MOLECULAR GRAPHICS USING THE DAP

Rod Hubbard and David Fincham

We have recently been exploring the use of the DAP in drawing realistic pictures of molecules. By 'realistic' we mean the type of picture often known as 'shaded space-filling'. An example appears in figure 1, though it is unlikely that the reproduction does full justice to the original which is a colour transparency. These pictures have the following features. The individual atoms are represented by spheres of the appropriate van der Waal's radius; and are coloured according to atom type or some other relevant criterion. The spheres are shaded as if illuminated by a point source, to give an impression of shape; full removal of hidden surfaces is carried out.

Such computer-generated pictures are replacing laborious model building as a way to display the structures obtained by crystallographic techniques. This is more than just an exercise in producing pretty pictures as, particularly for the large molecules of biochemical interest, the chemical behaviour of the molecule is very much dependent on its shape and its surface topography. The space-filling picture then becomes an important research tool in studying biochemical activity.

At present these pictures can be produced by raster-graphics terminals which are usually driven by mini-computers. However, this is a slow process. For example the insulin dimer shown in figure 1 would take several minutes to draw because of the extensive computations involved. There are two aspects to the computations; the removal of the hidden surfaces, including the calculations of the correct intersections between the parts of the atoms that are visible, and the shading to get the effect of illumination, involving the specification of an intensity value for each point in the picture. Typically raster pictures consist of a 512x512 array of pixels (picture elements) so there are over a quarter of a million values to be specified. The reader will begin to appreciate why the most powerful Cray computer has been sold to a Hollywood movie company! However, this is a mis-use of such a machine. For most purposes computer-generated raster pictures need no more than 8 bit data values for each pixel, allowing 256 distinct shades, and the use of a 64 bit floating-point processor to calculate them is a real case of overkill.

The potential and suitability of the DAP in tackling this problem should be obvious to all regular readers of this Newsletter. First, it consists of a square 64 by 64 array of processing elements working in parallel, so it is possible to process 4096 pixels simultaneously. Second, the processing elements can perform arithmetic at any required precision, and the lower the precision the faster the machine goes.

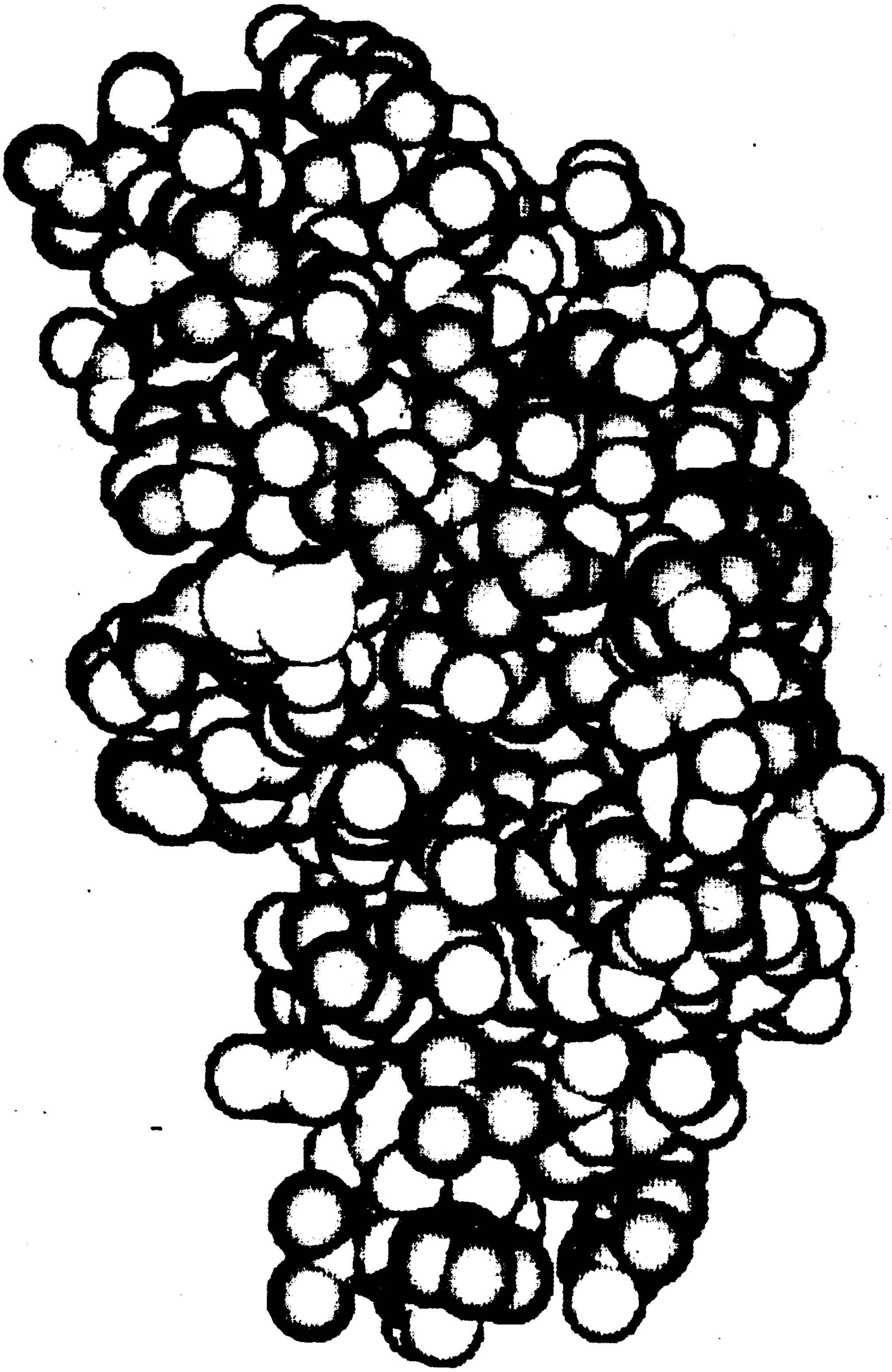
Our approach is the following. The complete picture of 512 by 512 pixels is divided into blocks of 64 by 64. For each block two DAP matrices are stored, an INTEGER*1 matrix of colour levels (i.e 8 bits) and an INTEGER*2 matrix of z-values, the z direction being perpendicular to the plane of the picture.

For each distinct type of atom in the picture to be displayed we pre-compute a correctly shaded sphere within a similar 64 by 64 block, but outside the main picture store. The DAP Fortran routine to do this is shown in figure 2. Each pixel is given an x and y index (with the origin at the centre of the block) and the z value of each pixel, $SQRT(R^2-(x^2 + y^2))$, is calculated. Also for each pixel an intensity level between 0 and 15 is defined; for a sphere illuminated from the front the intensity is simply proportional to z.

To draw the picture we loop over atoms one by one. A simplified version of the code which adds an atom into the picture is shown in figure 3. A copy of the previously computed sphere of appropriate radius is simply shifted, using built-in DAP Fortran shift functions, so that it occupies the correct position ready for copying into the appropriate picture block. (Extra code is needed to handle the case where the atom overlaps boundaries between picture blocks). Both the intensities and the z values are shifted, the colour bits are added into the intensities, and the z value of the atom centre is added to the sphere values. Where the resulting z values of the atom are greater than those already existing in the picture (i.e. a part of the new atom lies in front of the picture that has been created so far), the appropriate picture intensity and z values are updated with the atom values. This simple technique determines the correct intersections between the spheres without any geometrical calculations being required.

We have indeed found that the DAP is extremely effective in producing molecular pictures. The picture of insulin shown, which has about 900 atoms, took only half a second to calculate, and this is before any attempts to optimise the program. Smaller molecules are correspondingly quicker to compute. This opens up two exciting possibilities.

First, it should be possible to produce motion pictures to study dynamical processes. Members of CCP5 are mostly interested in systems of many simple molecules, and here the application would be to study collective dynamical effects of a complicated nature which it is difficult to characterise in terms of simple response functions. Another group of people who could make use of such movies are the protein chemists. It is being increasingly realised that dynamical properties are crucial to many aspects of biological function, and the production of a movie is an unparalleled aid to understanding the behaviour of a simulated model system. Very few such movies have been made to date because of the slowness of production of individual frames on conventional systems. Now that we have demonstrated the ability of the DAP to perform these calculations we are trying to obtain funding to attach a high speed graphical output and recording system to facilitate movie-making.



Second, it is well known that ICL have plans to develop a smaller DAP in an interactive workstation environment. This work has demonstrated that it should be possible to produce real-time molecular modelling systems involving both graphical display of space-filling pictures with the necessary computational back-up such as energy minimisation calculations.

```
      SUBROUTINE SPHEREGEN(IRAD,SPHEREZ,SPHEREI)
      INTEGER*1 IRAD,SPHEREZ(,),SPHEREI(,)
C
C this subroutine generates a 64*64 pixel shaded sphere of
C radius IRAD
C
      REAL*3 Z(,)
      INTEGER*2 SQUARES(,),INDEXVEC2(),IRADSQ
      INTEGER INDEXVEC()
      LOGICAL SPHERE(,)
C
      IRADSQ=LENGTH(IRAD,2)**2
      CALL X05SHORTINDEX(INDEXVEC,-31)
      INDEXVEC2=INDEXVEC**2
      SQUARES=MATC(INDEXVEC2)+MATR(INDEXVEC2)
      SPHERE=SQUARES.LT.IRADSQ
      Z=0.0
      Z(SPHERE)=SQRT(FLOAT(LENGTH(IRADSQ-SQUARES,3)))
      SPHEREI=0
      SPHEREI(SPHERE)=(16.0*Z/IRAD)-0.5
      SPHEREZ=Z+0.5
      TRACE 5(SPHEREI)
      RETURN
      END
```

Figure 2

```
      SUBROUTINE ADDATOM(IB,JB,MOVEX,MOVEY,ICOL,ZCENTRE
     1                   ,SPHEREZ,SPHEREI
     2                   ,PICTUREZ,PICTUREI)
      INTEGER IB,JB,MOVEX,MOVEY
      INTEGER*1 ICOL
      INTEGER*2 ZCENTRE
      INTEGER*1 SPHEREZ(,),SPHEREI(,)
      INTEGER*2 PICTUREZ(,,8,8)
      INTEGER*1 PICTUREI(,,8,8)
C
C This subroutine adds at atom into the picture. Its centre is in
C the block (IB,JB). The previously generated sphere needs to be
C shifted by (MOVEX,MOVEY) pixels. This simplified version does
C not handle atoms going over block boundaries, nor does it handle
C negative shifts.
C
      INTEGER*2 TEMPZ(,)
      INTEGER*1 TEMPI(,)
      LOGICAL SPHERE(,),MASK(,)
C
      SPHERE=SPHEREI.NE.0
C
      TEMPI=SHSC(SHEC(SPHEREI,MOVEX),MOVEY)+ICOL
      TEMPZ=SHSC(SHEC(SPHEREZ,MOVEX),MOVEY)+ZCENTRE
C
      MASK=TEMPZ.GT.PICTUREZ(,,IB,JB)
      PICTUREZ(MASK,IB,JB)=TEMPZ
      PICTUREI(MASK,IB,JB)=TEMPI
C
      RETURN
      END
```

Figure 3