Editorial – Communications

The uses of computer graphics are many and varied. They range from photorealistic images for films and television, through business graphics, mapping and drawings for scientific visualisation, to computer-aided design, with all its ramifications in technical drawing and presentation imagery. And not only as graphics used for output of information but also for input too, as well as for assisting in user interaction. Underlying these uses are, of course, algorithms and methods both for creating graphical images and for manipulating graphical images and for manipulating graphical data in efficient and well-understood ways.

It is impossible in any one issue of the *Journal* to do full justice to all these multifarious aspects. What we have tried to do is look at some of the underlying technical matters by means of ten papers which cover an interesting subset of the field. Some of these papers – for instance those by Bell and Mason, by Hunter and Willis, and by Moltedo and Noferini – cover methods which might underpin virtually any computer graphics use. Others are more specialised in their application. The papers can be conveniently put together in four groups.

In the first of these groups, Bell and Mason look at how quaternions can be used to allow octree representations of graphical objects to be conveniently manipulated. Quaternions were invented in the nineteenth century by the great Irish mathematician, Sir William Hamilton (1805–1865) for entirely different purposes, and it is only in the last five or six years that their relevance to some of the problems of computer graphics has become apparent. Hunter and Willis examine ways of labelling the nodes of quadtrees (which are the two-dimensional equivalents of octrees) to achieve a compact and simple referencing method. They show that there is a minimum number of bits needed to represent any node in a quadtree, and give us an innovative technique which uses exactly this number of bits. In addition, they describe another new technique which, though not optimal, is more compact than any previously published one. This has the additional virtue of being easy to understand. Suffern shows an application of quadtree techniques, this time for efficient contouring of functions of two variables. By use of this method – which in essence adaptively subdivides the plotting area into manageable portions – some of the disadvantages of earlier methods are overcome. It is worth noting that the idea of adaptive subdivision of the drawing space in order to concentrate computational effort on areas where it is most needed has a long history in computer graphics. It was, for instance, the basis of Warnock's hidden-line removal method – one of the earliest efficient algorithms for this tricky, computational intensive problem.

When, early in the 1960s, I first began using computers for architectural design purposes, my efforts were often criticised by architect colleagues on the grounds that the flowing, curved lines that manual approaches to designing sometimes encourage could not arise by computer means. Indeed, it *was* rather difficult to draw curved shapes with the primitive algorithms we had at that time. I had not, however, noticed much propensity among my colleagues to use free-flowing curves in the architecture of the 1960s and 1970s. Nowadays, it is actually easier to create curved lines which blend seamlessly together by using computer graphics techniques than it is by using any hand methods. The credit for this must go to the many workers who have pursued computer graphics techniques for creating circles, ellipses and other curved forms in an efficient and congenial way. Even now though, the problems are far from completely solved. Two contributions to this subject comprise the second group of papers.

Neal and Pitteway look at some ways of calculating points which make up the circumference of a circle. They base their work on a technique which, as they point out, seems to have arisen from a programming error in the 'Sketchpad' graphics system of the early 1960s. The algorithms they derive from an analysis of this early and apparently erroneous technique are at least as efficient as others and more efficient than most. In particular, their method has considerable advantages when used for cutting circles with computer-controlled machines. Pham describes his new method for dealing with free-flowing curves which have to fair smoothly into one another. He derives an efficient method which is easier to implement and use than existing ones.

The third group of papers look at some of the difficulties which arise when dealing with digitised graphical images. Puliti and Tascini put forward a knowledge-based approach to interpreting technical drawings of the sort architects create. The aim is interactively to help the computer acquire the 'meaning' of what it sees in a digitised drawing (analogous to the sense in which optical character reading programs give computers the 'meaning' of text). This is, however, a very difficult task, but their paper gives us valuable clues on how it might be tackled. Dehne and Ficocelli examine a subset of the task: how to get the computer to distinguish between the 'pepper and salt' spots that inevitably show up on digitised graphical images and the organised marks in a dotted line. Their algorithm can make this distinction efficiently and accurately provided there are no more than three times as many noise dots as there are dots in the line.

The last group of papers concern themselves with making the management of graphics systems and programs simpler and more functional. Moltedo and Noferini describe their technique for encoding and decoding graphical data in ways which facilitate efficient and accurate transfer between different graphics systems. The ways in which congenial graphical user interfaces can be developed are looked at in the paper by Mullarney and Neelamkavil. They describe a prototype system and methodology aimed at making the construction of graphical user interfaces simpler, more easily reconfigurable and separable from the applications which they serve. Finally, Bilideau and Laguitton outline some methods of providing a given commercial FORTRAN compiler for microcomputers with useful screen management subroutines. They evaluate the methods from a number of standpoints and conclude that ROM BIOS services made available by means of assembler and direct video programming give the best results.

All these papers offer us new insights into some of the problems of computer graphics and present us with innovative solutions to them. I have no doubt that anyone having an interest in the applications, methods, techniques and algorithms of computer graphics will find much of value here.

JOHN LANSDOWN