

Rendering volumetric data in molecular systems

David S. Goodsell, I. Saira Mian and Arthur J. Olson

Research Institute of Scripps Clinic, La Jolla, CA, USA

A raster-based computer graphics method of imaging volumetric data (data sampled on a three-dimensional grid) has been added to an existing molecular rendering program. Any molecular property expressed as a 3D grid of scalar data values can be displayed around and inside of the molecular van der Waals surface as a collection of colored clouds and transparent surfaces. Different anti-cal properties, such as color and opacity, are assigned to ranges of the molecular property and are rendered by a ray-tracing technique simulating the shading and shadowing of real objects, making the final images highly readable. We have found useful applications in the visualization of crystallographic electron density, electrostatic potentials and protein active sites.

Keywords: volume rendering, raster graphics, transparent surfaces, cast shadows

INTRODUCTION

The effective visualization of the complex properties of macromolecules is becoming increasingly important as the number of three-dimensional (3D) structure determinations grows. Many of these properties are observed as scalar data values sampled over a grid in three dimensions—as volumetric data. Examples include electron density from X-ray crystallography, 3D reconstructions from electron microscopy and electrostatic potentials calculated around molecules. Traditionally, this type of data has been analyzed by calculating an intermediate surface composed of dots^{1,2} or a polygonal mesh,^{3,4} which is examined on vector-based graphics hardware. The technique works well for the interactive study of specific regions of the molecule, but complex vector images of larger regions or of multiple nested surfaces become all but uninterpretable, lost in a maze of lines and dots.

In this article, we describe an alternate approach to the display of 3D molecular properties combining several raster-based techniques of modeling natural phenomena.

Optical properties, such as opacity and color, are assigned to the array of data values, creating a model image composed of colored clouds and transparent surfaces. The resultant images are very readable, as the rendering technique mimics the shading and shadowing of real objects. Distant parts of the molecule are obscured by foreground objects, so that the distracting build-up often seen in vector images is not a problem.

OVERVIEW OF THE VOLUME-RENDERING ALGORITHM

Two different approaches to the rendering of volumetric data have been reported: ray tracing⁵⁻⁸ and cell projection.^{8,9} We have chosen the ray-tracing method for several reasons. It was easily layered over RMS,¹⁰ a z-buffer-based atomic surface rendering program. (Throughout this paper, the term “atomic surface” will refer to a representation of the molecule as a collection of spheres centered on the atomic coordinates. All other references to surfaces will refer to surfaces generated by the volume rendering technique.) The ability to include atomic surfaces is important in molecular images to provide the context for the volume-rendered properties. The ray-tracing method also lends itself to different types of representation—we have implemented both a transparent surface representation and a space-filling cloud representation.

A typical image is composed of several layers rendered in different manners. Figure 1a is a diagram of the imaging space. The background is calculated externally and input at the beginning of the run. For the figures in this paper we have used backgrounds with simple color gradients and horizontal lines. The atomic surface is added over the background, saving the height of the surface at each pixel in a z-buffer. Volume-rendered surfaces and clouds are then added simultaneously. A ray is cast from each pixel away from the viewer through the data grid, finally striking either an atomic surface or the background. As the ray passes through areas in the data grid tagged as being opaque or through a transparent surface, the ray is incrementally attenuated. When the ray terminates, the final summed opacity is used to obscure the atomic surface or background in the ray path (Figure 1b). Several refinements are

Address reprint requests to Dr. Goodsell at the Department of Molecular Biology, Research Institute of Scripps Clinic, La Jolla, CA 92037, USA.

Received 15 September 1988; revised 14 November 1988

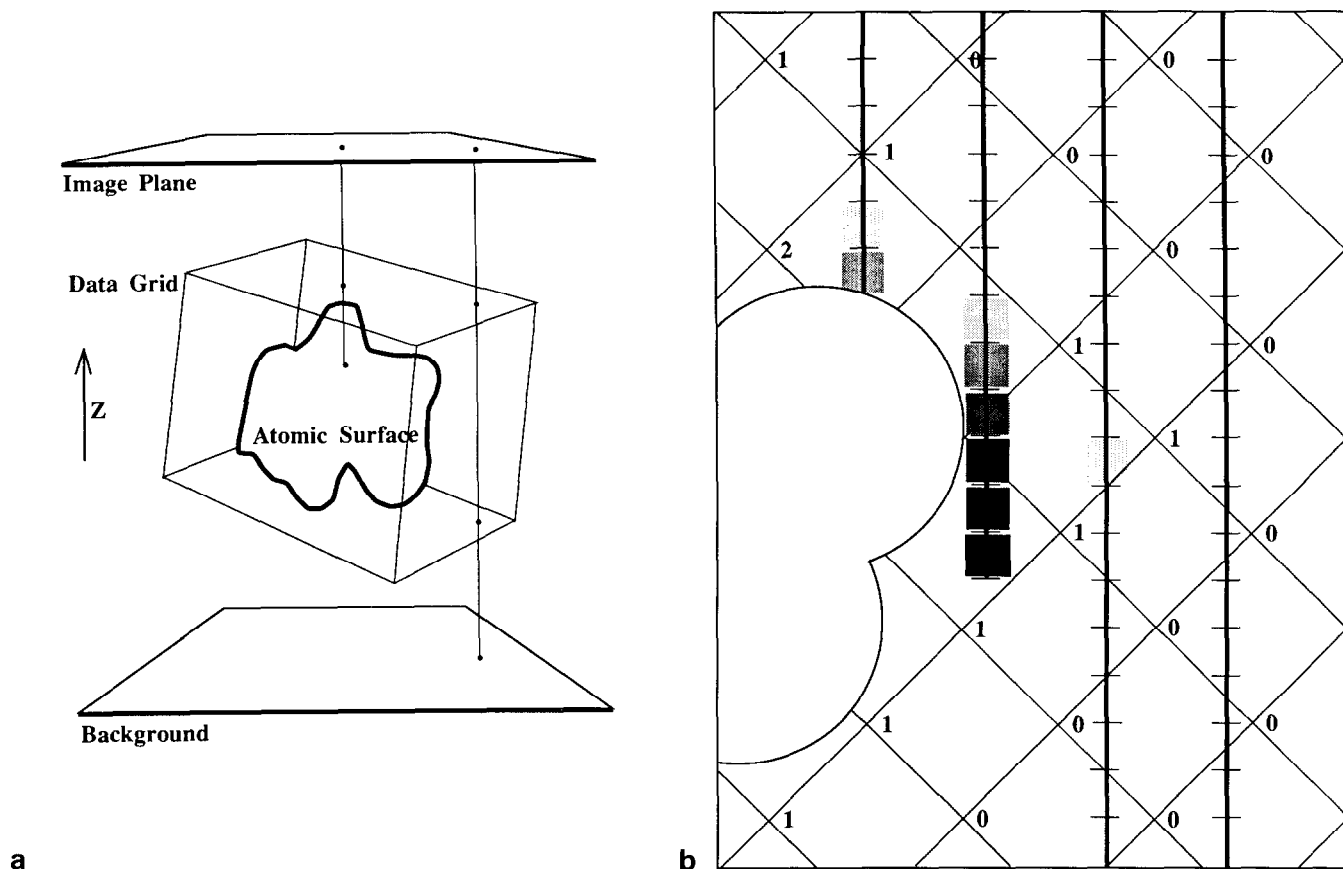


Figure 1. (a) Diagram of the imaging data space. The viewer is at the top of the diagram, looking down through the image plane at the atomic surface and the data grid. Two sample rays are shown. (b) Schematic of the incremental summation method of volume rendering. A portion of the data grid is shown, with the associated scalar data values. Several sample rendering rays are shown. The user has chosen to make steps along a ray opaque if the scalar data value at the step is greater than 1. The first ray becomes partially opaque before striking an atomic surface—the pixel will be colored partially by the cloud color and partially by the atomic surface color. The second ray becomes totally opaque in the cloud density—the pixel will be the color of the cloud. The third ray glances through a region with values greater than 1 and will be colored partially by the cloud and partially by the background. The fourth ray has the color of the background

implemented to improve the legibility of the images, including cast shadows for surfaces, atmospheric depth cuing and the ability to clip away a portion of the image near the viewer.

ATOMIC SURFACES AND SHADOWS

A spherical atom representation of the atomic surface is calculated prior to the rendering of volumetric properties, using the mapping algorithm described for RMS.¹⁰ Briefly, a shading template is calculated for eight atom types at the beginning of the run, containing the z-height, color and surface normal of each point on the sphere (Figure 2a). This template is then mapped over each atomic position in a z-buffer, saving the uppermost surfaces at each pixel. The result is a viewer-visible shell hanging above the background and below the image plane (Figure 2b).

An improved shadowing algorithm has been implemented combining the method of Williams¹¹ with the mapping algorithm described above. Williams generates a second image from the point of view of the light source. The surface points in the original image are then compared to the corresponding points in the second image to determine shadowed regions. If a point is visible in the original image but is not visible in the second image (not visible to the light source), it is in shadow and is shaded accordingly. Points visible in both images are fully lit.

To apply the mapping algorithm described above to this shadow technique, a different type of template is used, containing the z-height of all regions of the sphere visible to the light source, mapped into a plane co-planar with the image plane (Figure 2c). This template is mapped into a second shadowing z-buffer using modified atomic positions. The atomic coordinates (x_a , y_a , z_a) are modified to give the projection of the atom along the light ray onto the shadowing image plane (x_{sh} , y_{sh} , z_{sh}):

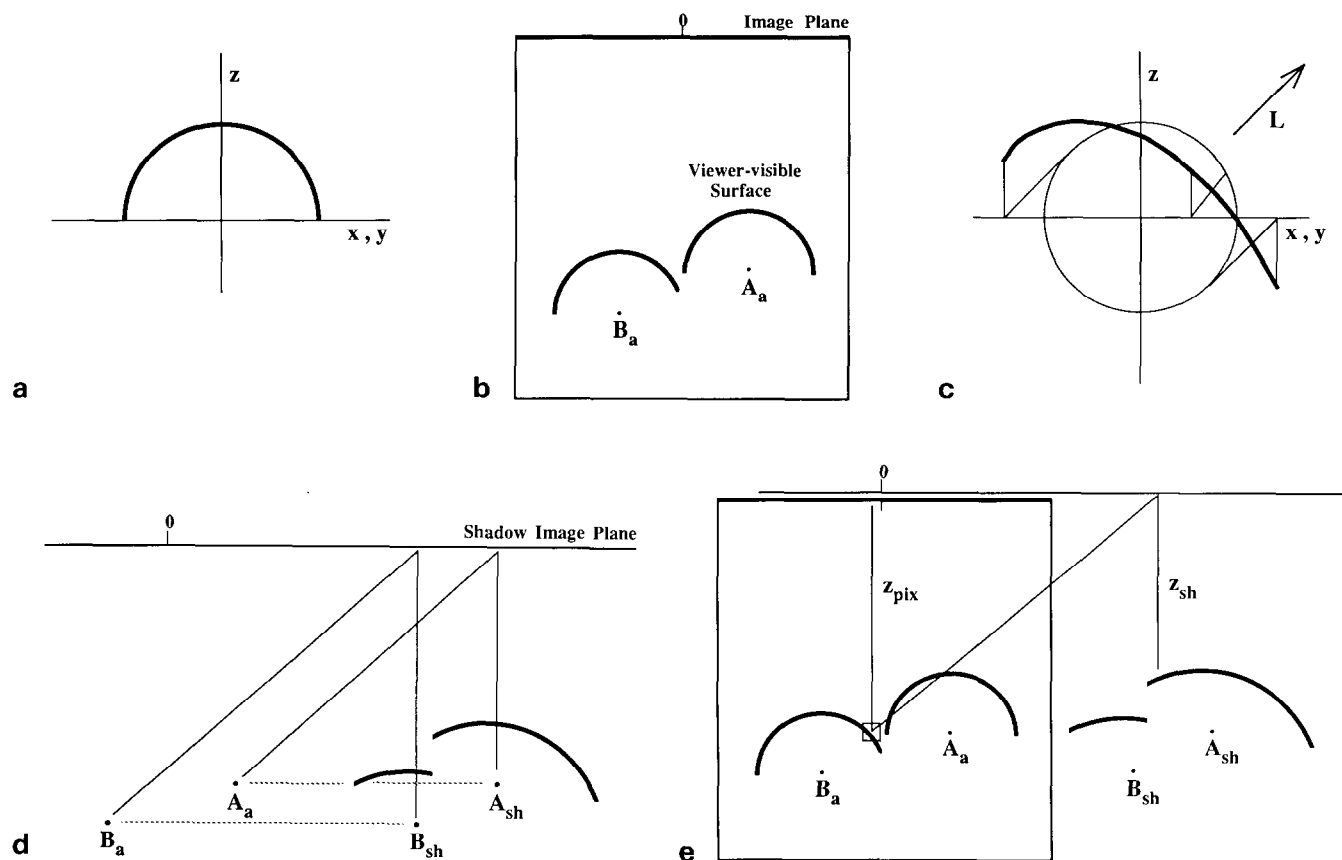


Figure 2. Diagrams of the mapping algorithms used for atomic surfaces and shadows. In each diagram, the viewer is at the top and the image plane is intersecting the paper with negative z below the horizontal intersection line. (a) Atomic surface template. (b) Templates mapped into the z -buffer, over atomic positions A_a and B_a . A portion of the surface of atom B is not visible to the viewer. (c) Shadowing template. The dark line is the template, derived from the sphere shown in light lines. The light vector (L) is shown at upper right. (d) Shadow templates mapped into a second z -buffer, over the new atomic positions A_{sh} and B_{sh} . A portion of the B shadow surface is blocked, so part of B will be in shadow. (e) Shadow testing. The two z -buffers are superimposed and points on the atomic surface are tested for shadowing. The boxed point on the B atomic surface is tested in the diagram and found to be in shadow

$$\begin{aligned} x_{sh} &= x_a - L_x \times z_a / L_z \\ y_{sh} &= y_a - L_y \times z_a / L_z \\ z_{sh} &= z_a \end{aligned} \quad (1)$$

where L_x , L_y and L_z are the direction cosines pointing at the light source. Figure 2d diagrams the mapping of the shadowing templates.

The complete shadowing z -buffer is used to test each pixel in the viewer-visible surface. Figure 2e diagrams the superposition of the two z -buffers and shows the testing of one pixel on atom B . The z -level of the pixel in the surface (z_{pix}) is compared to the z -level in the shadow map (z_{sh}), using the same coordinate transformations described above. If z_{pix} is greater than z_{sh} , the pixel is in shadow. Antialiasing of shadow edges is performed in the template, as described for atomic edges in RMS.¹⁰

This shadowing algorithm works well for light source positions within about 45 degrees from the normal to

the image plane. Of course, pathological cases can be found; with very deep images or images with many atoms outside of the window, shadowing atoms may fall beyond the borders of the shadowing image plane. In this case, a more rigorous approach can be taken by searching for all atoms that intersect the ray from the surface to the light source. Adding shadows to a typical atomic surface by this mapping algorithm will approximately double the surface computation time.

VOLUME RENDERING BY INCREMENTAL SUMMATION

The incremental summation method represents the volumetric data as space-filling clouds with color and opacity varying according to the scalar data values in the grid. The user defines a transfer function composed of linear ramps that maps a color and opacity to ranges of scalar

data values. Figure 3 shows some examples of different transfer functions applied to the same data set. The use of piece-wise linear ramps allows the user to create a wide range of different images, from the glowing clouds in Color Plate 1 to the opaque contours in Color Plate 3. Shading is not performed in this representation, as the self-shadowing algorithms necessary for proper shading of cloud densities were judged too costly for systems of this complexity. Shading cues can be added through the use of isovalue surfaces.

Calculation of space-filling clouds begins by casting a ray from each pixel into the data grid. At user-defined intervals along this ray, scalar data values are calculated by trilinear interpolation of the surrounding eight data points. The transfer function is then used to look up the partial opacity of the interval (α_{step}) and the color of the interval (I_{step} —one for each red, green and blue). This partial opacity is added to a growing total opacity of the pixel:

$$\alpha_i = \alpha_{i-1} + (1 - \alpha_{i-1}) \alpha_{\text{step}} \quad (2)$$

where α_i is the opacity of the pixel at interval i along the ray. The pixel intensities are summed similarly (for color images, separate intensity values are calculated for red, green and blue):

$$I_i = I_{i-1} + (1 - \alpha_{i-1}) I_{\text{step}} \quad (2)$$

The ray is followed until: (1) the ray passes through the back surface of the data grid, (2) an atomic surface or the background is reached, or (3) the opacity α_i exceeds a given threshold, typically 0.99. The final summed opacities (α_{volume}) and colors (I_{volume}) are used to obscure the atomic surface or background intensity of the pixel (I_{back}), to give the final intensity of the pixel (I_{pixel}):

$$I_{\text{pixel}} = (1 - \alpha_{\text{volume}}) I_{\text{back}} + \alpha_{\text{volume}} \times I_{\text{volume}} \quad (4)$$

By careful choice of transfer functions, the clouds will gradually fade out at silhouetted edges, and if the size of the stepping interval along the ray is kept small (on the order of 0.5 to 1.0 pixel widths), sampling artifacts are small.

VOLUME RENDERING BY THRESHOLDING

Isovalue surfaces are added to the image in the same ray-tracing operation used for the cloud representation. If two successive interpolated data values along the ray bracket a user-defined surface isovalue, a surface has been crossed. The opacity and color of this surface is added to the growing opacity and color of the pixel in exactly the same way as the cloud densities are summed. Surface normals for shading are computed as the gradient of the data at the point of intersection of the ray and the surface. The gradient is obtained by trilinear interpolation of the gradients of the eight surrounding points.

To reduce computation time for the shading calcu-

lation, a shading template is created at the beginning of the run containing the surface intensity for each possible forward-looking surface normal ($N_x^2 + N_y^2 \leq 1$ and $N_z > 0$). This template is used as a lookup table to map a color to a given surface normal. We noticed in early images that the specular highlights, which are important visual cues for the perception of surface features, tended to get washed out on transparent surfaces. As a solution to this problem, we create an additional template for the opacity making the surface more opaque at the specular reflection:

$$\alpha_N = \alpha_{\text{surface}} + \alpha_{\text{specular}} \times \cos(s)^n \quad (5)$$

$$\alpha_{\text{surface}} + \alpha_{\text{specular}} \leq 1$$

where α_N is the opacity of a surface with normal N , α_{surface} and α_{specular} are the constant opacity of the surface and the amount by which the opacity is augmented at the specular reflection respectively and $\cos(s)^n$ is the angular term defining the specular reflection, as described by Phong.¹² The increased opacity models a bright reflection that saturates the color of the pixel.

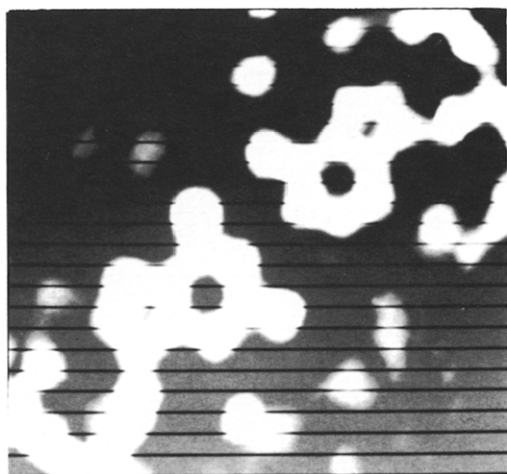
Shadows and atmospheric depth cuing are added to images during the ray-tracing operation. Shadows cast from atoms onto volume rendered surfaces are calculated by the shadowing algorithm described above, using the same shadowing z -buffer used for the atomic surfaces. A more rigorous approach is also implemented, casting a ray back toward the light source as each surface is crossed. This yields fully self-consistent cast shadows, but requires prohibitive amounts of computation time. Depth cuing is performed using the "fog" described for RMS. As the ray penetrates deeper into the image space, the color of each step is increasingly attenuated by a user-specified fog. The images gradually fade into the distance, as seen in Color Plate 3. Both shading and depth cuing modify the color intensity term in Equation 3:

$$I'_{\text{step}} = (I_{\text{step}} \times P_{\text{shadow}})(1 - \alpha_{\text{fog}}) + \alpha_{\text{fog}} \times I_{\text{fog}} \quad (6)$$

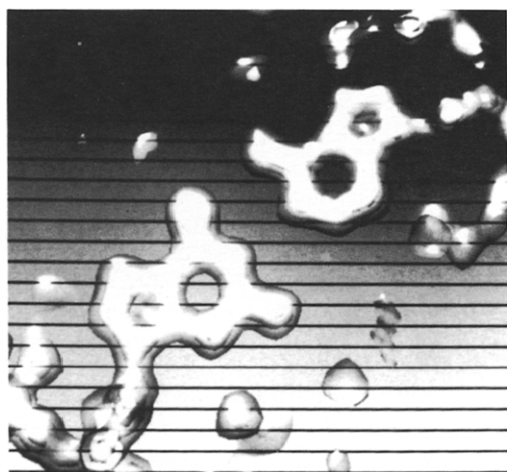
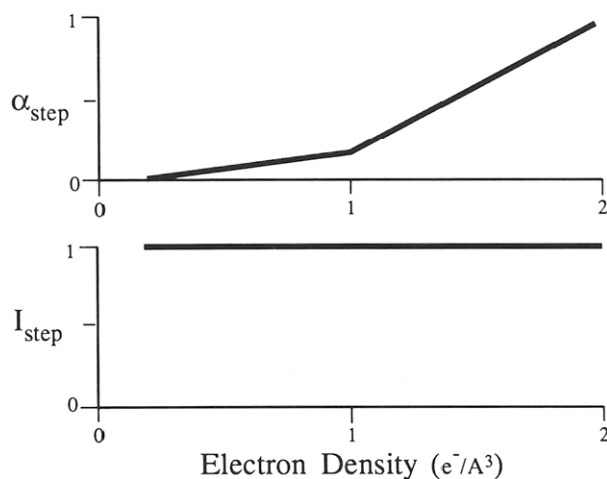
P_{shadow} varies from 1 to 0 as the surface is increasingly in shadow, α_{fog} varies from 0 to positive values less than 1 at increasing depths and I_{fog} is the color of the fog.

APPLICATIONS

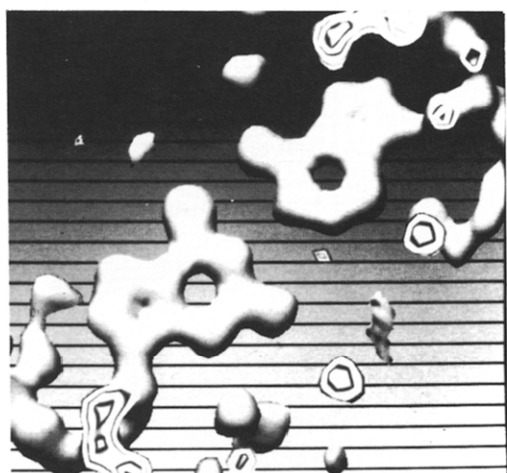
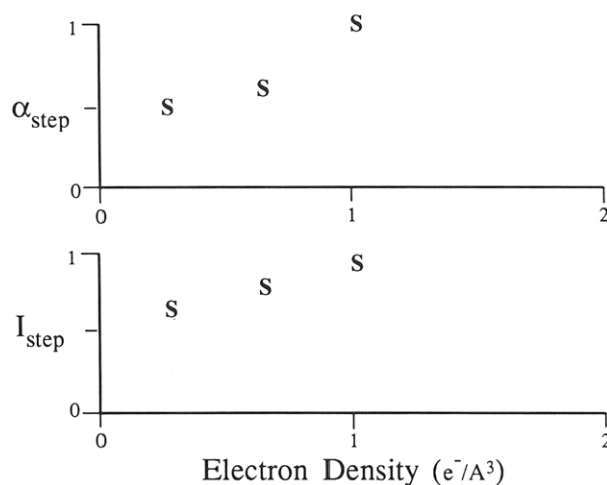
We have had significant success in visualizing molecular electrostatic potentials with this technique. Two examples are presented here: a 12-residue poly-alanine alpha helix and poliovirus. In Color Plate 1, the electrostatic potential of a short alpha helix is rendered by the incremental technique with low opacity in regions of low potential magnitude and gradually increasing opacity as the potential increases near the molecule. As the opacity increases, the color of the cloud is gradually shifted from saturated blue or red toward white, producing a glowing "energetic" center to regions of strong potential. The image shows the distribution of potential in the space around the molecule in relation to the atomic surface. The obvious dipole seen across the length of



a



b



c

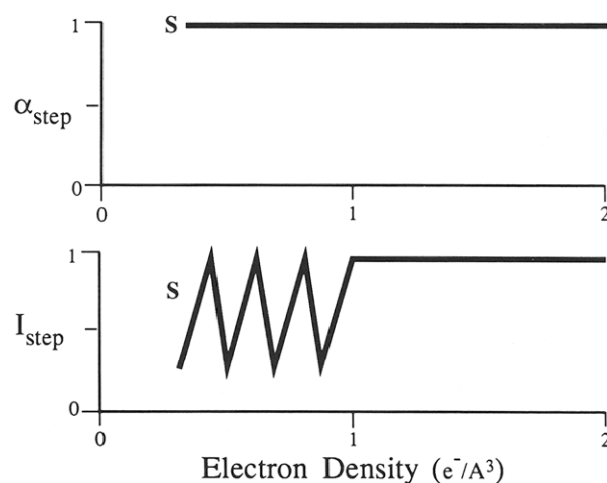


Figure 3. Three different transfer functions applied to the same volumetric data. A Fourier electron density map from the structure determination of a DNA decamer¹⁴ is imaged, showing only the central G-A mispair. The opacity transfer function (α_{step}) and intensity transfer function (I_{step}) used to calculate each image are shown to the right of the image. (a) Cloud representation. Two ramp functions were defined, reaching total opacity at the highest electron density levels. (b) Surface representation. Three concentric transparent surfaces are drawn, with colors and opacities shown with a boldface S on the graphs. Surfaces at higher levels were made progressively brighter and more opaque to increase contrast in the image. (c) Combined representation. Totally opaque clouds are drawn inside a glossy opaque boundary surface, with the cloud color varying with density level. The clouds are seen as a contoured surface when the molecule is clipped at the image plane

Table 1. Calculation times for the Color Plates

	Resolution (pixels)	Grid size	Grid spacing (Å)	Computation time (min.)	
				Atoms	Volumes
Color Plate 1	260 × 500	30 × 30 × 60	0.5 × 0.5 × 0.5	2	35
Color Plate 2	450 × 450	90 × 90 × 90	2.0 × 2.0 × 2.0	3	18
Color Plate 3	480 × 480	48 × 48 × 48	0.68 × 0.55 × 0.63	—	26
Color Plate 4	500 × 500	60 × 60 × 60	0.5 × 0.5 × 0.5	1	24

All calculations were performed on a Convex C1 computer and displayed on a Sun/TAAC1 frame buffer at 24 bits per pixel—8 bits for each red, green and blue

the helix is due solely to the alignment of mainchain amides, as uncharged ends were used for calculation of the electrostatic potential grid.

The poliovirus¹³ potential surface is shown as a 70% opaque isovalue surface in Color Plate 2. Phong shading and cast shadows heighten the 3D aspect of the potential and atomic surfaces. This type of image is useful for getting an empirical "overall" view of protein electrostatics, without the limitations inherent in imaging techniques that look at only a planar section through the data space. Ideas about substrate steering and molecular association are facilitated: the potential surface may be thought of as a "capture radius" (i.e., charged molecules that diffuse into the surface have a high probability of being pulled toward the protein and ultimately contacting the atomic surface inside).

Electron density maps from X-ray crystallography are also effectively rendered by this method. Color Plate 3 displays a Fourier map from the high-resolution structure solution of the DNA oligonucleotide CCAA-GATTGG,¹⁴ highlighting the unusual guanine-adenine base pair formed at the center of the DNA duplex. Both integrative and thresholding techniques are used; an opaque green boundary surface surrounds totally opaque clouds with colors that gradually change as the density level increases. The clouds are seen as colored contours when the data grid is clipped at the image plane. A very sharp depth cuing gradient is used to simplify the image, highlighting only the region of interest.

The volume-rendering method is applicable to any other molecular property expressed as a 3D sampled scalar field. Color Plate 4 highlights the shape and electrostatic complementarity of the phosphorylcholine binding site in the Fab molecule McPC 603.¹⁵ This volume-rendered surface is calculated using a Gaussian approximation for the protein atoms, similar to that described by Blinn,¹⁶ choosing parameters such that a given fraction of the atom (0.8 in this case) is enclosed inside the van der Waals radius. By changing the contour level used for rendering, the percentage of each atom enclosed by the smooth surface can be changed. In the plate, we have chosen a contour level that places the surface at the van der Waals radius. Notice the triangular pocket that perfectly holds the quaternary amine of the phosphorylcholine (shown as a ball-and-stick model). The electrostatic complementarity is visualized by coloring the surface with a second grid contain-

ing the electrostatic potential of the protein. The triangular pocket holding the positively charged amine is strongly negative and the positively charged protein lobe seen above the hapten interacts with the phosphorylcholine phosphates.

PERFORMANCE

The final program is extremely flexible. A consistent representation can be used to render any property that is calculated on a 3D grid, regardless of resolution, scale or source. This flexibility has a cost: volume-rendering methods are not computationally cheap. The images in this paper were calculated on a Convex C1 computer and photographed from a Sun/TAAC1 frame buffer. In a typical application, we calculate a number of low-resolution images, about 100 by 100 pixels requiring 1 or 2 minutes of computer time, to do a general survey of the molecule. Then, several interesting views are chosen for the final high-resolution images. Calculation times for each of the plates are included in Table 1. The calculation time depends mainly on the resolution of the image, not on the size of the volumetric data grid; on average, 24-bit high-resolution images require about 25 minutes. This is certainly not prohibitive, and we can expect a substantial decrease in the computation time as more highly parallel computer systems are used.

ACKNOWLEDGEMENTS

We thank Gilbert Prive and Richard E. Dickerson for use of their electron density map, Jim Hogle for use of the poliovirus coordinates and Michael Pique for his technical assistance. This work was supported in part by Grant DRG 972 from the Damon Runyon-Walter Winchell Cancer Research Fund and NIH Grant GM35-221144.

REFERENCES

- 1 Getzoff, E. D., Tainer, J. A. and Olson, A. J. Recognition and interactions controlling the assemblies of β barrel domains. *Biophys. J.* 1986, **49**, 191–206

- 2 Klapper, I., Hagstrom, R., Fine, K. and Honig, B. in *Proteins: Structure, Function, and Genetics 1*, 1986, 47-59
- 3 Koide, A., Doi, A. and Kajioka, K. Polyhedral approximation approach to molecular orbital graphics. *J. Mol. Graphics* 1986, **4**, 149-155
- 4 Jones, T. A. *J. Appl. Crystallog.* 1978, **11**, 268-272
- 5 Kajiya, J. and Von Herzen, B. Ray tracing volume densities. *Computer Graphics* 1984, **18**, 165-174
- 6 Levoy, M. Display of surfaces from volume data. *IEEE Comp. Graphics & Appl.* 1988, **8**, 29-37
- 7 Sabella, P. A rendering algorithm for visualizing 3D scalar fields. *Computer Graphics* 1988, **22**, 51-58
- 8 Upson, C. and Keeler, M. V-BUFFER: visible volume rendering. *Computer Graphics* 1988, **22**, 59-63
- 9 Drebin, R. A., Carpenter, L. and Hanrahan, P. Volume rendering. *Computer Graphics* 1988, **22**, 65-74
- 10 Goodsell, D. S. RMS: programs for generating raster molecular surfaces. *J. Mol. Graphics* 1988, **6**, 41-44
- 11 Williams, L. Casting curved shadows on curved surfaces. *Computer Graphics* 1978, **12**, 270-274
- 12 Phong, B. T. Illumination for computer generated pictures. *Comm. ACM* 1975, **18**, 311-317
- 13 Hogle, J. M., Chow, M. and Filman, D. J. Three-dimensional structure of poliovirus at 2.9 Å resolution. *Science* 1985, **229**, 1358-1365
- 14 Prive, G. G., Heinemann, U., Chandrasegaran, S., Kan, L.-S., Kopka, M. L. and Dickerson, R. E. Helix geometry, hydration, and G-A mismatch in a B-DNA decamer. *Science* 1987, **238**, 498-504
- 15 Segal, D. M., Padlan, E. A., Cohen, G. H., Rudikoff, S., Potter, M. and Davies, D. R. The three-dimensional structure of a phosphorylcholine-binding mouse immunoglobulin Fab and the nature of the antigen binding site. *Proc. Natl. Acad. Sci. USA* 1974, **71**, 4298-4302
- 16 Blinn, J. A generalization of algebraic surface drawing. *ACM Trans. Graphics* 1982, **1**, 235-245