

Evaluating Second Life as a tool for collaborative scientific visualisation.

Paul Bourke

WASP, University of Western Australia
M024, 35 Stirling Hwy
Crawley, West Australia 6009
Phone: +61 8 64888097
paul.bourke@uwa.edu.au

Abstract

This paper explores the potential of using the online digital world Second Life as a scientific visualisation tool, in particular, for remote collaborative exploration of scientific datasets. Second Life at its core provides a means whereby multiple remote participants can engage with 3D geometry within a virtual environment. It has been chosen for this evaluation for a number of reasons, these include the easy to learn user interface, its relatively widespread uptake, the availability of the player on the three main computer platforms, its non-aggressive social networking foundation, and the scripting capability. This paper will outline the desirable characteristics expected of any online collaborative tool in science research and will discuss, through examples, the extent to which Second Life meets those expectations.

Keywords

Scientific visualisation, game engine, multiplayer, immersion, Second Life.

1. Introduction

Scientific visualisation is the process whereby computer graphics is employed to present data in such a way as to reveal new relationships and improve understanding of the underlying physical processes. The main sources of data in the sciences arise from experimental acquisition, the output from numerical simulations, or from the mathematics describing physical processes. It often employs leading edge algorithms and specialist hardware to present results in such a way as to maximise the interpretive opportunities. While much of the scientific visualisation process occurs in the confines of the researchers office or in a centralised visualisation laboratory, increasingly the need arises for the visualisation to be a shared experience between a distributed and remote group of collaborators [1,2]. The software tools available for visualisation very rarely support any form of remote collaborative experience. There have been some attempts at this [3] but they have either not escaped the research/exploratory stage, have depended on proprietary hardware/software, are high cost commercial products, are only available for a limited range of hardware platforms and operating systems, or are rather difficult to install and operate. As a result, collaborative visualisation has been relegated to experiences based upon the sharing of images and movies. Even these 2D projections generally do not form a real-time collaborative experience but a delayed in time file exchange through email and web servers. Video conferencing for collaborative research is relatively common in many industries including the sciences but shared white boards and video do not provide the 3D and

interactive experiences those employing visualisation within their home institutions expect, they are certainly inferior experiences to those the researcher is able to engage in with local collaborators.

In contrast, multiplayer games are expected to be cross platform and easy to install. By their very nature, at least for first person shooters, they allow the participant to engage naturally and interactively with 3D geometry within a virtual world. Additionally they tend to exploit the capabilities of modern graphics cards to achieve the highest visual quality at a target frame rate. There have been various investigations into the use of game engines for a range of visualisation problems [4] but rarely have these involved scientific data visualisation, distinguished by the need to convert data into geometry, colour, and opacity. The majority of these explorations have targetted teaching applications [5] or deal with the presentation of data to a wider audience in an offline fashion rather than to collaborators in an active concurrent experience. Such applications, often referred to as illustrative visualisation, are able to involve a separate modeling/design stage. In contrast, for the data visualisation applications contemplated here there is a need to automatically convert data and generate the visualisation in a live fashion.

The proposal discussed here then is whether Second Life, a readily accessible multiplayer environment in which data can be represented as 3D geometry within a virtual world, can be employed to benefit a collaborative scientific visualisation process. The goal of such a process is to enable remotely located researchers to occupy one virtual space where they all experience, explore, and discuss their datasets. There have been attempts at collaborative visualisation using Second Life, but most relate to information technology, business, and architectural visualisation. One of the few attempts at using Second Life for scientific data visualisation has been a weather visualisation [6] system that additionally took live data feeds from weather stations, this was not intended as a collaborative experience but rather an in-game presentation of weather data.

The following are considered to be desirable characteristics for any software seeking to support the collaborative scientific visualisation process.

1. One must be able to input either raw data or preprocessed representations into the environment and as automatically as possible turn that data into visual entities.
2. Visualisation very often requires a large number of geometric primitives; as such any proposed software must be efficient and utilise current algorithms and exploit the capabilities of modern graphics cards.
3. Visualisation employs visually appealing graphics, this includes a rich set of geometric primitives from which to represent data and the ability to map colour, texture, and opacity to that geometry.
4. The geometry forming the visualisation needs to be able to be explored within a virtual 3D environment in an intuitive way. The ownership of the data assets needs to support both a private and shared/editable mode.
5. For collaborative visualisation one is expecting primarily to be able navigate and explore the data in a virtual environment with the remote collaborators represented as avatars. One would also expect to be able to engage in more conventional form of interactions at the same time such as video conferencing, voice, and text chat.

6. Any proposed software needs to be supported on multiple hardware and operating systems in order to maximise the chances of it being available on the desktop of any particular collaborator.

2. Evaluation

Second Life has been chosen for this investigation and it will be evaluated against the above criteria. It is a massively multiplayer environment with remotely managed assets accessed by software running on the local machine and taking advantage of, or compensating for, whatever graphics hardware is present. The evaluation largely consists of determining to what extent could Second Life represent some of the classical datasets found in the scientific visualisation context. This includes the representation of surfaces, colour mapping of additional data variables, geometry represented by high level primitives, and volumetric data. Of importance is the process of getting the data into the environment and the extent to which the visualisation processes can be performed using the internally available scripting language. Related to this is how automated these process can be made.

Second Life was additionally chosen because it is supported on both MSWindows, Mac OS-X, and Linux. While Mac OS-X is a small part of the total computer installed base, it has a significantly higher market share in the scientific community due to its UNIX underpinnings. Also, Linux has a large market share in the science industry for the same reason and because it is almost unrivaled as the operating system for high performance computing used in the sciences.

By its very nature any conclusions drawn from actual experiments made may be disputed by those capable of more sophisticated coding and the development of addons, plugins or modules. The aim here was to use the software in its unmodified form and at a level one might expect of a typical science researcher with average programming skills.

3. Results

There are a number of ways in which data can be imported into Second Life. The main approach tested here is to simply copy/paste data, represented as arrays, and then develop scripts within Second Life that turn that data into 3D representations. The arrays are themselves described as text files in the array declaration format dictated by the Second Life scripting language (SLSL) and therefore one would generally add this as an export format to upstream software involved in creating or manipulating the data. One serious limitation for this approach is the maximum size of the arrays allowed by the SLSL.

The major limitation with Second Life is the very low volume of data that can be housed per unit area of land. This is somewhat acceptable for most activity in Second Life that involves the manual creation of data (buildings, appearances, and vehicles) consisting of relatively small numbers of primitives with the semblance of detail by the judicious use of textures but the limits are woefully inadequate for most interesting datasets. As an example consider the representation of molecules. The molecule on the left in figure 1 has around 30 atoms and the one of the right 160 atoms. It was necessary to engage in tricks in order represent the later (the Aspirin molecule) and yet in molecular visualisation it would be considered a relatively simple example. The limits placed upon the amount of

geometry allowed are pragmatic ones arising from the need to retain an interactive experience while at the same time there is the need to transfer all geometric data between the remote server and the local workstation.

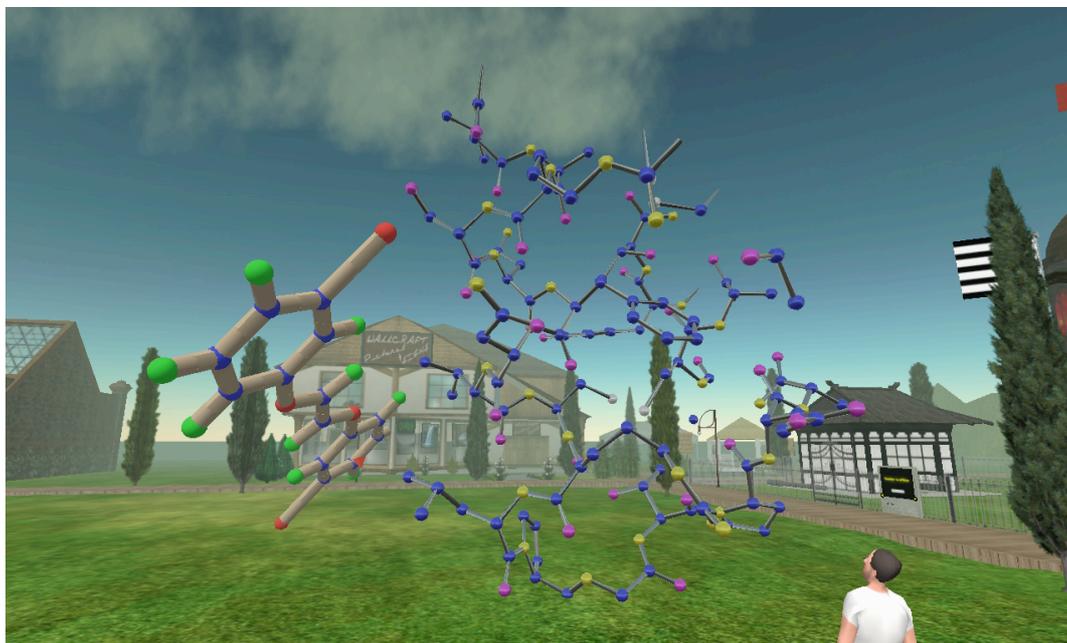


Figure 1. Representation of 3D molecules in Second Life, JAYFEV strand (left) and Aspirin (right).

The primitive set in Second Life with which data can be represented is certainly very rich, and would rival most 3D packages and APIs. The molecules in figure 1 are created using spheres and cylinders but there are also many other primitives each with a range of variables that can be used to create variations of the base geometry. An interesting challenge is volume rendering where at each point (voxel) in some 3D volume a scalar is known. This is a very common type of data and is encountered in almost every scientific discipline, a familiar medical example being MRI scans. Volume rendering involves mapping those scalars in each voxel to colour and transparency in order to reveal the structure and relationships within the volume. While there are more sophisticated algorithms, many volume visualisation packages simply map slices through the volume as textures onto planes. This is readily achieved in Second Life, in the example in figure 2 the textures for each plane are rendered externally including the transparency. These textures are uploaded and a SLSL script repeatedly adds planes to the scene and assigns the correct texture.

An obvious omission from the Second Life primitive set is a general mesh. This has serious limitations for representations of scientific datasets which very often involve surfaces constructed from arbitrarily orientated triangular polygonal elements. The closest thing Second Life provides is a sculpt map, an image where the red, green, blue values represent distances in polar coordinates. While this can represent some surfaces it is by no means meets the surface mesh requirement of many visualisation problems in the sciences. Figure 3 illustrates how sculpt maps can be used to represent spherical harmonics, solutions the Laplace equations in spherical coordinates. Note that although the sculpt map is an image it only defines the surface geometry, additional variables can still be mapped to the

surface such as a colour or bump map. If sculpt maps can represent the desired geometry then they are relatively straightforward to create and the sculpt map image can readily be added as an export format to other software.

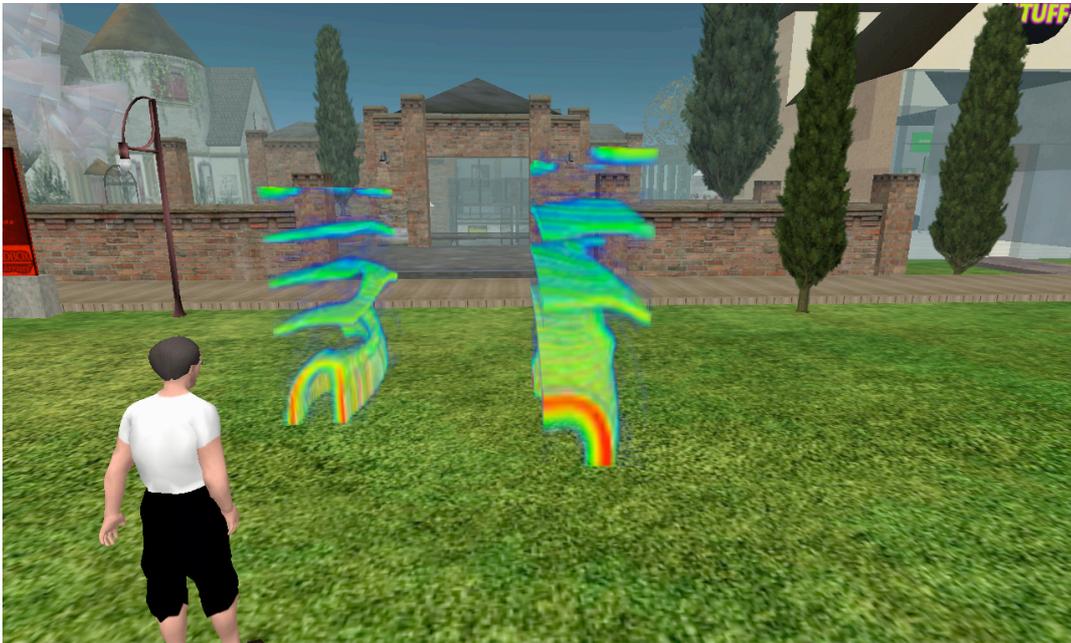


Figure 2. Volume visualisation of helix wave formation in fluid flow [7].



Figure 3. Representation of solutions to the Laplace equations in spherical coordinates.

One thing that Second Life does well is texture mapping, that is, a means of presenting apparently complex structure by using images rather than geometry. Modern graphics cards are optimised to handle these texture maps and Second Life exploits this capability. This strength can be employed by some visualisation tasks that involve more geometry than Second Life can handle by prerendering the

geometry to texture maps. For example, in figure 4 the visual field from within a crystal and its Hershfield surface is rendered as a full 360x180 spherical projection. This can then be applied to a sphere in Second Life and if the participants are located at the center of the sphere they can look around and experience being inside the molecule. This can result in a very powerful sensation of 3D geometry (figure 4) even though it is simply a texture mapped onto a sphere. A similar effect can be achieved as shown in figure 5, a panorama to assist in the visualisation of a proposed site for the SKA (Square Kilometer Array).



Figure 4. Preprocessed spherical projections from inside a crystal including the Hershfield surface, computed from Crystal Explorer [8].

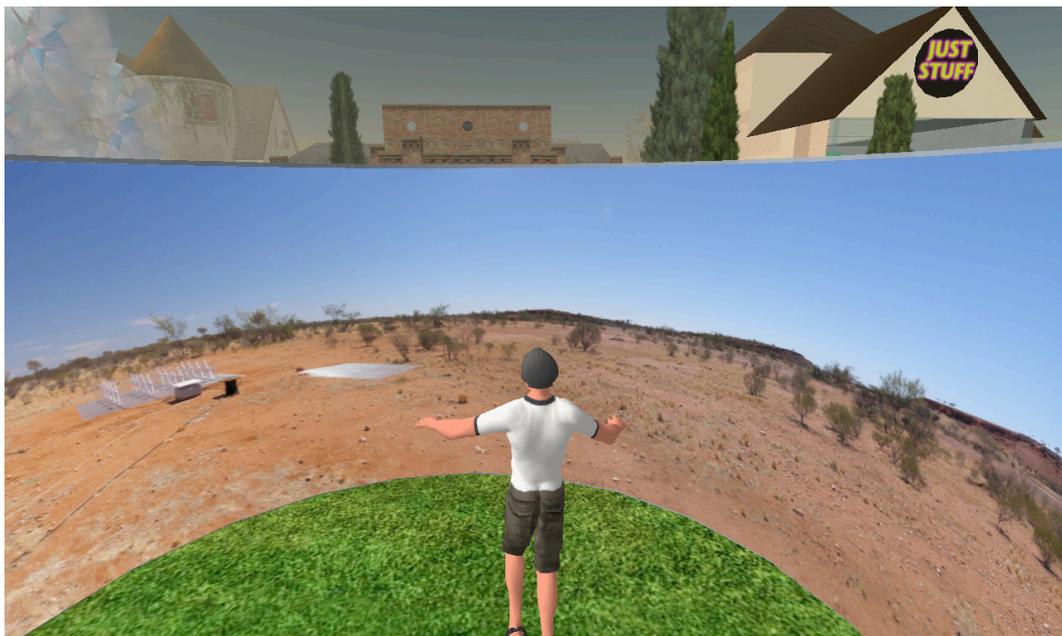


Figure 5. Virtual environment, onsite at the Wide Field Array telescope site in Western Australia.

Interactions with the data in Second Life, besides exploration in 3D, is limited in time fidelity due to the communication process at each time step with the remote server for any geometry that changes. In addition the ability to map movies as textures onto geometry is quite limited. As such it is not expected that Second Life would find many applications for time varying visualisation.

Geometric data, while usually owned by a single person can be shared by setting appropriate permissions or by giving copies to another participant. This is particularly useful since it allows multiple participants to rotate the objects to present various views, an alternative that can have benefits in a presentation mode compared to an individual walking/flying around or through the visualisation.

Conclusion

While Second Life provides an interesting environment that satisfies many of the requirements for collaborative visualisation in the sciences, there are clear limitations. Of course it should be pointed out that in the discussion presented, an attempt is being made to use software for a task it was not necessarily intended to perform. The strengths revolve around the collaborative aspects that are largely unavailable in any of the existing visualisation tools. The limitations are predominantly based upon the low geometric complexity supported and the lack of a surface mesh primitive. If the scientific visualisation task fits within these limitations then Second Life as collaborative experience is perhaps unsurpassed, it is cross platform and elegantly handles multiple participants interacting with data representations within a 3D virtual environment.

Future work will involve identifying particular collaborative science projects and produce scripting tools to enable the researchers to evaluate this media for themselves.

References

- [1] L.W. Pettersson, Collaborative Visualization: Designing and evaluating systems for co-located work. PhD thesis, Uppsala: Uppsala University, Interfaculty Units, Acta Universitatis Upsaliensis. 2008.
- [2] K.W. Brodlie, D.A. Duce, J.R. Gallop, J.P.R.B. Walton and J.D. Wood, Distributed and Collaborative Visualization, Computer Graphics Forum, Volume 23, Number 2, pp 223-251, 2004.
- [3] J. Wood, H. Wright and K. Brodlie. CSCV Computer Supported Collaborative Visualization. Proceedings of BCS Displays Group International Conference on Visualization and Modelling, 1995.
- [4] A. Herwig, P. Paar, Game Engines: Tools for Landscape Visualization and Planning?, Trends in GIS and Virtualization in Environmental Planning and Design, Wichmann Verlag, Heidelberg, 2002, pp. 161--172.
- [5] D.M. Antonacci, N. Modares, Second Life: The Educational Possibilities of a Massively Multiplayer Virtual World (MMVW). SouthWest Regional Conferences (2006).
- [6] Aimee Weber Studio and the National Oceanic and Atmospheric Administration. 3D weather visualization, 2006.
- [7] P. Kettunen, P.D. Bourke, H. Hashimoto, T. Ameniya, S.C. Müller, T. Yamaguchi, Computational study of helix wave formation in active media. Mathematical and Computer Modelling, Volume 41, pp 1013-1020, 2005.
- [8] D. Jayatilaka, S.K. Wolff, D.J. Grimwood, J.J. McKinnon, M.A. Spackman, CrystalExplorer: a tool for displaying Hirshfeld surfaces and visualising intermolecular interactions in molecular crystals. *Acta Cryst.* (2006).