Not only computing — also art

JOHN LANSDOWN

Over the last fifteen years or so I have not been in a position to do much creative work in an area which interests me a great deal, although it's one in which I seem to be the only proponent. This is the use of computing to choreograph dances.

There are two reasons for this inactivity. One, lack of time and, two, lack of a dance group interested in performing works choreographed in this way. Time, compounded by incipient laziness, is still a problem (indeed, more of a problem now than ever before). Also until this June, I had no-one to choreograph dances for. But the other day, out of the blue, I received a fax from a dancer, Kai Tai Chan — with whom I co-directed a dance group in the 1970s — asking me if I'd do something for his company's repertoire for next season. The joy of receiving a commission by fax is only exceeded by receiving unsolicited cheques through the post and I readily accepted — particularly as Kai Tai now runs Australia's leading modern dance company, The One Extra Group, and has an international reputation as a dancer and innovative choreographer.

On with the dance

I first began work in this field in 1968, creating a piece for the Computer Arts Society's first exhibition, Event One, at the Royal College of Art. The big problem was deciding the way in which the computer would present its instructions to the dancers. At the time of Event One, the results were output by the computer as a script in Benesh notation. This is a dance notation written on five-line musical paper (Figure 1) which sets down the position and movement of the dancer's limbs. However, though the notation is very popular as a way of recording and reconstructing dances and is employed by many of the great ballet companies, it is normally used through the medium of a 'choreologist' who is a person skilled in the writing and interpretation of the symbols. There are a number of different dance notations each with their own merits but dancers themselves — particularly modern dancers — are not very familiar with them. Thus, I soon abandoned the notation approach in favour of something which dancers could interpret more readily. After a number of experiments and false starts, I settled on getting the computer to output more or less pictorial, comic strip-like images which could easily be read by anyone after a few moments instruction. Firstly, these were in a stick-figure format (Figure 2). Later, I used more pictorial outline images (Figure 3).

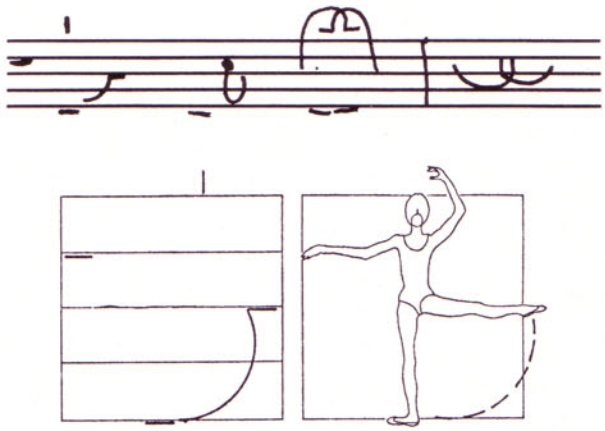

Figure 1

(top) how the notation works
(bottom) a movement square

These images show the key points in the movement to the dancers, and they have to compose the linking actions to get from one position to another in the time that the computer allows. (The generating program calculates that this is possible before choosing the positions). It should be noted that the pictures do not

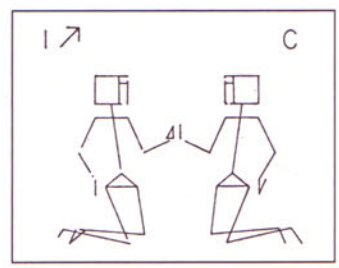

Figure 2

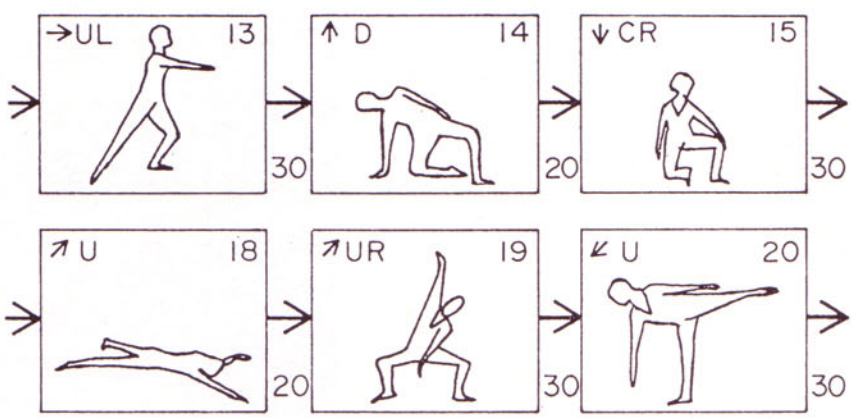

Figure 3

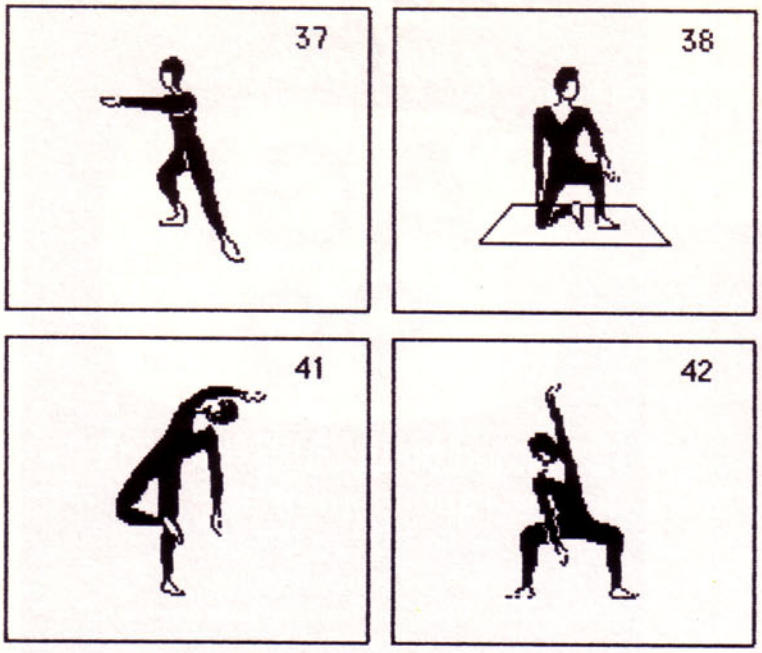

Figure 4

necessarily illustrate the dance as seen from the point of view of the audience: the direction of facing and stage position are also included in the script, and these modify the appearance of the dance. Because of developments in raster graphics since the seventies, I am now able to output the scripts in an even more pictorial way and the new dance will exploit this fact (Figure 4).

Why, I think I hear you ask, use a computer to generate that most human of art forms, dance? Well, primarily because I am intrerested in modelling (or simulating, if you like) human creativity in order to understand more about it. Further, because the method I use actively involves the dancers in the creative process: they do not just interpret, they also compose and their technique and ability comes into play in a manner analogous to that of more conventional ways of choreography.

There is one other reason, too. Dancers are highly programmed beings. The training and discipline of dance tends to give them semi-automatic responses and stylised movements. The unexpected and sometimes bizarre configurations that the computer employs challenges this pre-programming and can lead to stunning results. I hope that the new dance might bring forth one of these occasions.

The grand tour (1990s style)

One of the Master's degree courses on which I teach at Middlesex Polytechnic involves students moving about computer graphics centres in various places in Europe, picking up credits as they go. Recently, a group from the University of Paris were with us and two of them, Beatrice Selleron and Remi Bichet, contrived what they call the 'Imaginary City' to illustrate what they had learned about lighting and rendering. This city is constructed by the computer according to certain rules. For example, the amount of light falling on an object (which is invisible) causes certain forms of building to grow: a spherical one if the light is at a maximum, a rectilinear one if the light is at a minimum, a conical one for lighting of average intensity. Other rules determine transparency, height, colour and so on (Figures 5, 6 and 7). Thus, rather than choosing to show their understanding of lighting techniques by simply creating a particular object photorealistically rendered, they have done so by inventing a new world in which the techniques have more fundamental purposes. They were not with us long enough to create a movie of this fantastic place, but work with this promise gives me great hope for the future of computer graphics.

Figure 5

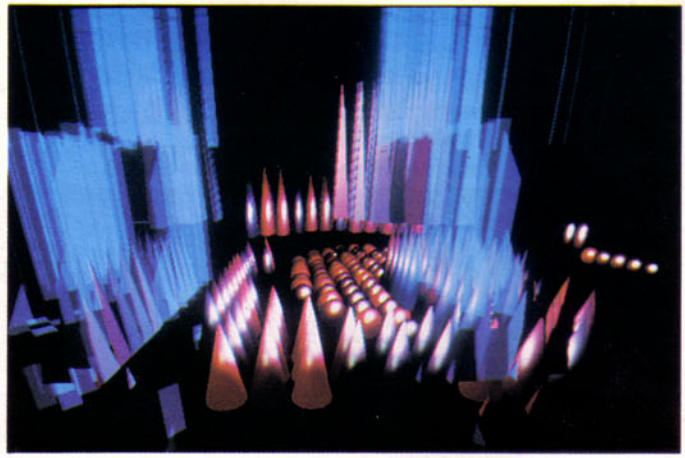

Figure 6

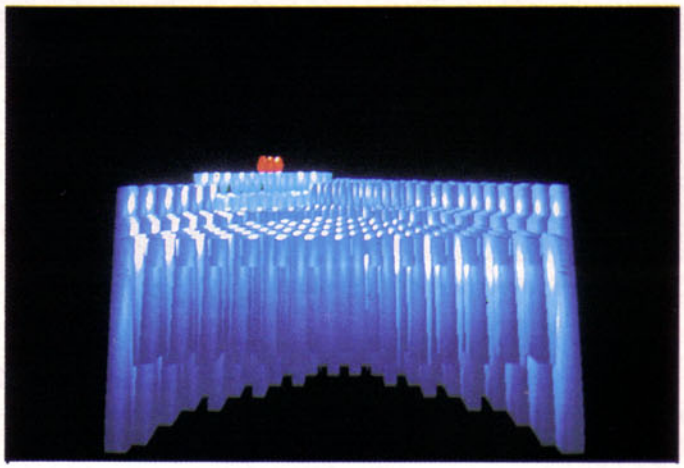

Figure 7