Display algorithm for space-filling molecular models using a video array processor

Peter Schultze and Kurt Wüthrich

Institut für Molekularbiologie und Biophysik, Eidgenössische Technische Hochschule, Hönggerberg, CH-8093 Zürich, Switzerland

A new algorithm for the generation of coloured, space-filling molecular models is described. It relies on the parallel processing facilities of a DeAnza IP8500 raster graphics system to increase the speed of display generation. Using suitable storage of the information about the type as well as the colour of each atom, it becomes possible to switch rapidly between various schemes of colouring. Results are shown with DNA and protein molecules.

Keywords:

received 17 June 1985, revised 9 August 1985

The authors' laboratory is mainly concerned with studies of the conformation and dynamics of biopolymers in solution. For the presentation of the results, graphical representation plays an important role. The authors use two different types of computer graphics devices. The first is a highly interactive Evans and Sutherland MPS vector graphics system that allows 3D manipulations of line drawings in real time. A program package that uses this system to handle protein structures has been described previously[1]. The other is a DeAnza IP8500 raster image processor for the generation of space-filling molecular models, which allow inspection of the molecular surface. For a raster image of a molecular model consisting of shaded atomic spheres, intensity values must be computed for each pixel. Although recent developments have led to systems (for example the Megatek Merlin) which are capable of moving space-filling models of small molecules in real time, the present hardware does not allow continuous motions of macromolecules. Therefore, an image has to be recomputed each time a new viewing angle is desired. This task can in principle be achieved with different techniques. Filled polygon algorithms[2] belong to a class of methods which are generally applicable to the modelling of arbitrary surfaces and have been implemented on some modern raster graphics systems. Alternatively, molecular models consisting of atomic spheres have been visualized with the use of a scan-line algorithm[3]. With this technique the image is computed by the host-computer and sent line by line to the raster system, which is thus used only as a display screen. In this paper a fundamentally different algorithm is described which results in a faster build-up of images. It makes use of the IP8500 array processor which is capable of executing arithmetic and logical operations on whole image frame-buffers without direct involvement of the host-computer. The program was designed to allow flexible colouring schemes and to accept the same type of coordinate files as the vector-graphics program used to drive the Evans and Sutherland MPS. Additional features are the rotation of the molecular model about three orthogonal axes, successive display of two views with appropriately different viewing angles for stereoscopic slides, generation of a 'rocking motion' of the molecule to improve the 3D impression, display of models with atomic radii different from the normally used Van der Waals radii, and addition of lettering to the image.

HARDWARE

The Gould DeAnza IP8500 raster graphics system used in this study is connected to a DEC PDP11/34 as host computer. The vector graphics system is controlled by the same host, which allow easy communication between the different programs used to drive the two devices. The IP8500 configuration contains four image frames. Each of these memory buffers is capable of storing 512 × 512 pixel values, with 8 bit/point. Any three buffers can be combined under software control to form a 2^{24} colour image, where 2^8 intensity levels are given separately for each primary colour by the corresponding buffer. With an alternative mode of operation one buffer alone can store a 2^8 colour image. Here each eight-bit value represents an index to three look-up tables which contain the actual intensities for the red, green and blue components as defined by the user's program. Since the algorithm requires three invisible buffers to store auxiliary information, the latter method was used, even though this reduces the number of simultaneously visible colours from 2^{24} to 2^8. In addition to its image memory the system is equipped with an array-processor, which executes arithmetic and logical operations on whole buffers within 1/40 s, i.e. the refresh period of the screen. The buffers can be connected to the array processor under program control via up to 10 input channels. The contents are then routed through a pipeline of up to six selectable processing steps and transformation tables for various arithmetic and logical operations on entire buffers or selected regions of pixels. The result after the first processing step in two of the ten input channels

 0263-7855/86/020108-04 $03.00

can be used for selecting the type of operation to be applied to each pixel-position at the second step. The output of the array-processor can be directed via four channels to any of the buffers, including those already connected to input channels.

ALGORITHM

In the following description A,B,C and D refer to the buffers (see Figure 1). A_{xy}, B_{xy}, C_{xy} and D_{xy} designate values stored at the position (x,y) in the corresponding buffer. The coordinate system is related to the screen corrdinates as follows: x-axis: horizontal screen direction, y-axis: vertical screen direction, z-axis: depth of image.

Image generation

The image is generated using the following procedure:

1) Read atom coordinates from disc, assign atom-type codes, scale and centre to screen dimensions.
2) Sort atoms according to ascending z-values.
3) For each atom-type write a hemisphere with its particular Van der Waals radius to buffer A.
4) Scroll buffer A in the x- and y-directions to position the selected sphere and set window coordinates in order to exclude all other spheres from being processed.
5) If the z-coordinate of the atom considered lies within the range of the z-buffer (buffer B) go to step 6; otherwise reset z-buffer.
6) For every pixel with coordinates (x,y) within the window, for which $A_{xy} + z > B_{xy}$ and $A_{xy} > 0$, enact the following transfers:

$A_{xy} + z \longrightarrow B_{xy}$ (z-buffer)
Norm $(A_{xy}) \longrightarrow C_{xy}$ (visible buffer)
atom type code $\longrightarrow D_{xy}$

7) If there are further atoms go to step 4; otherwise stop.

The z-buffer can only store values up to 255. The z-coordinates are scaled by the same factor as the x- and y-axes, namely up to values of 511. A decrease of the resolution in the z-direction is not desirable, because it would lead to more jagged intersecting curves between spheres. Therefore, steps 2 and 5 were included to prevent overflow. When the upper limit of z is reached, the z-buffer is reset using step 5 by a single array-processor operation, which transforms all values

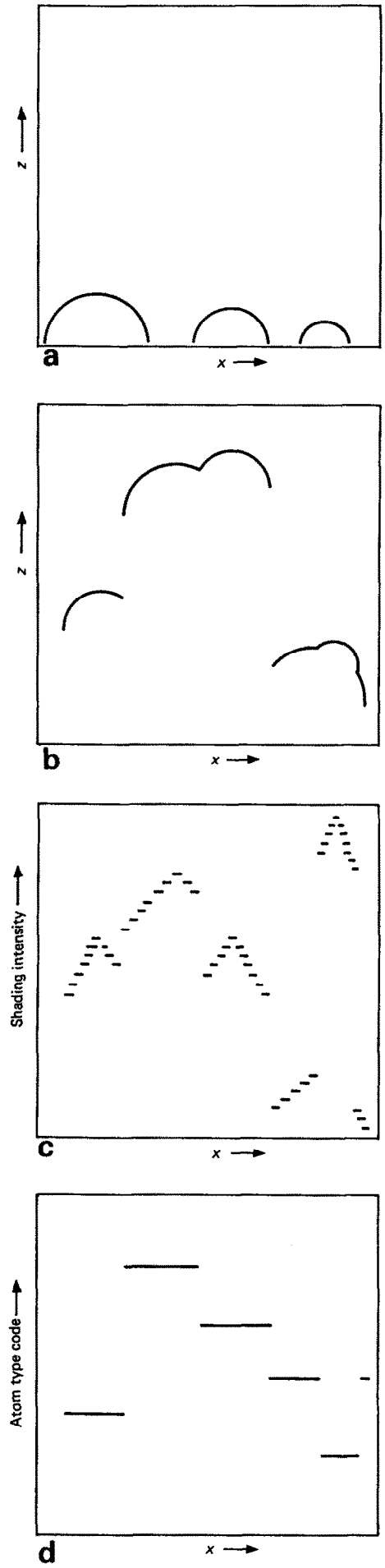

Figure 1. (right) Schematic cross-sections through the four memory buffers at an arbitrary time during the display process. The horizontal axis corresponds to the x-direction of the screen and the memory buffers. It represents one half of a horizontal row of 512 pixels at the same y-position in each buffer. On the vertical axis the contents of the memory cells from each buffer are plotted as follows: (a) hemispheres, with different radii, corresponding to the different atom types. The spheres are computed by the host-computer and written to this buffer before the display-process is started. The sphere in the centre is at the position where the transfer of the last processed atom took place. (b) z-buffer, which contains the visible surfaces of the already processed atoms. (c) visible buffer containing the actual display information. Its content consists of linear shading values for atoms in different colours, but does not show the depth information. (d) contains the atom-type code on all visible pixels of each atom

between 1 and $(255-r_{max})$ to zero and subtracts $(255-r_{max})$ from higher values (r_{max} is the radius of the largest atom). No information is lost in this operation because all values whch are zeroed belong to previously drawn atoms, which stay visible in C unless overwritten by further atoms with larger z-values. Note that if in the z-buffer more than eight bit/point were permitted by the hardware, the atoms could be processed at no loss of image quality without previous sorting.

In step 6 the values in the z-buffer (see Figure 1(b)) are compared with the sum of the z-values of the selected sphere in buffer A and the z-coordinate of the new atom. All transfers to buffers B,C and D take place only at those pixel positions where the z-buffer values are lower than this sum and only within a quadratic window enclosing the desired sphere. The whole of step 6 is carried out by a single array processor operation within one refresh-period. The addition of each new atom to the image by steps 4–6 does not take longer than 1/20 s. The function Norm (A_{xy}) is generated via an internal 'look-up-table', which transforms the actual spherical z-values from buffer A to a linear scale with a limited range of shading intensity numbers (see later), which are then stored in buffer C.

Change of colouring

1) Transfer C_{xy} − Assign $(D_{xy}) \rightarrow C_{xy}$
2) Load new look-up-table with the altered colour assignments.
3) Transfer C_{xy} + Assign $(D_{xy}) \rightarrow C_{xy}$

In step 6 of the image generation, the geometric z-values of each sphere were normalized to a linear scale of shading intensity numbers (buffer C) representing indices to the look-up-tables which control the display by giving the actual values for the red, green and blue intensity components at every pixel on the colour screen. In buffer D an atom type code ranging from 1–255 is stored on those pixels that are not hidden by other atoms. This allows the differentiation of up to 255 different atom types. Thus e.g. all sidechain- and backbone atoms in each of the 20 standard amino acids can be distinguished. To each atom type a colour code in the range from 1–15 is assigned via a look-up-table represented by the function Assign (D_{xy}). The colour codes are multiplied by the maximal shading intensity number before they are loaded to the look-up-table. The maximal shading intensity number, 17, and the maximal colour code, 15, were chosen so that their product does not exceed 255.

To change the colouring scheme, the colours are removed from the image by subtraction of the colour codes from buffer C in step 1. Subsequently, new colour assignments are written into the look-up-table. In step 3 the new colour codes, which are generated by transformation of the atom type codes from buffer D with the altered look-up-table, are added to buffer C. The symmetrical shading of the spheres thus has to be computed only once to set up the display look-up-tables and not for each individual atom. Furthermore the colouring scheme can be changed without recomputation of the image.

RESULTS AND DISCUSSION

The algorithm has been coded as a FORTRAN IV program on a PDP11/34 computer. Up to 3650 atoms can be displayed at a rate of 20 atoms/s. Results of distance geometry calculations with experimental data from nuclear magnetic resonance experiments[5] can be displayed as well as structures from the Brookhaven Protein Data Bank files or the Cambridge Crystallographic Data File. Fifteen colours are available to distinguish different types of atoms, so that each sphere can be shaded in 17 tones of colour. The colour-assignments can be changed within a few seconds by reading a small data-file. Each assigned colour may be varied instantaneously by giving new red-green-blue-values on the terminal. In addition to the presentation of molecular models consisting of opaque atomic spheres, the α-carbon atoms along the protein backbone can be connected by a smooth curve, which greatly facilitates recognition of chain-folding and secondary structures ('snake' representation).

A striking 3-D impression is achieved by rocking the molecule around its vertical axis. This effect is based on the computation of four separate views, whereby the molecule is rotated by 6° from image to image. Since all four memory buffers are necessary for the computation of any one image, the first three images are stored temporarily on disc and read back after the completion of the last one. The rocking motion is generated by going forwards and backwards through the series of four images, so that each view is displayed for 0.1 s.

Subroutines have been included into the program which permit inclusion of lettering of variable size and colour at any desired position of the image. Rescaling of the atomic radii can simply be carried out by writing the new spheres to plane A and repeating the display process. Hardcopies of the screen displays in the form of colour-slides were prepared by photographing the monitor with a conventional 35 mm camera mounted on a tripod (see Colour Plates 1–8).

Colour Plate 1 shows a space-filling model of a proteinase inhibitor from bull seminal plasma (BUSI IIA) in aqueous solution. The coordinates are those of a structure determined by distance geometry calculations using data from proton nuclear resonance experiments[5]. Atoms are coloured according to the conventional code as follows: C: grey; H: white; N: blue; O: red; S: yellow. Colour Plate 2 shows a space-filling model of BUSI IIA in the same orientation as in Colour Plate 1. The amino acid side chains are coloured according to their polarity[6] using the following code: backbone: grey; hydrophobic: pale yellow; neutral: yellow; uncharged polar: green; positively charged: blue; negatively charged: red. With our program the change of colouring between Colour Plates 1 and 2 is completed within a few seconds. Colour Plate 3 is a 'snake' presentation of the backbone of BUSI IIA in the same orientation as in Colour Pate 1. The smooth curve was generated with a cubic spline interpolation function[7] through all α-carbon atoms along the backbone. In Colour Plate 4 the snake presentation of BUSI IIA is supplemented with a colour code indicating the molecular dynamics manifested by the amide-proton exchange rates[8]. Extremely fast, fast, medium and slow exchange is indicated by red, yellow, green and blue, respectively. The central α-helix is easily identified as a region with particularly slow exchange.

The structures in Colour Plates 5–8 are based on an updated version[9] of average coordinates for A- and B-DNA obtained from fibre-diffraction data[10]. The complete double-strands were assembled using a com-

puter program which adds increments of rotation and translation to the given starting coordinates for each single nucleotide. A DNA segment consisting of an arbitrary sequence of 15 base pairs is shown. In Colour Plate 5 the same atom colouring scheme as in Colour Plate 1 is used for a presentation of B-DNA. Colour Plate 6 presents B-DNA as in Colour Plate 5 with a different colouring scheme, where the sugar units appear grey, the P-atoms violet and the bases as follows: adenine: blue-green; thymine: yellow; guanine: red; cytosine: blue. Colour Plate 7 shows A-DNA using the standard atom-colouring, whereas Colour Plate 8 uses the colouring scheme of Colour Plate 6.

The following are important factors which differentiate the present parallel algorithm from the scan-line method[3]. The image build-up speed depends linearly on the number of displayed atoms, whereas in the scan-line method it also depends on the number of pixels to be computed and the complexity of the image, i.e. lines with many intersecting spheres require more computation time than less populated ones. The resulting differences in image build-up times depend on the size of the molecule (a space-filling model with 64 atoms requires 39 s with the scan line method and 3.2 s with the parallel algorithm; with 2437 atoms the corresponding times are 265 s and 123 s). The parallel algorithm is executed mainly by the graphics processor, and the host-computer is used exclusively to send the atomic coordinates and type information to the graphics system. This reduces the number of time-consuming data-transfers and saves computer memory, which becomes available to store larger molecules. Increased image resolution by extension of the hardware would not affect the computation time of the parallel algorithm, but would require considerably more computation time for the scan-line method, where all pixels are computed sequentially[3].

The image display by the parallel algorithm, which proceeds from the back to the front of the molecule, is in itself quite instructive. Following the atom-by-atom build-up gives a vivid impression of the side-chain packing, or of the location and architecture of active sites. The possibility of interrupting the procedure at any desired stage provides a simple way of generating views at internal atomic surfaces corresponding to 'cross-sections' through the molecule.

Recently Hubbard and Fincham[4] have independently developed a similar algorithm for the computation of space-filling models on a fast, general-purpose parallel processing computer. Combined with further hardware developments use of such methods with parallel processing graphics systems might eventually lead the way toward truly interactive raster graphics for molecular modelling.

ACKNOWLEDGEMENTS

The authors would like to thank Mr F Klein for a helpful introduction to the programming of the array-processor and Dr M Billeter for stimulating discussions. Use of the facilities at the ZIR (Zentrum für Interaktives Rechnen) of the ETH Zürich and assistance by its staff, especially by Mr A Gautschi is gratefully acknowledged.

REFERENCES

1 **Billeter, M, Engeli, M and Wüthrich, K** 'Interactive program for investigation of protein structures based on ^{1}H NMR experiments' *J. Mol. Graph.* Vol 3 No 3 (1985) pp 79–83

2 **Pavlidis, T** *'Algorithms for Graphics and Image Processing'* Springer-Verlag, FRG (1982) p 337

3 **Porter, T K** 'Spherical shading' *Computer Graph.* Vol 12 (1978) pp 282–285

4 **Hubbard, R and Fincham, D** 'Shaded molecular surface graphics on a highly parallel computer' *J. Mol. Graph.* Vol 3 (1985) pp 12–14

5 **Williamson, M P, Havel, T F and Wüthrich, K** 'The solution conformation of proteinase inhibitor IIA from bull seminal plasma by ^{1}H nuclear magnetic resonance and distance geometry' *J. Mol. Biol.* Vol 182 (1985) pp 295–315

6 **Guy, H R** 'Amino acid side-chain partition energies and distribution of residues in soluble proteins' *Biophys. J.* Vol 47 (1985) pp 61–70

7 **International Mathematical and Statistical Libraries (IMSL)** 'Subroutine for cubic spline interpolation' *ICSCCU*

8 **Wüthrich, K, Strop, P, Ebina, S and Williamson, M P** 'A globular protein with slower amide proton exchange from an α-Helix than from antiparallel β-sheets' *Biochem. Biophys. Res. Comm.* Vol 122 (1984) pp 1174–1178

9 **Arnott, S** Personal communication (1984)

10 **Arnott, S and Hukins, D W L** 'Optimised parameters for A-DNA and B-DNA' *Biochem. Biophys. Res. Comm.* Vol 47 (1972) pp 1504–1509