Not only computing

Although computer art is often thought of as largely to do with pictures, computers are probably far more used in music than in the visual arts. Nowadays almost no-one would think of creating a piece of electronic music, either for classical or for pop use, unless they had a computer to help them.

JOHN LANSDOWN

Until comparatively recently, however, if you wanted to make a piece of computer music you had to use a fairly large machine either to synthesise the sounds by digital means or to control an external (usually analogue) synthesiser.

Music for all

All this changed with the advent of the low-priced digital keyboard synthesiser and the MIDI interface. MIDI stands for Musical Instrument Digital Interface, an international communication standard which came to the fore in the 1980s, and most personal computers and many digital synthesisers now have MIDI connectors. Using the interface a personal computer can control banks of electronic musical instruments allowing you to compose new, or play existing, music in a way not previously feasible for non-professionals.

You can program the MIDI directly if you are so inclined. Indeed, if you want to compose algorithmically, this is probably the only option open to you. But if you want to compose or play more conventionally via standard musical notation, there is a great deal of excellent software already available and more is coming onto the market all the time. Despite its having one or two drawbacks, my favourite package at the moment is one for the Macintosh called the Deluxe Music Construction Set. This allows you to create Macintosh sounds or to control external synthesisers and have them play music set up on the screen. You can see from Figure 1 (below) that you can input the musical instructions by choosing notes on the keyboard or by entering them on the musical staves (or a combination of both). The sound of the individual notes are displayed as you enter them by either method but, of course, the main advantage is that the computer stores all the input and can play it back at any speed. Because the interface naturally allows two-way communication, in addition to all this, you can play a piece of music conventionally on your MIDI-connected keyboard, and then edit out mistakes and infelicities for a perfect performance on playback!

Asking for more

About twenty years ago at an Online Computer Graphics Conference, I listed out some of the things that artists might want from computing. One of these was an economical facility to print musical notation (both conventional and composer-specific). A rather pompous professor from Canada told me that as one of his students had already patented a method of doing this, I had not kept up to date on the state of the art in the subject. I don't know what happened to the student's patent but it seems to me that it is only in the last couple of years that it has been economically possible to put together scores by personal computer, and Figure 2 shows an example of part of one.

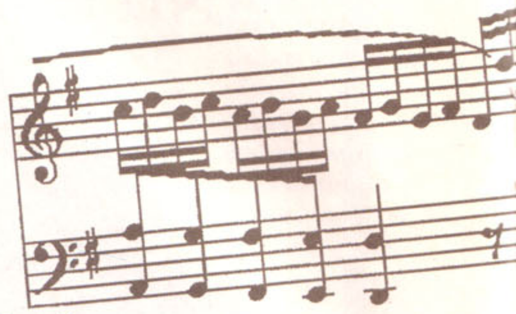
Figure 2

One day I must look again at the paper I gave at that conference and see what else I asked for, and how close the computing profession and industry is to providing us with it.

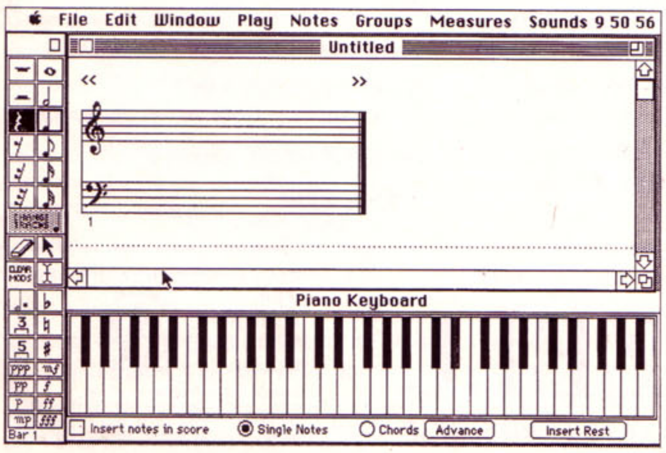

Figure 1

The fifth cavalry

Anyone who has had to program a sequence of graphic images for things like computer animation or graphical interaction must surely have occasionally wished for a handy text that would describe a fast method of approximating 3-D Euclidean distance or computing surface normals and other such items that have to be performed many times over before images can be displayed. Which of you, writing a ray-tracing program, hasn't yearned for a fast ray-box or ray-polygon intersection method? Who, digitising a 2-D shape, hasn't wanted a good ready-made algorithm for automatically fitting curves to the points created?

Well, help is at hand. It comes in the form of a new beautifully presented 833-page book edited by Andrew Glassner called *Graphics Gems* (published by Academic Press, 1990, price $49.95). This excellent volume collects together graphics algorithms or short tutorials from around fifty contributors. Most of the algorithms are presented in both pseudo-code and as C language procedures. Those I've tried seem to

— also art

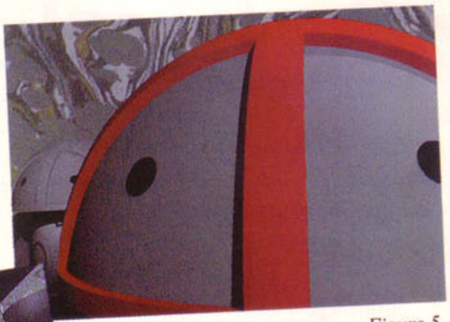

Figure 5

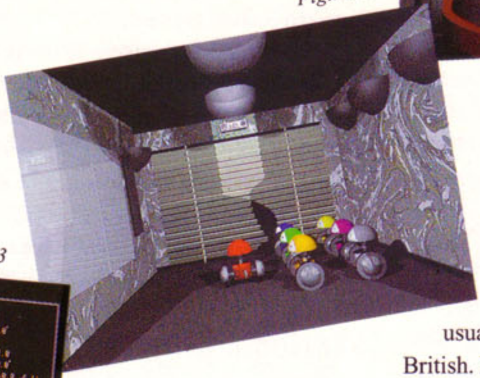

Figure 4

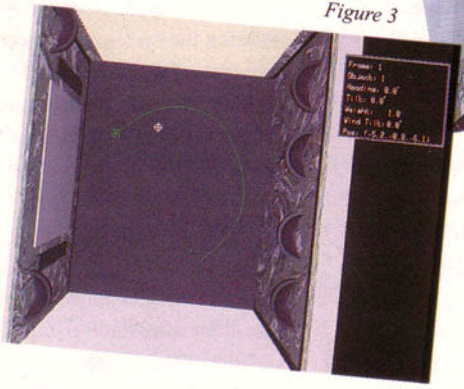

Figure 3

arrives

work as written, and I haven't spotted any mistakes. (Although, incidentally, I do not agree with a statement on page 287 by Gervautz and Purhathofer that 'the human eye is able to distinguish about 200 intensity levels of the three primaries red, green and blue. All in all, up to ten million different colours can be distinguished.' Because of the differences in our sensitivity to the different primaries, I think this figure overestimates our ability to distinguish colour differences by a large factor. However, this quibble does not detract from the usefulness of the algorithm that Gervautz and Purhathofer provide.)

This book should not only be at the elbow of every graphics programmer but, because of its compact and informative tutorials on such things as quarternions and pixel transforms, should also be essential reading for graphics programming students. Andrew Glassner calls for further contributions which might be incorporated into another volume at a later date. Get the book, and send him your favourite algorithms.

Actors, animators and doctors

When my colleagues and I were doing computer animation for TV and films in prehistoric times — that is, the mid 1970s to early 1980s — virtually every scene, every object and movement of the object or the synthetic camera had to be specially programmed. Of course, we were able to re-use a lot of code from one job to another and, mostly, the objects we had to deal with were fairly simple wire-framed things. But, despite this, we never seemed to have enough time to set up a generalised system that could be used to put objects in a setting and move them about, either interactively or by script as one would with live actors (although we did use the concept of actors and cameras in some of our programs).

Fortunately for all of us in computing, however, indefatigably beavering away in the background in universities and polytechnics all over the world are those most dedicated of people, PhD students. They usually have the time, skill, application and, sometimes even funding, to tackle problems that industry can't usually break off from production to deal with. So nowadays anyone doing commercial computer animation can call on a number of well-configured animation packages to help them. These are often based on the results of PhD student research though alas, more usually American or Japanese than British. Recently, however, I examined the work of a PhD student, Nigel John, who was in Professor Philip Willis' Department in the University of Bath. As part of his successful Doctorate work Nigel had devised an animation system called 'Controller', which would have made my life much less difficult a few years back and, even now, would be enormously valuable to any computer animation production house.

In this system the creation, positioning and movement of all the elements of a scene is done interactively. The paths that the actors or the camera are to take can be shown on a plan of the 'studio floor' (Figure 3) making it simple to determine how frequently shots should be taken and in which direction the camera should be pointing for each. Figure 4 shows a scene from the short movie employing robot-like actors Nigel created using this technique. The faces of two of them are shown close-up in Figure 5. Getting close-ups right in the old days used to take a great deal of trial and error. Interactive working as exemplified by the Controller system makes the job a pleasure. It leaves time for animators to concentrate on giving a scene meaning rather than having them distracted by technological problem-solving.

Getting close-ups right in the old days used to take a great deal of trial and error. Interactive working as exemplified by the Controller system makes the job a pleasure . . .