vectors in the simpler objects are used to calculate the position and normal vectors in the more complex forms. Each level in the deformation hierarchy requires an additional matrix multiply for the normal vector calculation. In addition to simulating the bending of bars or sheet metal, deformations can also be utilised for flexible objects such as plastic, fabric, or rubber.

5.2 Einstein summation notation

Barr (1984a, b) introduces the use of the Einstein summation notation for a short-hand method of expressing multi-dimensional cartesian equations. The advantages of this representation include ease of manipulation of long expressions, particularly for cross products, determinants, rotations, and matrix inverses. Further references to this convention can be found in the texts describing tensor analysis and 3D mechanics (Segel 1977; Solkolnikoff 1956).

5.3 Differential and algebraic geometry

The aspects of differential geometry of relevance to computer graphics are those concerned with manifolds, differential forms, and connections. Algebraic geometry contributes homogeneous polynomials, factorisation, elimination, and the theory of plane curves (Spivak 1965, 1975; Kajiya 1984; Hartshorne 1977).

5.4 Dynamics and motion

Classical Newtonian mechanics can provide the basis for the animation of objects. There are many texts which summarise these aspects of computational physics (Courant and Hilbert 1962; Arnold 1975; Goldstein 1980).

5.5 Human interface modelling

Graphics is often an integral part of the interface between man and machine. Effective utilisation of computers will rely increasingly on amore rigorous and quantitative assessment of human-computer interface characteristics (Newman 1987). This in turn will benefit from a modelling of the processes at the interface; a study of the appropriateness of the choice of input tools for particular applications; and the construction of methodologies for specifying the design of the interface (Preece et al. 1987; Pfaff 1985).

5.6 Fractal mathematics

Uncovering orderliness within the framework of apparent chaos is one of the purposes of the investigation of nature, and its representation by means of models, theories, and mathematical constructions. Interestingly, there appears to be a symmetry between the micro analysis of nature and its macro appearance and representation in the large. This may be coincidental but it is more likely to reflect the deep structure of nature and the laws that govern the relationship of its constituent parts. Examples of macro structure and appearance are trees, clouds, coastlines, rivers, mountain ranges, and landscapes – all apparently random – but in fact encapsulating an underlying principle of order which can be expressed in terms of simple mathematics. However, Euclidean geometry is inadequate for this purpose; fractal geometry (Mandelbrot 1977, 1983) provides the basis for a framework for representing shapes in nature and also entities in mathematics. Fractal geometries exhibit the property of self-similarity, i.e. the whole replicates the part, and also has a dimensionality. Earlier geometric forms such as dragon curves, the von Koch snowflake curve, and the Sierpinski curve, all display exact self-similarity, whereas objects in nature display statistical self-similarity. For example, a coastline drawn at different scales (equivalent to zooming into or out of a given region) produces a set of pictures that are fundamentally similar. In fact, they are so similar that they could be taken for different sections of the same coastline all at a constant scale. In physical terms this appears like regularity within irregularity. In mathematical terms it represents a high degree of invariance under changes of scale.

The combination of Mandelbrot sets, computers, and computer graphics has provided a powerful toolset for exploring the complex plane and the behaviour of dynamical systems: "Imagining the formerly unimaginable" (Salinger 1987). This has contributed greatly to our understanding of both mathematics and complex systems, and has been well documented (e.g. Pietgen and Richter 1985). This phenomenon is illustrative of an important point: computer graphics has provided a powerful tool to uncover mathematical and physical behaviour, and in turn the mathematics has provided the basis for developments in computer modelling and representation of natural scenes. The former is a very interesting analogue of the latter, and vice-versa. There has been some discussion on the extent to which the methods used by computer graphicists (e.g. Fournier et al. 1982) embody the mathematical purity of the Mandelbrot fractals (Mandelbrot 1982), but the pictures produced continue to be impressively realistic, whether for natural terrain, flakes, or clouds. Voss (1985) summarises some of the principal mathematical constructs and relations underlying the definition of fractals.

5.7 Space-filling curves

Some recent developments in space-filling curves and Peano curves embody a different strategy for generating pixel images giving greater speed. These are described in Peano (1890), Witten and Wyvill (1983), and Cole (1987).

5.8 The mathematics of parallelism

Designing algorithms for parallel architectures and their exploitation for the manipulation and display of objects is an area of current interest. Moore et al. (1987), Dew et al. (1985, 1986), Händler et al. (1986), and May and Shepherd (1986) provide a summary of the current work, including systolic arrays and their implications. Fuchs (1987) gives a summary of current VLSI work.

Techniques for mapping synchronous, data-independent calculations such as convolutions and transformations are well understood. However, extensions to the synchronous and data-dependent cases are much more difficult – currently there is no formal and comprehensive treatment. Mapping algorithms on to transputers requires a partition-

ing which minimises the need to communicate with the overall model.
Exploiting parallelism is the subject of some recent investigations (e.g. Theoharis 1986). Occam provides a parallel processing environment. Goldfeather et al. (1986) describe a method of exploiting spatial parallelism by using a central control and a logical processor at each node to evaluate the polynomial. For quadratic primitives the results have been impressive in terms of executing the processes required by the CSG trees. However, in order to develop automated methods for implementing algorithms on parallel processors, some form of general model representation for parallelism is needed.

5.9 Methodology of design

Recent studies have focussed on the area of design with a view to obtaining greater understanding and elaboration of the design process (Lansdown 1985, 1987a–c; Lawson 1983, 1987; King 1987). A number of models have been postulated: firstly, those based on gradual iteration towards the final design in a well-defined way (so called 'robust' designs), and secondly those that represent progress in a more discontinuous way – related to the generation of new ideas (so called 'lean' designs). Models in Catastrophe Theory have been used to illustrate that these two approaches can be represented by an overall unified model. Modification of existing designs can be aided by the provision of 'standard' options for the designer to choose from. Where a chosen sequence of these options is interrelated (e.g. in designing a building the doors and windows cannot be larger than the walls) it should be possible to incorporate knowledge-based approaches into the design process. However, dealing with the incompleteness which is the essence of the process in a non-trivial task.

6 Conclusions

6.1 Algorithm formulation and complexity

Students of algorithms have demonstrated the inherent complexity in even the seemingly simplest of algorithms. Computer graphics algorithms concerned with hideen-line and hidden-surface removal have received considerable attention with a view to optimisation and improvement. Bresenham's algorithm (1965) has been the subject of much investigation and refinement – even involving program transformations (Sproull 1982). However, formal and mathematical analysis of anything other than the simplest of algorithms (e.g. sorting and searching) has proved inordinately difficult (Tucker 1985).

6.2 Handling geometric and algorithmic complexity

Assemblies of objects or aggregations of components are surprisingly difficult to represent in an informationally complete sense, such that all the requirements of the model can be satisfied unambiguously. In addition, the complexity of computations arising from even the simplest operation is such that greater optimisation or more powerful hardware is needed for the task to be performed in a reasonable time. Shamos (1975) has noted the complexity arising in geometric operations. Mapping algorithms on to parallel architectures is a non-trivial task. Forrest (1987) argues for the application of rigorous software engineering techniques when constructing large and complex geometric systems, such that computations can be performed reliably, accurately, and consistently.

6.3 Notation and conceptualisation

Appropriate notation and representation of abstraction will enable a better understanding of the processes and problems involved. In addition, a mental model that is able to represent a mass of complexity often suggest new ways of thinking about problems. This is often the way forward.

6.4 Integration of computer graphics and computer vision

Computers are rapidly moving from information processing machines to vision processing machines. The input process corresponds to the transformation of an object scene into an object representation in the computer for analysis and manipulation. The output process corresponds to the transformation of object data into picture data.

If all this has to be performed in real-time, then the processes have to be efficiently represented and executed. A model of a vision processing machine is needed into which these components will fit as integral parts.

6.5 Mathematics, models and computer graphics

Some recent studies in the area of models have demonstrated the power, capability and advantages of a rigorous conceptualisation framework. Examples are Kunii (1987) in CAD and graphics communication networks, Woodwark and Quarendon (1987) in graphics, and Hall (1987) in colour reproduction and illumination models. Future work should build on this rigorous and systematic approach. Computer graphics is rapidly moving from a discipline based largely on pragmatics and trial and error solutions to one based on rigorous analysis and formal methods. The unifying tools in this transition are models, metrics and mathematics.

References

Arnold VI (1975) Mathematical methods of classical mechanics. Springer, Berlin Heidelberg New York

Barnhill RE, Riesenfeld RF (1974) Computer-aided geometric design. Academic Press

Barr AH (1984) Global and local deformations of solid primitives. SIGGRAPH Tutorial Notes

Barr AH (1981) Superquadrics and angle-preserving transformations. IEEE Comput Graph Appl 1(1):11–23

Barr AH (1984) Introduction to the Einstein summation notation. SIGGRAPH Tutorial Notes

Barr AH (1986) Ray tracing deformed surfaces. ACM SIGGRAPH 20(4):287–296

Barsky BA (1987) Computer graphics and geometric modelling using beta-splines. Springer, Berlin Heidelberg New York Tokyo

Bartels RH, Beatty JC, Barsky BA (1984) An introduction to the use of splines in computer graphics. University of Waterloo TR CS-83-09, UC Berkeley TR UCB/CSD 83-136

Blinn JF (1982) A generalization of algebraic surface drawing. ACM Trans Graph 1(3):235–256

Blinn JF (1984) The algebraic properties of homogeneous second order surfaces. SIGGRAPH Tutorial Notes

Brady M (1981) Computer Vision. North-Holland, Amsterdam

Brady M (1984a) Representing Shape. Report, MIT AI Laboratory

Brady M (1984b) Criteria for rerpresentations of shape. In: Rosenfeld, Beck (eds) Human and machine Vision. Academic Press

Bresenham JE (1965) Algorithm for computer control of a digital plotter. IBM Syst J 4(1):25–30

Brodlie KW (1985) Methods for drawing curves. In: Earnshaw RA (ed) Fundamental algorithms for computer graphics. Springer, pp 304–323

Cameron SA (1984) Modelling solids in motion. PhD Thesis, University of Edinburgh

Carpenter L (1980) Computer rendering of fractal curves and surfaces. ACM SIGGRAPH (Abstract) 14(3):109

Chazelle B, Dobkin DP (1980) Detection is easier than computation. Proc 12th Annual ACM Symp Theory Comput, pp 146–152

Cohen E (1983) Some mathematical tools for a modeller's workbench. IEEE Comput Graph Appl 3(7):63–66

Cole AJ (1987) Compaction techniques for raster scan graphics using space-filling curves. Comput J 30(1):87–92

Courant R, Hilbert D (1962) Methods of mathematical physics. Wiley

Dew PM, Dodsworth J, Morris DT (1985) Systolic array architectures for high performance CAD/CAM workstations. In: Earnshaw RA (ed) Fundamental Algorithms for Computer Graphics. Springer, pp 659–694

Dew PM, Manning LJ, McEvoy K (1986) A tutorial on systolic array architectures for high performance processors. Rep No 205, Dept Computer Studies, University of Leeds, UK

Dubrulle AA (1983) A class of numerical methods for the computation of pythagorean sums. IBM J Res Dev 27(6):582–589

Duff T (1984) Numerical methods for computer graphics. SIGGRAPH Tutorial Notes

Earnshaw RA (1985) A review of curve drawing algorithms. In: Earnshaw RA (ed) Fundamental algorithms for computer graphics. Springer, Berlin Heidelberg New York Tokyo, pp 289–301

Pfaff GE (ed) (1985) User interface management systems. Springer, Berlin Heidelberg New York Tokyo

Forrest AR (1985) Computational geometry in practice. In: Earnshaw RA (ed) Fundamental algorithms for computer graphics. Springer, pp 707–724

Forrest AR (1987) Computational Geometry and Software Engineering. In: Rogers DF, Earnshaw RA (eds) Techniques for computer graphics. Springer, Berlin Heidelberg New York Tokyo (to be published)

Fournier A, Fussel D, Carpenter L (1982) Computer rendering of stochastic models. CACM 25:371–384

Franklin WR, Barr AH (1981) Faster calculation of superquadric surfaces. IEEE Comput Graph Appl 1(3):41–47

Franklin WR (1986) Problems with raster graphics algorithms. In: Kessener LRA, Peters FJ, van Lierop MLP (eds) Data structures for raster graphics. Springer, Berlin Heidelberg New York Tokyo

Fuchs H (1987) VLSI for Graphics. In: Rogers DF, Earnshaw RA (eds) Techniques for Comput Graphics. Springer, Berlin Heidelberg New York Tokyo (to be published)

Goldfeather J, Fuchs H (1986) Quadratic surface rendering on a logic-enhanced frame-buffer memory. IEEE Comput Graph Appl 6(1):48–59

Goldfeather J, Hultquist JPM, Fuchs H (1986) Fast constructive solid geometry display in the pixel-powers graphics system. ACM SIGGRAPH 20(4):107–116

Goldstein H (1980) Classical Mechanics. Addison-Wesley

Guillemin V, Pollack A (1974) Differential Topology. Prentice-Hall

Hall R (1987) Color reproduction and illumination models. In: Rogers DF, Earnshaw RA (eds) Techniques for computer graphics. Springer, Berlin Heidelberg New York Tokyo (to be published)
Händler W, Haupt D, Jeltsch R, Juling W, Lange O (eds) (1986) CONPAR86. Proc Conf Algorithms and Hardware for Parallel Processing, Aachen (September 1986) Lect Notes Comput Sci 237
Hanrahan P (1983) Ray tracing algebraic surfaces. ACM SIGGRAPH, pp 83–90
Hartshorne R (1977) Algebraic geometry. Springer, Berlin Heidelberg New York
Kajiya JT (1984) Transform Theory. SIGGRAPH Tutorial Notes
Kajiya JT (1984) Differential and algebraic geometry. SIGGRAPH Tutorial Notes
Kedem G, Ellis JL (1984) Computer structures for curve-solid classification in geometric modelling. TR84-37, Microelectronic Center of North Carolina
King M (1987) Towards an integrated computer art system. In: Earnshaw RA, Lansdown RJ (eds) Computer Graphics in Art, animation and design. Springer, Berlin Heidelberg New York (to be published)
Kunii TL (1987) A model-driven approach to CAD and graphics Communication Networks. In: Rogers DF, Earnshaw RA (eds) Techniques for Computer Graphics. Springer, Berlin Heidelberg New York (to be published)
Lansdown RJ (1985) Requirements for knowledge-based systems in design. System Scinulation, London
Lansdown RJ (1987a) Computer Graphics in Design. In: Rogers DF, Earnshaw RA (eds) Techniques for Computer Graphics. Springer, Berlin Heidelberg New York Tokyo (to be published)
Lansdown RJ (1987b) A theory of computer-aided design. In: Earnshaw RA, Lansdown RJ (eds) Computer Graphics in Art, Animation and Design. Springer, Berlin Heidelberg New York Tokyo (to be published)
Lansdown RJ (1987) Some notes of fractals. In: Earnshaw RA, Parslow RD, Woodwark JR (eds) Geometric Modelling and Computer Graphics Applications and Techniques. Tech Press
Lawson B (1983) How designers think. Architectural Press, London
Lawson B (1987) Intelligent building systems and coordinated drafting systems. In: Earnshaw RA, Lansdown RJ (eds) Computer Graphics in Art, Animation and Design, Springer, Berlin Heidelberg New York Tokyo (to be published)
Mandelbrot BB (1977) Fractals: form, chance and dimension. Freeman, San Francisco
Mandelbrot BB (1983) The fractal geometry of nature. Freeman, San Francisco
Mandelbrot BB (1982) Comment on computer rendering of fractal stochastic models. CACM 25:581–584
May D, Shepherd R (1986) Communicating process computers. Conf Communicating parallel architectures. Esprit Summer School on Future Parallel Computers
Meagher DJR (1982) The octree encoding method for efficient solid modelling. IPL-TR-032, Image Processing Lab, RPI
Moler C, Morrison D (1983) Replacing square roots by pythagorean sums. IBM J Res Dev 27(6):577–581
Moore W, McCabe A, Urquhart R (eds) (1987) Systolic Arrays. Proc First Internat Workshop on Systolic Arrays (July 1986) Oxford, England. Adam Hilger, Bristol Boston
Mudur SP, Koparker PA (1984) Interval methods for processing geometric objects. IEEE Comput Graph Appl 4(2): 7–17
Mudur SP (1986) Mathematical elements for computer graphics. In: Enderle G, Grave M, Lillehagen F (eds) Advances in Computer Graphics I. Springer, Berlin Heidelberg New York Tokyo
Muuss MJ (1987) Understanding the preparation and analysis of solid models. In: Rogers DF, Earnshaw RA (eds) Techniques for Computer Graphics. Springer, Berlin Heidelberg New York Tokyo (to be published)
Nelson RC, Samet H (1986) A consistent hierarchical representation for vector data. ACM SIGGRAPH 20(4):197–206
Newman WM (1987) Designing integrated systems for the office environment. McGraw-Hill, pp 421–422
Oppenheim AV, Shafer RW (1975) Digital Signal Processing. Prentice-Hall, Englewood Cliffs New Jersey
Overveld van CWAM (1987) A family of algorithms for generating discrete embeddings of continuous objects. In: Theoretical Foundations of Computer Graphics and CAD. Springer, Berlin Heidelberg New York Tokyo (to be published)
Pavlidis TJ (1982) Algorithms for Graphics and image processing. Springer, Berlin Heidelberg New York
Peano G (1890) Sur une courbe, qui remplit toute une aire plaine. Math Ann 36:157–160
Pietgen H-O, Richter PH (1985) The beauty of fractals: images of complex dynamical systems. Springer, Berlin Heidelberg New York Tokyo
Pietgen H-O, Saupe D (1983) Julia – a scheme for the generation of self-similar images. Proc CG83, Online, pp 731–741
Porter T (1984) Motion Blur. SIGGRAPH Tutorial Notes
Pratt WK (1978) Digital Image Processing. Wiley
Preece J, Davies G, Woodman M, Ince DC (1987) A coherent specification method for the user interface of documentation systems. In: Earnshaw RA (ed) Workstations and Publication Systems. Springer, Berlin Heidelberg New York Tokyo (to be published)
Preparata FP, Shamos MI (1985) Computational geometry. Springer, Berlin Heidelberg New York Tokyo
Requicha AAG, Tilove RB (1978) Mathematical foundations of constructive solid geometry – general topology of closed regular sets. TM-27a, Production Automation Project, University of Rochester
Requicha AAG (1980) Representations of rigid solids: theory, methods and systems. ACM Comput Surv 12(4):437–464
Requicha AAG, Voelcker H (1982) Solid modelling: a historical summary and contemporary assessment. IEEE Comput Graph Appl 2(2):9–24
Requicha AAG, Voelcker H (1983) Solid modelling: current status and research directions. IEEE Comput Graph Appl 3(7):25–37
Rogers DF, Adams JA (1976) Mathematical elements for computer graphics. McGraw Hill, New York
Salinger DL (1987) Private communication
Samet H (1984) The quadtree and related hierarchical structures. ACM Comput Surv 16(2):187–260
Samet H (1985) Approximating CSG trees of moving objects. TR-1472, Comput Sci, University of Maryland
Samet H, Tamminen M (1985) Bintrees, CSG trees and time. ACM SIGGRAPH 19(3):121–130
Segel LA (1977) Mathematics Applied to continuum mechanics. Macmillan

Shamos MI (1975) Geometric complexity. Seventh ACM Annual Symp on Theory of Computing, pp 224–233
Solkolnikoff (1956) Mathematical theory of elasticity. McGraw Hill
Spivak M (1965) Calculus on manifolds. Benjamin
Spivak M (1975) A Comprehensive introduction to differential geometry, vol 5. Berkeley
Sproull RF (1982) Using program transformations to derive line-drawing algorithms. ACM Trans Graph 1(4):259–273
Theoharis TA (1986) Exploiting parallelism in the graphics pipeline. MSc Thesis, Oxford University Programming Research Group, PRG-54
Tilove RB (1981) Exploiting spatial and structural locality in geometric modelling. PhD Thesis, University of Rochester
Tilove RB (1984) A null-object detection algorithm for constructive solid geometry CAcM 27(1):684–694
Tucker JV (1985) Theoretical considerations in algorithm design. In: Earnshaw RA (ed) Fundamental algorithms for computer graphics. Springer, Berlin Heidelberg New York Tokyo, pp 855–878
Turkowski K (1982) Antialiasing using Coordinate Rotations, ACM Transactions on Graphics, Vol 1, No 3, pp 215–234
Volder JE (1959) The CORDIC Trigonometric technique. IRE Trans Electronic Comput EC-8(3):330–334
Voss RF (1985) Random fractal forgeries. In: Earnshaw RA (ed) Fundamental Algorithms for Computer Graphics. Springer, Berlin Heidelberg New York Tokyo, pp 805–835
Witten IH, Wyvill B (1983) On the generation and use of space filling curves. Software Pract Exper 6:519–525
Woodwark JR, Quarendon P (1987) The model for graphics. In: Rogers DF, Earnshaw RA (eds) Techniques for Computer Graphics. Springer, Berlin Heidelberg New York Tokyo (to be published)

Rae A. Earnshaw *is Head of the Graphics Team at the University of Leeds, with interests in graphics algorithms, integrated graphics and text, display technology, CAD/CAM, and human-computer interface issues. He has been a Visiting Professor at Illinois Institute of Technology, George Washington University, and Northwestern Polytechnical University, China. He has acted as a consultant to US companies and the College CAD/CAM Consortium and given seminars at a variety of UK and US institutions and research laboratories. He is a Fellow of the British Computer Society and Chairman of the Computer Graphics and Displays Group. He was a Director of the 1985 ASI "Fundamental Algorithms for Computer Graphics, and Co-Chair of the BCS/ACM International Summer Institute on "State of the Art in Computer Graphics" held in Stirling, Scotland in 1986. Earnshaw received his BSc and PhD in computer science from the University of Leeds.*