INSTITUTE OF PHYSICS PUBLISHING MODELLING AND SIMULATION IN MATERIALS SCIENCE AND ENGINEERING

Modelling Simul. Mater. Sci. Eng. **11** (2003) 173–177 PII: S0965-0393(03)57873-1

AtomEye: an efficient atomistic configuration viewer

Ju Li

Department of Materials Science and Engineering, Ohio State University, Columbus, OH 43210, USA

Received 10 December 2002
Published 30 January 2003
Online at stacks.iop.org/MSMSE/11/173

Abstract

AtomEye is free atomistic visualization software for all major UNIX platforms. It is based on a newly developed graphics core library of higher quality than the X-window standard, with area-weighted anti-aliasing. An order-N neighbourlist algorithm is used to compute the bond connectivity. The functionalities of AtomEye include: parallel and perspective projections with full three-dimensional navigation; customizable bond and coordination number calculation; colour-encoding of arbitrary user-defined quantities; local atomic strain invariant; coloured atom tiling and tracing; up to 16 cutting planes; periodic boundary condition translations; high-quality JPEG, PNG and EPS screenshots; and animation scripting. The program is efficient compared to OpenGL hardware acceleration by employing special algorithms to treat spheres (atoms) and cylinders (bonds), in which they are rendered as primitive objects rather than as composites of polygons. AtomEye can handle more than one million atoms on a PC with 1 GB memory. It is a robust, low-cost tool for surveying nanostructures and following their evolutions.

1. Introduction

Modelling physical processes at the nanoscale is ever more important as experimental techniques such as atomic force microscopy (AFM) and high-resolution transmission electron microscopy (HRTEM) overlap in lengthscale with which achievable in direct atomistic calculations [1]. In a way, visualization of simulation results plays the same role as microscopy in experiments: we rely on it to extract mechanistic information, before we decide what to do next (more calculations or experiments) to accentuate a certain finding or theory. In this feedback, the capabilities and ease-of-use of the visualization software employed are actually important towards scientific discovery, even though one may initially consider it as merely a technical detail.

Very powerful commercial software packages such as Materials Studio™ and ChemOffice™ are currently available, but their costs of entry are quite high. Here I would like to introduce AtomEye, a low- or no-cost general visualization tool whose purpose is similar to that of RASMOL [2], a ground-breaking software that has broad impact, although

manifesting deficiencies at handling large number of atoms ($>50\,000$). At present, there are high-quality visualization programs such as XCrySDen [3] for displaying crystal structures and electron densities, MOLSCRIPT [4] for creating vector graphics outputs, and Raster3D [5] for ray-tracing still-frames, but AtomEye's main function is closer to that of RASMOL and XMol [6], which is *real-time surveying of large-scale atomic structures*. A leader in this aspect is VMD [7] which is based on OpenGL-driven hardware acceleration [8].

2. Concepts

For *general-purpose* real time three-dimensional graphics such as those used in computer games, there is no question that hardware acceleration, taking advantage of the modern graphics cards, is the best solution. Currently, this technology is mainly based on the assemblage of polygons, whose transform and lighting, vertex shading and texture mapping, etc are all performed in the graphics pipelines (hardware). Yet for certain categories of atomistic visualization, if almost all one renders are *spheres* (for atoms) and *cylinders* (for bonds), then the above approach may not be the optimal solution. Particularly, a sphere requires a large number of polygons ($\sim$20) to represent, which leads to drawbacks in both speed (when there are $\sim 10^6$ atoms on the scene) and quality.

An alternative approach is to regard spheres and cylinders as *primitives* and use specialized algorithms to render them directly without first decomposing them into polygons. In AtomEye, it is done by memory caching, where at the start of the program one uses mathematical formulae to shade a range of spheres in very high-quality pixel maps with area-weighted anti-aliasing at the border [9], keep these pixel maps in the main memory, and in use simply paste the maps to the scene with z-buffering. Because only simple operations (mostly memory copying) are performed in real-time, this is very fast. Also, the spheres and cylinders have to pass a few fast visibility screening tests before being committed to the z-buffer. In scenes with $>10^4$ spheres, this approach has been confirmed to be competitive compared to the general hardware acceleration approach using OpenGL [8], and it gives overall higher quality outputs [10] even though the lighting and shading is not 100% physical. Graphics card technology has evolved, but so has CPU/main memory, as they are from the same microfabrication technology. In the foreseeable future, it is unlikely that this approach loses to the hardware acceleration approaches which perform general polygon assemblage.

X-window [11] is the common graphics standard on UNIX and Linux platforms. But its core drawing functions are severely outdated. I have rewritten all two-dimensional drawing and clipping functions (for displaying wireframes, etc in AtomEye) with geometrically exact area-weighted anti-aliasing [9], and the entire scene image is generated in the main memory first without calling any of the old X-window drawing functions. X-window is used only as an input-events handle and for remote display over network. If shared memory extension (for local display) is available, then it is invoked automatically. It also queries and handles different display colour depths and big/small endian issues automatically.

The Protein Data Bank (PDB) [12] atomic structure file format, though widespread with great impact, does have drawbacks such as fixed (and therefore possibly insufficient) digits/accuracy, no velocity specification standard (for molecular dynamics), and is uncompressed. AtomEye supports PDB format, but also a new atomic configuration file format CFG [10] which addresses the above problems. All input configuration files can be compressed by gzip [13] or bzip2 [14]; AtomEye recognizes compression by magic numbers and automatically decompress the file (temporarily, without the user's notice) by executing appropriate command, if available, in the user's shell environment. By default, the program assumes the configuration to be under periodic boundary condition (PBC). The

internal representation in AtomEye is that of an $\boldsymbol{H}$-matrix specifying the three edges of the PBC box, with reduced coordinates $0 \leqslant s_{i\alpha} < 1$ specifying the atom positions. The bond connectivity is computed using an order-N neighbourlist algorithm [15], in which the supercell is partitioned into bins chosen such that only atoms in the nearest-neighbour bins can form bonds. The bin–bin, bin–atom, atom–bin, and atom–atom lists are maintained as the atoms drift and the supercell varies in shape and volume.

AtomEye combines mouse, hot-key and terminal input controls. It supports simultaneous multiple views of the same configuration in different resizable windows by employing lightweight POSIX threads [16]. Both parallel and perspective projection methods are supported and the user may navigate easily to anywhere inside or outside of an atomic configuration and look in any direction. As such, it is ideal for tracking and studying defects in the configuration. AtomEye can produce screenshots in PNG [17], JPEG [18] and EPS [19] file formats, the latter being a simple wrapping for JPEG compression. Because the scene image is created in the main memory first, its resolution is not limited by that of the screen, and high-quality screenshots such as in 2400×2400 pixel resolution is usually sufficient for even the most demanding paper publishing needs, while the JPEG compression does a good job in keeping the file small with little loss in quality.

AtomEye provides many customizable functions to identify and accentuate certain structural characters of an atomistic configuration. First, the coordination numbers of atoms are automatically computed and can be used for colour-encoding; and one can easily make atoms with certain coordination number invisible, so only the crystalline defects are rendered. Second, it can compute the local atomic strain [15], and use the tensor invariants for colour-encoding: this is good for visualizing the long-range elastic fields of and elastic interactions among defects. Third, AtomEye can dye an initial configuration, then track the coloured atoms in subsequent configurations. This is useful for visualizing diffusion- and/or deformation-induced atomic displacements. Finally, the user may save an arbitrary set of calculated atomic properties (potential energies, kinetic energies, local atomic stresses, local atomic moduli, etc) in the CFG file or in separate atomic property 'patch' files, and AtomEye will be able to render those properties as well, in many different colour-mapping schemes. Also, one can arbitrarily translate the configuration under PBC, and specify up to 16 cutting planes. After interactively finding a good vantage point and choosing other rendering options, AtomEye can render a sequence of configuration files and save the frames in JPEG format according to a user-supplied script, which can then be further compressed into a movie.

3. Features

- Order-N in both execution time and memory used, where N is the number of atoms; designed for condensed-matter systems, no problem with >1 million atoms.
- Auto-detect 8-, 16- and 32-bit shared memory or remote X-displays.
- Multiple resizable windows in POSIX threads; 0% CPU usage if not moving.
- Geometrically exact area-weighted anti-aliasing for atoms, bonds, and wireframes.
- Fast rendering of atoms by caching pixel- and z-maps in the main memory.
- Quick toggle between parallel and perspective projections.
- Full three-dimensional navigation.
- Support PBC.
- Support PDB input file format.
- Support arbitrary-precision and extensible CFG input file format for large-scale, reloadable molecular dynamics simulations.
- Auto-decompress gzip- or bzip2-compressed input configuration files.

- JPEG, PNG, and high-resolution EPS screenshots.
- Customizable atom radii and colouring schemes.
- Coordination number colour-encoding with customizable cutoff radii and invisibility controls.
- Local atomic von Mises shear strain invariant colour-encoding.
- User-defined property colour-encoding, in hsv, jet, and other colour-mapping choices.
- Colour-marking an initial configuration to track subsequent atomic displacements.
- Cooperative X-terminal input with GNU readline/history.
- Up to 16 arbitrary cutting planes with advancing/rotation/flipping controls.
- Animation script for making movies.

AtomEye has been extensively tested for more than two years by an active group of modelling researchers. It is currently being integrated into the Parallel Molecular Dynamics Stencil project [20], a library of MPI [21] functions for doing atomistic simulations in shared-memory multiprocessor or Beowulf [22] parallel computers. AtomEye itself will be parallelized in the near future in a computing-slaves/graphics–display–server paradigm. Even on an ordinary serial Linux PC with 1 GB memory, AtomEye has no problem rendering configurations with $\sim$1.5 million atoms.

4. Access

AtomEye binary executables for Linux (Intel i686, Compaq Alpha), SGI IRIX/IRIX64, Sun Solaris, HP UX, Compaq Alpha Tru64 UNIX, and Microsoft Windows PC (Cygwin [23] and a working X-server such as [24] should be installed first) are available free of charge at [10]. Source code access and other binary builds can be arranged by contacting the author at liju99@alum.mit.edu.

Acknowledgments

I would like to thank Hideo Kaburaki, Hajime Kimizuka, Futoshi Shimizu, Dongyi Liao, Wei Cai, Antonino Romano, Dongsheng Xu, Jinpeng Chang, Krystyn J Van Vliet, Ting Zhu, Shigenobu Ogata, Marjorie Bertolus, and Joao F Justo for providing building platforms and making insightful suggestions, a lot of which have already been implemented in the AtomEye feature set. I would like to thank Sidney Yip for his mentoring and encouragements.

References

[1] Li J, Van Vliet K J, Zhu T, Yip S and Suresh S 2002 Atomistic mechanisms governing elastic limit and incipient plasticity in crystals *Nature* **418** 307–10
[2] Sayle R A and Milner-White E J 1995 RASMOL: biomolecular graphics for all *Trends Biochem. Sci.* **20** 374–6
[3] Kokalj A 1999 XCrySDen—a new program for displaying crystalline structures and electron densities *J. Mol. Graphics Modelling* **17** 176–9
[4] Kraulis P J 1991 MOLSCRIPT: a program to produce both detailed and schematic plots of protein structures *J. Appl. Crystallogr.* **24** 946–50
[5] Merritt E A and Bacon D J 1997 Raster3D: photorealistic molecular graphics *Methods Enzymol.* **277** 505–24
[6] XMol, version 1.3.1, Minnesota Supercomputer Center, Inc., Minneapolis, MN, 1993
[7] Humphrey W, Dalke A and Schulten K 1996 VMD—visual molecular dynamics *J. Molec. Graphics* **14** 33–8
[8] Shreiner D (ed) 1999 OpenGL architecture review board *OpenGL(R) Reference Manual: The Official Reference Document to OpenGL, Version 1.2* 3rd edn (Reading, MA: Addison-Wesley)
[9] Foley J D, van Dam A, Feiner S K and Hughes J F 1995 *Computer Graphics: Principles and Practice* 2nd edn in C (Reading, MA: Addison-Wesley)

[10] The AtomEye home page, http://alum.mit.edu/www/liju99/Graphics/A/ or http://www.google.com/search?q=AtomEye
[11] Scheifler R W and Gettys J 1986 The X window system *ACM Trans. Graphics* **5** 79–109
[12] Westbrook J *et al* 2002 The Protein Data Bank: unifying the archive *Nucleic Acids Res.* **30** 245–8
[13] The gzip home page, http://www.gzip.org/
[14] The bzip2 and libbzip2 home page, http://sources.redhat.com/bzip2
[15] Li J 2000 Modeling microstructural effects on deformation resistance and thermal conductivity *PhD Thesis* Massachusetts Institute of Technology
[16] Richard Stevens W 1993 *Advanced Programming in the UNIX(R) Environment* (Reading, MA: Addison-Wesley)
[17] Portable Network Graphics (PNG) image file format specification, http://www.libpng.org/
[18] Joint Photographic Experts Group (JPEG) image file format specification, http://www.jpeg.org/
[19] 1992 *Encapsulated PostScript File Format Specification, Version 3.0* Adobe Systems Incorporated, San Jose, CA
[20] The Parallel Molecular Dynamics Stencil home page, http://stencil.koma.jaeri.go.jp/
[21] MPI—The Message Passing Interface Standard, http://www-unix.mcs.anl.gov/mpi/
[22] A Beowulf-class PC cluster running MOSIX, http://wulf.mit.edu/
[23] Cygwin is a UNIX environment for Microsoft Windows, http://www.cygwin.com/
[24] Xmanager is X window server software running on Microsoft Windows, http://www.netsarang.com/