Brian R Kent

Chapter 1

Introduction

1.1 Visualization in the sciences

Science is all-encompassing with its ability to provide a logical path to facts, revolutionize technology and change the world as we know it, as well as inspire the next generation of technological advancements. Through the scientific method we form questions and develop hypotheses, and visualization of data plays a key role in testing and analysis. Insight into experiments is often found by how we present scientific and engineering results, concepts and ideas to all audiences, both non-scientific and scientific.

Modern computing has given the scientific community tools to take data exploration to the next level. The mining of large data sets requires efficient data organization as well as visualization. At times this is due to the sheer disk size of data, limited by memory or processing power. Developments in graphics processing units (GPUs), both for driving high-end graphics applications and threaded processing, allow for sophisticated high resolution images and animations to be created. In other scenarios the data may have a complex phase space to explore. Data may have N-dimensional tables or correlations among quantities that only reveal themselves through 3D visuals.

Many cutting-edge studies in a variety of scientific disciplines benefit from scientific visualization.

Astronomy. Telescope observations across the electromagnetic spectrum now generate terabytes of data every hour. Storing and managing the raw data is a challenge, let alone visualizing the processed data that have been pushed through the necessary calibration procedures. A survey of a million galaxies from a single wavelength regime will have metadata that need to be appropriately organized and indexed to search efficiently [1, 2]. Wide-band radio spectroscopy with radio telescopes gives 2D information about emission from the sky as well as frequency information along a third axis [3].

Physics. Cutting edge projects such as the Large Hadron Collider require advanced visualization software [4]. Theoretical models in particle physics, lattices in solid state physics, the physics of high-temperature plasmas

doi:10.1088/978-1-6270-5612-0ch1

and gravitational waves all benefit from new visualization techniques [5]. Visualizing scientific apparatus before they are designed can help to optimize the engineering and design of experiments.

Chemistry/Biology. Complex molecules and molecular dynamics during reactions can be examined with 3D animations [6]. Studying proteins can reveal their structure using data from nuclear magnetic resonance (NMR) studies. GPU-accelerated processing has allowed for scalable solutions for visualizing complex viruses [7].

Geography/Planetary Science. A wealth of mapping data exist, not only for the planet Earth, but other surveyed planets in our Solar System. Using special data storage models and GPU processing, 3D maps of planetary surfaces can now be rendered in real time [8].

Medicine. Visualizing computed axial tomography (CAT scans) can show a transparent view of organic structures. Classification and feature extraction while performing diagnoses benefits greatly from be able to view real time 3D movies of internal organs [9].

1.2 What is Blender?

Blender is open source software that allows a user to create high quality animations of 3D models and data. It has wide usage in the video game and entertainment industries. The software is also extremely useful for generating high quality scientific visualizations. With its well organized Python application programming interface (API), it can be scripted to load data from numerical simulations. The power of Blender becomes evident when the user is given complete control over the camera angle, field of view and rendering aspects of the final animation.

Blender's traditional user base has been 3D graphics specialists working in modeling and animation. However, with the intuitive graphical user interface (GUI) and availability of Python libraries for reading a variety of scientific data, Blender brings an exciting and unique visualization suite to the modern scientific workbench. Developer/creator Ton Roosendaal and the Blender Foundation have created a community to maximize the amount of on-line material for developers and users[1]. The goal of this book is to provide the reader with an interesting and practical introduction to Blender by way of science examples, each showing important software features and 3D graphics techniques.

Figure 1.1 shows the structure of Blender and its main capabilities. This book will examine each of the following topics and then use them in concert together for a number of example visualization projects:

- Meshes and models
- Lighting
- Animation
- Camera control
- Scripting
- Composites and rendering

[1] http://www.blender.org/forum/

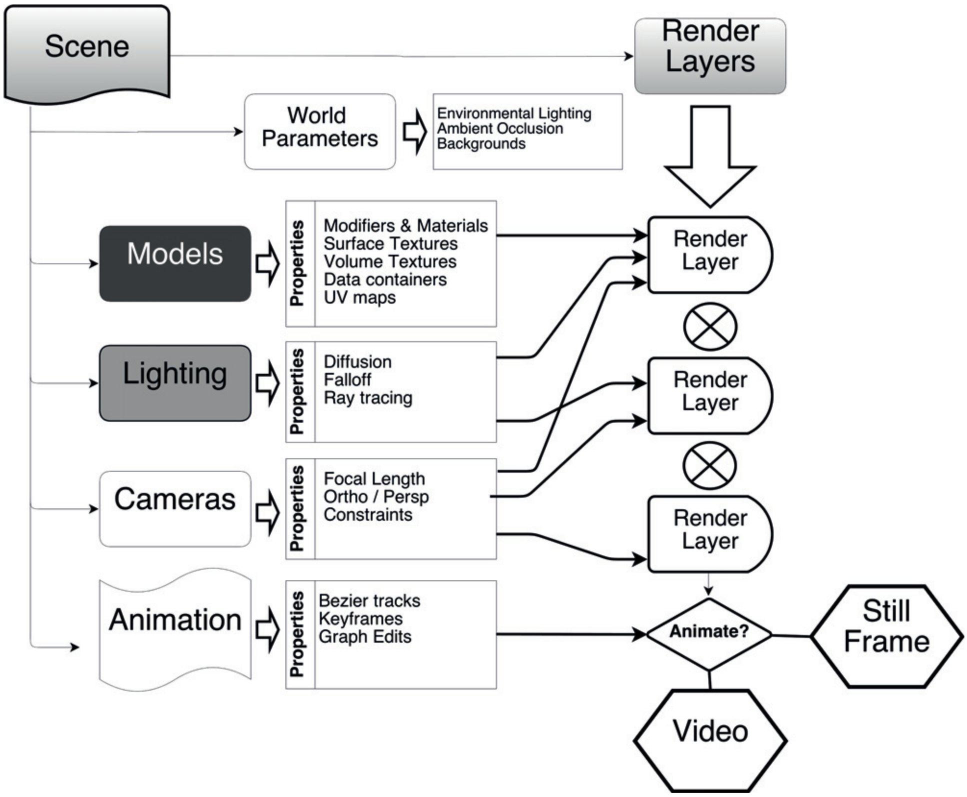

Figure 1.1. This flow chart shows how a Blender workflow can be used in scientific visualization. Different aspects of the Blender GUI and API will be analyzed throughout the book to show how various pieces can fit together in a project [10]. Copyright 2013 Brian R Kent, publications of the Astronomical Society of the Pacific.

There are also external rendering engines that can be used with Blender. While the examples and exercises in this book will use the internal Blender engines Render and Cycle, the reader is encouraged to examine other rendering options for different scenarios[2].

Blender can take advantage of multi-core central processing units (CPUs). The software also provides hooks for NVidia CUDA and OpenCL utilizing accelerated GPU rendering. This can decrease rendering times by a factor of three or four [10].

Blender also contains two powerful tools for building visualizations—the node compositor and video sequencer. The node compositor allows multi-layered visualizations to be combined into a final animation, through the use of nodes—a visual type of flow-chart style access to the Blender API and functionality. The video sequencer is useful for editing and combining animations for presentation.

[2] http://www.blender.org/download/external-renderers/

The software has modest desktop and laptop hardware requirements, depending on the level of complexity for the visualization. The software runs equally well on Windows, Mac OS X and many flavors of Linux. Hardware requirements, on-line help and documentation, along with the software itself can be found at http://www.blender.org.

Regardless of whether a desktop or laptop computer is used, it is highly recommended that a three-button mouse with a scroll wheel be used in Blender sessions. The numerical keypad on a standard keyboard also comes in handy when manipulating the 3D view, which will be described in a later section. If the user is on a laptop, then the top row numerical keys can be configured in the menu File → User Preferences → Input and by checking 'Emulate Numpad'. We will cover how these numerical keys are used in a later section.

1.3 Rendering engines

While the vast majority of visualizations can be handled with the internal Blender rendering engine and its next-generation engine Cycles, a number of popular third-party applications are worth exploring.

Render. Most general purpose data visualizations that scientists will encounter can be found with the default renderer. This engine supports GPU processing and volumetric rendering. http://wiki.blender.org/index.php/Doc:2.6/Manual/Render

Cycles. The rendering path can be configured within the node-based compositor in Cycles, allowing for rapid prototyping of a scene. More realistic caustics are added with its rendering algorithm. http://wiki.blender.org/index.php/Doc:2.6/Manual/Render/Cycles

LuxRender. This open-source third party rendering engine has plug-ins for Blender and achieves fast ray tracing with GPU processing (see the references in [11]; http://www.luxrender.net/).

Mitsuba. A physically based rendering engine that can handle volume rendering. http://www.mitsuba-renderer.org/

YafaRay. Ray tracing and the ability to save files to high dynamic range images formats are features of Yafaray (EXR [12], http://www.yafaray.org/).

1.4 Community support

Blender has wide ranging community support, knowledge-bases and forums to help both new users starting out and advanced Blender aficionados looking to take their 3D visualization skills to the next level. Some popular websites include:

Blender Guru http://www.blenderguru.com/
Blender Artists http://blenderartists.org/
NASA 3D Model repository http://nasa3d.arc.nasa.gov/
BlenderNation http://www.blendernation.com/
BlenderCookie http://cgcookie.com/blender

1.5 Types of data visualization in the sciences

There are different kinds of scientific visualization to consider, each presenting unique technical scenarios when working with 3D graphics [13]. Each of the visualization types will be utilized to explain and expose useful features of Blender.

Solid models/surfaces/rigid body simulations. Surfaces work well for 2D waveforms, rigid body mechanics and 3D contour surfaces. They are also useful when working with geographic information system (GIS) maps. These types of visualizations are often textured to mimic a real world surface and externally illuminated. If glass or reflective surfaces are used, then ray tracing will increase the rendering time.

Data cubes/transparent/translucent rendering. If data are 3D and need to be rendered above some given noise level with a transfer function, then transparent rendering can be used. This is useful in medical and astronomical imaging. For medical imaging, the investigator wants to see multiple layers from a CAT scan. With astronomical imaging, the 3D nature of the data cube reveals positions on the sky as well as a Doppler shifted frequency, which can show the dynamics of rotating objects.

3D catalogs. These kinds of visualizations are akin to 3D scatter plots. They are useful for catalogs with three parameters in basic curvilinear coordinates, including Cartesian, cylindrical, or spherical systems.

***N*-body simulations.** These are useful for displaying the results of gravitational interactions between point masses. The data structure utilization in Blender is similar to that for catalogs, except that *shape keys* are used to animate the simulation. Shape keys are used to increase the efficiency and reduce the memory footprint of the metadata in the Blender file.

Soft body simulations. These are used for simulating internally the physics of deformable objects within Blender. The results are often use to simulate the look of said objects visually and are useful for numerical simulations that cannot be solved analytically.

Surface/terrain maps. GIS maps can be loaded and color coded to show differences in terrain. Vertex painting can be utilized to overlay various features and color coded maps. An example will be given in a later chapter using Martian mapping data.

Phenomenological models. Blender has some utility in creating molecular models, as seen with the BioBlender project[3] [14]. Chemical interactions can be animated and studied.

This book introduces important Blender concepts and, along the way, shows users the critical parts of the interface that they will need to know to accomplish their visualizations.

[3] http://www.bioblender.eu/

Bibliography

[1] Szalay A S 2002 The SDSS Skyserver: public access to the sloan digital sky server data, arXiv: cs/0202013

[2] Gray J 2002 Data mining the SDSS Skyserver database, arXiv: cs/0202014

[3] Kent B R 2013 Visualizing scientific graphics for astronomy and astrophysics *Proc. ACM SIGGRAPH'13 (Anaheim, CA, 21–25 July 2013)* (New York: ACM) article no. 113

[4] Fedorko I, Lefebure V, Lenkes D and Pera M O 2012 New data visualization of the LHC Era Monitoring (Lemon) system *J. Phys.: Conf. Ser.* **396** 042019

[5] Sanders B C, Senden T and Springel V 2008 Focus on visualization in physics *New J. Phys.* **10** 125001

[6] Moreland J L, Gramada A, Buzko O V, Zhang Q and Bourne P E 2005 The Molecular Biology Toolkit (MBT): a modular platform for developing molecular visualization applications *BMC Bioinform* **6** 21

[7] Stone J E, Vandivort K L and Schulten K 2013 GPU-accelerated molecular visualization on petascale supercomputing platforms *Proc. 8th Int. Workshop on Ultrascale Visualization (Denver, CO, 17 November 2013)* (New York: ACM) 6

[8] Myint S, Jain A, Cameron J and Lim C 2011 Large terrain modeling and visualization for planets *4th IEEE Int. Conf. on Space Mission Challenges for Information Technology (SMC-IT) (Palo Alto, CA, 2–4 August 2011)* (Piscataway, NJ: IEEE) 177–83

[9] Linsen L, Hagen H, Hamann B and Hege H-C (ed) 2012 *Visualization in Medicine and Life Sciences* vol 2 (Berlin: Springer)

[10] Kent B R 2013 Visualizing astronomical data with Blender *Publ. Astron. Soc. Pac.* **125** 731–48

[11] Price R L, Puchala J C, Rovito T V and Priddy K L 2011 Physics accurate layered sensing model *Proc. 2011 IEEE National Aerospace and Electronics Conference (NAECON) (Dayton, OH, 20–22 July 2011)* (Piscataway, NJ: IEEE) 291–6

[12] Debevec P 1998 Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography *Proc. 25th Annual Conf. on Computer Graphics and Interactive Techniques, Orlando, FL, 19–24 July 1998* (New York: ACM) 189–98

[13] Friendly M 2006 A brief history of data visualization *Handbook of Computational Statistics: Data Visualization* vol 3, ed C Chen, W Härdle and A Unwin (Heidelberg: Springer)

[14] Andrei R M, Callieri M, Zini M F, Loni T, Maraziti G, Pan M C and Zoppè M 2012 Intuitive representation of surface properties of biomolecules using BioBlender *BMC Bioinform.* **13** S16