Not only computing—also art

JOHN LANSDOWN

Since Christmas I seem to have done little else but digitise images of various sorts for subsequent processing—mainly frame-by-frame animation. As a result of this unenviable experience I conclude that we must go a long way yet before the task of getting complex visual information into computers is the simple, trouble free matter it ought to be. Correctly inputting two-dimensional (and two-and-a-half dimensional) objects is difficult enough particularly if, like lettering, they contain a lot of arbitrary curves—the eye can tolerate quite large errors in the shapes of letters consisting of straight lines such as A and H, but minute point misplacements in letters like O and S stand out like beacons. However, three-dimensional objects are the real headaches.

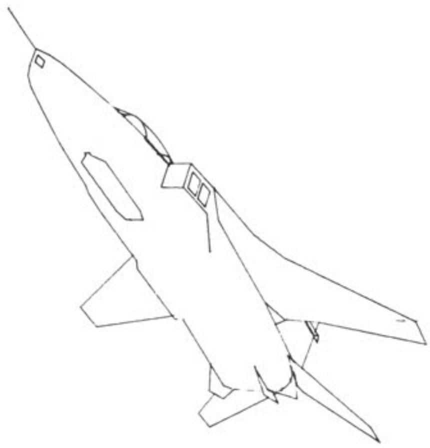
The problem with three-dimensional objects, such as Figure 1, is that in order to take advantage of the semi-automatic facilities of digitisers and tablets, we must have *a priori* fairly complex mathematical models of objects in the machine. The only model we need in the two-dimensional case is that some points begin a continuous line and others are part of one but, for three dimensions, much more information is needed. If we know that the object consists of rectangular prisms then, again, a simple model is all that's required but objects like the Sepecat Jaguar of Figure 1 are not amenable to simple treatment.

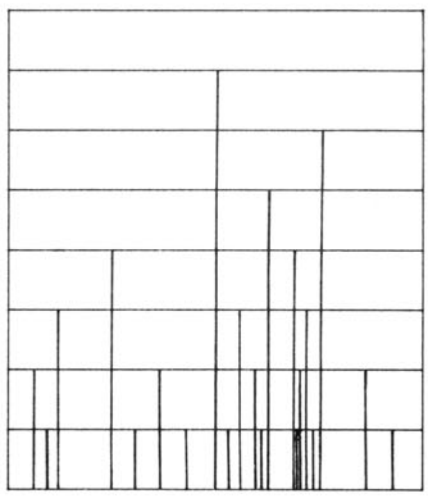
Ian Braid, in his book *Designing with Volumes* and his paper 'The Synthesis of Solids Bounded by Many Faces' (*Communications of* ACM, April 1975 pp 209, 216), shows how objects built up of cubes and cylinders can be fairly described by, as it were, adding and subtracting volumes in a special way. I would like to see this work further developed and more algorithms published so that we can all experiment with the ideas of synthesising objects from primitives rather than trying to describe them more explicitly.

Skip and divide

Some years ago Alan Sutcliffe was looking for a graphics algorithm which, though simple nevertheless produced an interesting effect. I have mentioned on another occasion the resulting Skip and Divide algorithm which, applied to a set of intervals along a line, goes like this:
Move along the line alternatively skipping an interval and dividing the next into two equal intervals. Treat the end of the line as joined to the beginning so that the process can continue as long as needed.
There are many ways to display the results. Figure 2 shows the first seven cycles, beginning with a single interval. As the process continues, something like a spectrum analysis ensues. Each division of an interval produces two equal intervals and it is clear that, at the next cycle, one of these will be further divided and one skipped. At a Computer Arts Society course at the Electronic Music Studio at Stockholm some years ago, Alan mentioned the algorithm to Lambert Meertens, himself a prolific computer artist and mathematician who works at the Mathematical Centre in Amsterdam. The next day Lambert came back with a proof that, whilst a particular interval may remain undivided for several cycles, no interval will remain undivided indefinitely.

More recently, Alan came up with a simple way of applying the algorithm to an area rather than a line and the resulting sequence of events is shown in Figure 3 starting with a double square. The rule is that a square is divided vertically into two double squares, and a double square is divided horizontally to make two squares. Left is taken before right and lower before upper. Figure 4 shows the result when, at the end of each cycle, areas are alternatively shaded and left blank. The result from several cycles are concatenated to form a continuous pattern.





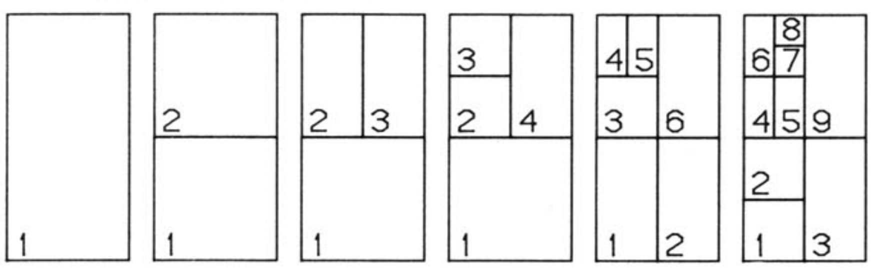


Figure 1 (top)
Figure 2 (centre)
Figure 3 (right)
Figure 4 (below)