Not only computing – also art

JOHN LANSDOWN

God is not dead, but his patents have lapsed

Until I recently caught sight of the intriguing title of a new book at the Science Reference Library, I had not realised that genetic engineers were sharing one of the problems of software engineers. It appears that they too, are trying to find ways of protecting their intellectual property. We, in computing, are aware of the problems of copyrighting or patenting our algorithms and programs and, on the face of it, what seems to be a simple matter has turned out to be full if unexpected pitfalls.

Why there should be a problem, I do not know. Most of us who write software products would regard these as just as 'real' an invention as items of hardware but the law has not seemed altogether able to support this view. Those dealing with cell modification are also having some trouble in convincing legal authorities that the new cells they create are patentable.

Biotechnology is now big business and properly modified DNA molecules can be immensely valuable. It is not surprising, therefore, that companies who make these modifications should seek to protect their investments. However, granting patents on living organisms does seem to be a distinctly odd idea and obviously requires considerable thought.

The whole matter will impinge more directly on computing when organic cells start to be used for computer storage or even processing, so it is not something we can afford to ignore. The book, incidentally, is called *Patenting of Life Forms* (Banbury Report 10: Edited by Plant, Reiners and Zinder: Cold Spring Harbor Lab 1982). Dr Frankenstein, you will note, is not one of the contributors.

Fractals revisited

On one or two occasions in these columns in the past, I have remarked on the problems of adequately representing natural objects such as trees, mountains, stones, coastlines and so on in computer form. The most obvious difficulty in storing the enormous amount of data that such irregular objects normally need for their description. A few years ago, an IBM Research Fellow, Benoit Mandelbrot, showed that it was possible to simulate many of these seemingly random shapes by means of special mathematical elements which he called 'Fractals'. He did this in a fascinating book, *Fractals: Form Chance and Dimension* (W. H. Freeman and Co., San Francisco 1977), which contained a large number of computer-generated drawings of natural irregular shapes such as islands and mountains, as well as regular, mathematical, space-filling curves such as Peano and snowflake.

I say the book is fascinating, but not that it is either easy to read or, at least to someone of my intelligence, easy to understand. Indeed, although I felt I could almost grasp its underlying ideas, I could not deduce the steps needed to actually program Fractal drawings. Judging by recent correspondence in the ACM literature, it now seems that others too, and even those who managed to take things much further than I was able to do, have also not fully understood how Mandelbrot achieved his pictures. One of the fundamental features of a Fractal curve is its 'self-similarity'. By this is meant that, if a regular Fractal curve is divided into sections, then its parts can be exactly superimposed on one another. Mandelbrot suggested that irregular natural objects also display self-similarity. A map of a coastline, say, could be cut into sections and the parts would more or less match if superimposed. In addition, mountains are self-similar in that they look roughly the same at whatever distance they are viewed. In the case of such objects, of course, the self-similarity would be statistical rather than exact, in the sense that their 'averages' and 'variances' would be the same. Mandelbrot believes that, if we are to make convincing drawings of such things as trees, clouds and smoke, then the curves we use should display the same statistical properties.

A year or two ago, computer graphics workers began producing their own versions of Fractal pictures of landscapes, and very striking they are. In the June 1982 issue of the *Communications of the ACM*, Alan Fournier, Don Fussell and Loren Carpenter described how they achieve their results and published Pascal algorithms which we could all understand. In an Editor's note at the bottom of the first page of their paper, however, we were told that, 'B. Mandelbrot, on whose work this paper is based, has raised certain objections . . . '. Rumours were that, indeed, he was hopping mad.

In the August 1982 issue, Mandelbrot gave his comments which seem to indicate that Fournier *et al.* are not using Fractals at all! This is because their curves lack the principle of self-similarity. There is no doubt that the drawings of Fournier and his co-authors are much less realistic than Mandelbrot's but, as Fournier points out, they were concerned with creating reasonable simulations with the minimum amount of computer time. Mandelbrot regally dismisses such considerations and says the quality of the simulation is what matters.

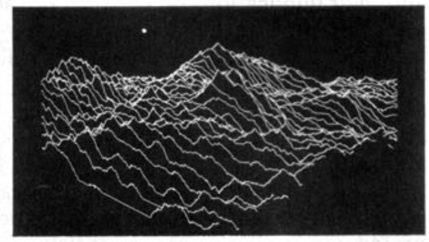

Figure 1

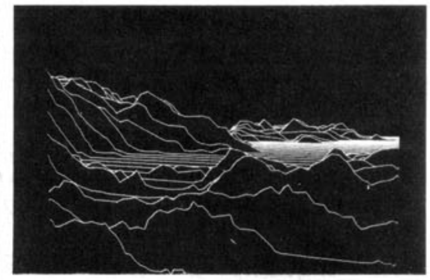

Figure 2

Figure 3

Meanwhile, others are beginning to use Fractals (or, we had better say, Pseudo-Fractals) in their drawings. Paul Brown and Chris Briscoe of Digital Pictures, for example, have produced the attractive landscapes shown in Figures 1 and 2, and Brian Wyvill and his students at Galgary, the recursively created tree of Figure 3.

There is, incidentally, another possible form of self-similarity. Art works display it. Musical form, for instance, embraces not only the overall structure of a work but also its details, so that a well-constructed composition is all-of-a-piece; its parts accurately reflecting the whole. Good architecture certainly displays the principle and, by just looking at a detail of a well-designed classical building, we know what the whole is like. Mandelbrot has done us a service in drawing our attention to Fractals – his book should be more widely read.