Personal computer-based visualization of molecular models by available ray-tracing software

Arnold S. Dion

Garden State Cancer Center at the Center for Molecular Medicine and Immunology, Newark, New Jersey, USA

Developed primarily for the graphic arts, ray-tracing algorithms offer a high level of flexibility with reference to photorealistic and surrealistic image rendering. The utility of these programs is further enhanced by portability (i.e., compatibility with a number of operating systems), accessibility through various sources, and low cost. This report documents, through the use of existing software, the application of these ray-tracing attributes to molecular graphics on a desktop computer. This application is especially pertinent in view of rapid speed enhancements in personal computers (PCs), which have enabled molecular modeling and dynamics on these systems. In this regard, ray tracing on a PC provides enhanced capabilities for molecular graphics rendering that are potentially equivalent to those achieved by workstations.

Keywords: molecular graphics, ray tracing, ray-tracing software, personal computer

INTRODUCTION

A principal challenge of molecular graphics is the rendering of a two-dimensional (2D) image so that it realistically portrays a three-dimensional (3D) structure. The latter might represent an image resulting from direct visualization of published atomic coordinates or might involve further refinements to the coordinate file (e.g., geometric transformations, residue substitutions, energy minimizations, etc.). For "low- to middle-end" molecular visualization/modeling software applications written for the desktop PC, the quality of graphic output varies considerably and often allows few options for variations in color, texture, shading, etc. In this regard, ray-tracing algorithms[1,2] are highly portable and offer economical and highly flexible programs for rendering high-quality molecular structural images on workstations, as exemplified by RASTER3D, written by David Bacon. The present report documents the application of a suite of programs, all available as freeware or shareware, for ray tracing molecular structures on a PC. The collective implementation of these programs also represents a suitable alternative to the *de novo* programming approach.[3]

SOFTWARE

All of the programs mentioned in this section are available as freeware or shareware. For shareware programs, a registration fee ($25–$70) is required beyond a trial period of usage. *Persistence of Vision Raytracer* (POVRAY)[4] is available as freeware for the DOS, Mac, Commodore Amiga and Unix-based operating systems. For the IBM-compatible PC employed in this report, POVRAY (Version 1.0) was compiled with the Intel Code Builder (Version 1.1) in 32-bit protected mode. Execution of this program for rendering the screen output of a specific molecular structure file (see examples in the next section) is accomplished from the command line. Global default options (POVRAYOPT and POVRAY.DEF) somewhat lessen the awkwardness of this file and variable options input. In addition, recent versions of POVRAY include a useful and user-friendly freeware program (POVMENU), which provides a menu-driven interface for POVRAY. POVRAY (Version 1.0) also includes a utility shareware program (PICLAB) for converting 24-bit, true-color output files (Targa, RLE or RAW) to more concise 8-bit, 256-color graphics interchange format (GIF) files initially developed by Compuserve. In this regard, it should be noted that POVMENU currently provides for only Targa file output. Optimal usage of POVMENU batch files also requires additional shareware programs (e.g., ALCHEMY [as an alternative to PICLAB], QEdit or Multiedit for structural file editing, LIST for file listing, VPIC or COMPUSHOW for visualization and IMPROCES for image processing). POVMENU supports functionally equivalent software to those cited, provided that program execution or redirection is possible from the command line.

All of these individual programs are obtainable from The Software Labs (Culver City, CA), as well as other software distributors. As alternatives, the latest version of POVRAY is available through various on-line (Compuserve, America Online, Internet, etc.) and bulletin board services. The

Color Plates for this article are on page 36.

Address reprint requests to Arnold S. Dion, Garden State Cancer Center at the Center for Molecular Medicine and Immunology, 1 Bruce Street, Newark, New Jersey 07103, Telephone: (201) 982-5414, Fax: (201) 982-7047.

Received 11 June 1993; accepted 12 October 1993

Waite Group Press (Corte Madera, CA) is a distributor of a software package that includes documentation and a variety of the programs previously mentioned (POVRAY, PICLAB, ALCHEMY, COMPUSHOW and IMPROCES).[5] Shareware programs obtained from these sources also require registration fees. All of these products are copyrighted by their respective authors or organizations.

HARDWARE AND OPERATING SYSTEM REQUIREMENTS

For IBM or compatible computers, POVRAY is executable only on 80386 or 80486 CPU-based systems with a math coprocessor and at least 2 MB of RAM. Systems lacking a math coprocessor are bypassed by emulation, but without speed enhancement. DOS, OS/2 and Windows 3.x are supported. However, POVRAY is only compatible with the enhanced mode of Windows 3.x as a consequence of protected mode operation. Various memory managers are compatible with POVRAY, including 386MAX, QEMM386, DOS 5.0 EMM386 and DOS = HIGH, and NETROOM. Finally, POVRAY and the programs for visualization and image processing designated earlier include a wide range of video graphics display drivers, as listed in their respective documentation.

All of the ray-traced models listed here were generated on an IBM-compatible 80386DX/25 MHz computer with a Cyrix 80387 math coprocessor and 8 MB RAM in a Microsoft DOS 5.0 environment. A Trident graphics display adapter (512K DRAM) was used for display on an SVGA monitor.

FILE FORMATS AND CONVERSIONS TO POVRAY SCRIPT FILES

To date, POVRAY has been employed in two primary applications—for direct visualization of atomic coordinates derived from Protein Data Bank (PDB) files or for rendering images resulting from molecular modeling/dynamics by PC-compatible programs including MOBY (Springer-Verlag, Berlin Heidelberg, Germany) and Alchemy III (Tripos Associates, Inc., St. Louis, MO). Although MOBY and Alchemy III directly convert PDB files to their own formats, POVRAY utilizes script files, as discussed later, that are distinct from both PDB files or molecular-modeling output formats. However, the conversion of these files to POVRAY script files can be easily accomplished by wordprocessing and spreadsheet programs, especially through the use of macros. For example, all of the output files mentioned earlier contain three basic sections consisting of header comments, atomic coordinates, and connectivities. Of these, only the atomic coordinates need to be retained for space-filling models rendered by POVRAY script files. The other sections can be deleted *en bloc* by wordprocessor (e.g., QEDIT) or spreadsheet programs. The pertinent section containing the atomic coordinates also includes other descriptors (atom number, atom type, etc.) depending on the file, which can be rapidly removed by spreadsheet programs. In addition, the "find-and-replace" function of wordprocessors is very useful for customizing each line of the POVRAY script file with reference to color or van der Waals radii assignments on the basis of atom type. Most of these operations are facilitated by macros, so that a structure composed of 2000 centers can be converted to a POVRAY script file in less than 15 minutes. This approach has the advantages of versatility (multiple formats are easily accommodated), speed (file conversion represents a small percentage of total rendering time), and flexibility (specific structural features can be highlighted in various ways). In this regard, it should be noted that POVRAY script files have a stringent syntax, but do not require the placement of individual line parameters in a defined column. This allows for indentation of file lines in order to keep track of, for example, opened and closed parentheses.

PROGRAM INTEGRATION FOR IMAGE RENDERING AND PROCESSING

Figure 1 presents an outline of POVMENU options for accessing various subdirectories containing script (*.pov), include (*.inc) and rendered image (*.tga and *.gif) files and programs. The contents of each file subdirectory can be listed with LIST (not indicated in Figure 1). Following conversion of the PDB or other files to POVRAY script files, they are saved to the corresponding subdirectory. In POVMENU, the active script file is first indicated along with the selection of anti-aliasing (see later section) and image resolution. The latter includes two preview modes (160 × 100 and 320 × 200 pixels), as well as four higher screen resolution modes (320 × 200, 640 × 480, 800 × 600 and 1024 × 768 pixels). Access to an ASCII text file editor (QEDIT) is also provided for modifying the *.pov file

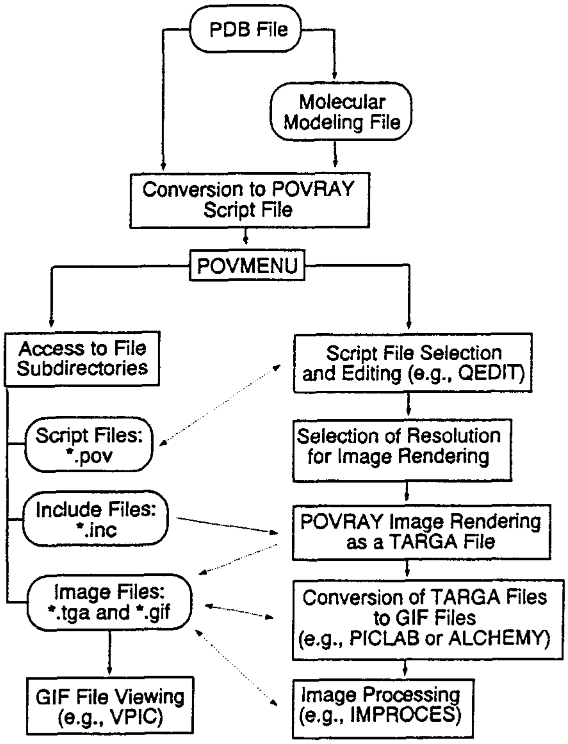
PDB File
Molecular Modeling File
Conversion to POVRAY Script File
POVMENU
Access to File Subdirectories
Script File Selection and Editing (e.g., QEDIT)
Script Files: *.pov
Selection of Resolution for Image Rendering
Include Files: *.inc
POVRAY Image Rendering as a TARGA File
Image Files: *.tga and *.gif
Conversion of TARGA Files to GIF Files (e.g., PICLAB or ALCHEMY)
GIF File Viewing (e.g., VPIC)
Image Processing (e.g., IMPROCES)

Figure 1. Outline of POVMENU functions for accessing and integrating files (rounded boxes) and programs (rectangular boxes) needed for image rendering and processing.

before or after rendering in order to "fine-tune" various input file parameters (camera or light source positions, rotations, translations, scalings, etc.). The preview modes for rendering images by POVRAY facilitate these modifications and the final high-resolution image is obtained as a 24-bit, true-color Targa (*.tga) file that is automatically saved to the image file subdirectory. These large graphic files can be significantly condensed by conversion to GIF files with little or no deterioration in the image quality. Both Targa and GIF files can be viewed individually, as well as cataloged, with VPIC. Finally, IMPROCES is employed as an image-processing program to add various enhancements in color, contrast, etc. This program is also useful for adding labeling to the final image.

RAY-TRACED MOLECULAR MODELS

Color Plate 1 represents a ray-traced rendering of epinephrine at a resolution of 800 × 600 pixels; both GIF and Targa files resulted in a similar image. This relatively simple structure was generated from the file presented in the appendix, which illustrates some basic points concerning input for POVRAY. For example, statements preceded by "//" or "/*" are ignored and are convenient for providing descriptive phrases. "Include" files (*.inc, Figure 1) operate to allow concise designations for shapes, colors or textures, following definitions in a *.inc file. For the appended file, these provisions are especially apparent for color, since this parameter is usually denoted as blends of red, green and blue, as exemplified by the last file statement for positioning and designating the white color of the light source (i.e., object {light_source {⟨0 20 -100⟩ color red 1 green 1 blue 1}}). However, following definition as a colors.inc file (#declare White = color red 1 green 1 blue 1), this statement is reduced to ". . . .texture{color White. . . . ," the color associated with the C atoms. The *phong* value, named for the developer (Phong Bui-Tuong) of an illumination model, adds a variable specular reflection to each atom. Color Plate 1 demonstrates two aspects of this model by a 90° rotation about the x-axis.

Color Plate 2 illustrates POVRAY renderings of a more complicated structure, insulin.[6] For this purpose, X-ray coordinate data (code 4INS) from the Protein Data Bank (PDB, National Biomedical Research Foundation, Brookhaven)[7] were incorporated into a POVRAY file. In brief, the complete PDB 4INS data set includes coordinates for two insulin molecules (I and II), wherein each molecule consists of two chains (A and B). Color Plate 2 (Inset A) is a ray-traced GIF file representation of the 420 centers of molecule I at a screen resolution of 800 × 600 pixels. In this example, each center is colored on the basis of atomic number. In contrast, differential coloration of the A and B chains of molecule I is shown in Color Plate 2 (Inset B), which was further modified to highlight the geometries associated with helical (residues 9–20) and sheet (residues 24–26) peptidyl segments of the B chain (Color Plate 2 [Inset C]). In addition, all of these color plates indicate the double sets of coordinates for four disordered residues of the B chain. These include Gln 4, Val 12, Glu 21 and Arg 22.[6]

IMAGE RENDERING TIMES

The time required for ray tracing a molecular structure is directly related to the complexity of the model (e.g., number of atoms, number of light sources, resolution, etc.), the use of anti-aliasing and computer processing speed. For example, the 17-atom structure representing epinephrine (Color Plate 1) at a resolution of 800 × 600 pixels required 44 mins. on a 80386DX/25 MHz computer; these structures were rendered in 6.5 mins. on a 80486DX2/50 MHz system. Similar rendering time improvements were found for the more complex insulin molecule (420 atoms; Color Plate 2)—approximately 8.75 hrs. and 1.25 hrs. for the 80386 and 80486 processors, respectively. It should be noted that these rendering times apply to the use of POVRAY in a DOS environment. On the average, rendering times were increased approximately twofold when ray tracing was performed as a DOS application in a Windows (enhanced mode) graphics user interface (GUI).

Anti-aliasing is a pixel supersampling algorithm designed to smooth the linear and curved contours of ray-traced objects. For POVRAY, the degree of smoothing is controllable. However, the use of this enhancement usually results in inordinate increases in rendering times and was not employed for the models displayed here. Whether or not smoothing is employed, an effective strategy for decreasing rendering times is to include all atom centers within a "bounded_by" shape. In this instance, ray interception by objects is tested only in this defined space. For densely packed and complex space-filling models, however, only minimal improvements would be expected.

In view of potentially lengthy rendering times for ray tracing complex molecular structures and the virtually limitless variables concerning display (camera and light positions, color, shape, texture, etc.), POVRAY has "preview mode" capabilities that allow visualization in a few minutes. Although of low resolution (160 × 100 or 320 × 200 pixels in POVMENU), the resulting image is of sufficient detail to sequentially test a number of variables in reasonably rapid succession.

IMAGE FILES

POVRAY-traced images are automatically saved as true-color, 24-bit Targa files, the only output format implemented by POVMENU. As expected, these files are quite large (e.g., each of the 800 × 600 pixel Targa files corresponding to Color Plates 1 and 2 contains more than 1.4 MB) and exceed the display capabilities of most desktop computers. To circumvent this limitation, PICLAB or ALCHEMY can be used for converting 24-bit Targa file output to 8-bit, 256-color GIF files. For these examples, corresponding GIF files were contained in less than 0.1 MB. In addition, IMPROCES offers a number of routines for displaying a 24-bit Targa file by using a color-reduced palette. Which of these options will provide the best quality is dependent on the image and is best determined empirically. As examples, the images in Color Plate 1 represent color-reduced Targa files, while Color Plate 2 illustrates the use of GIF files for visualization.

SUMMARY

This report demonstrates the feasibility of using existing ray-tracing software for obtaining high-quality molecular graphics from PDB and molecular-modeling output files on PCs. Although the examples provided only consisted of space-filling models, which do not require connectivities, it is also possible to generate stick (tubular), ball-and-stick as well as other structures. These more complicated applications will be described in future communications.

REFERENCES

1 Glassner, S.A. *An Introduction to Raytracing;* Academic Press: New York, 1989

2 Magnenat-Thalman, N. and Thalman, D. *Image Synthesis: Theory and Practice,* Springer-Verlag: New York, 1987

3 Kim, S., Yoon, C.W., Mhin, B.J., Kim, H.S. and Kim, K.S. Photorealistic image generation of molecular structure on PC screen using the ray-tracing technique. *J. Mol. Graphics* 1992, **10,** 218–221

4 The POVRAY program is available as freeware and was written by the POV development team, headed by D. Wells. It is based on DKBTrace (Version 2.12) developed by D. Buck and A.A. Collins

5 Wegner, T. *Image Lab;* Waite Group Press: Corte Madera, CA, 1992

6 Baker, E.N., Blundell, T.L., Cutfield, J.F., Cutfield, S.M., Dodson, E.J., Dodson, G.G., Crowfoot Hodgkin, D.M., Hubbard, R.E., Isaacs, N.W., Reynolds, C.D., Sakabe, K., Sakabe, N. and Vijayan, N.M. The structure of 2Zn pig insulin crystals at 1.5 angstroms resolution. *Philos. Trans. R. Soc. London* 1988, **B 319,** 369–456

7 Bernstein, F.C., Koetzle, T.F., Williams, G.J.B., Meyer, E.F., Brice, M.D., Rodgers, J.R., Kennard, O., Shimanouchi, T. and Tasumi, M. The Protein Data Bank: A computer-based archival file for macromolecular structures. *J. Mol. Biol.* 1977, **112,** 535–542

APPENDIX

// POVRAY file for ray-tracing Epinephrine

```
// Include files
#include “shapes.inc”
#include “colors.inc”
#include “textures.inc”
// Camera
camera {location ⟨0 0 -5⟩ direction ⟨0 0 1.1⟩ up ⟨0 1 0⟩ right ⟨1.33333 0 0⟩ look_at ⟨10 5 6⟩}
//C-atom coordinates
object {sphere{ ⟨12.528 4.598 5.572⟩ 1.85 } texture{color White phong 1}}
object {sphere{ ⟨11.779 3.887 6.561⟩ 1.85 } texture{color White phong 1}}
object {sphere{ ⟨10.359 3.91 6.566⟩ 1.85 } texture{color White phong 1}}
object {sphere{ ⟨9.758 4.738 5.577⟩ 1.85 } texture{color White phong 1}}
object {sphere{ ⟨10.497 5.372 4.58⟩ 1.85 } texture{color White phong 1}}
object {sphere{ ⟨11.886 5.277 4.566⟩ 1.85 } texture{color White phong 1}}
object {sphere{ ⟨13.979 4.542 5.57⟩ 1.85 } texture{color White phong 1}}
object {sphere{ ⟨14.584 5.178 6.995⟩ 1.85 } texture{color White phong 1}}
object {sphere{ ⟨16.532 5.999 8.125⟩ 1.85 } texture{color White phong 1}}
//N-atom coordinates
object {sphere{ ⟨15.912 5.629 6.741⟩ 1.75 } texture{color blue 1 phong 1}}
//O-atom coordinates
object {sphere{ ⟨14.454 5.355 4.556⟩ 1.65 } texture{color red 1 phong 1}}
object {sphere{ ⟨8.336 4.771 5.571⟩ 1.65 } texture{color red 1 phong 1}}
object {sphere{ ⟨9.751 3.409 7.571⟩ 1.65 } texture{color red 1 phong 1}}
//H-atom coordinates
object {sphere{ ⟨13.754 5.965 4.286⟩ 1.0 } texture{color Cyan phong 1}}
object {sphere{ ⟨9.318 4.113 8.185⟩ 1.0 } texture{color Cyan phong 1}}
object {sphere{ ⟨7.996 5.347 6.417⟩ 1.0 } texture{color Cyan phong 1}}
object {sphere{ ⟨15.938 6.427 6.186⟩ 1.0 } texture{color Cyan phong 1}}
// Light
object {light_source {⟨0 20 -100⟩ color red 1 green 1 blue 1}}
```