

Open Source Software for Daylighting Analysis of Architectural 3D Models

Terrance Mc Minn^a

^a Curtin University of Technology School of Built Environment, Perth, Australia
Email: t.mcminn@curtin.edu.au

Abstract This paper examines the viability of using open source software for the architectural analysis of solar access and over shading of building projects. For this paper open source software also includes freely available closed source software. The Computer Aided Design software – *Google SketchUp* (Free) while not open source, is included as it is freely available, though with restricted import and export abilities.

A range of software tools are used to provide an effective procedure to aid the Architect in understanding the scope of sun penetration and overshadowing on a site and within a project. The technique can be also used lighting analysis of both external (to the building) as well as for internal spaces.

An architectural model built in *SketchUp* (free) CAD software is exported in two different forms for the Radiance Lighting Simulation Suite to provide the lighting analysis. The different exports formats allow the 3D CAD model to be accessed directly via *Radiance* for full lighting analysis or via the *Blender* Animation program for a graphical user interface limited option analysis. The *Blender Modelling Environment for Architecture* (BlendME) add-on exports the model and runs *Radiance* in the background.

Keywords *Lighting Simulation, Open Source Software*

INTRODUCTION

The use of daylight in buildings has the potential for reduction in the energy demands and increasing thermal comfort and well being of the buildings occupants (Cutler, Sheng, Martin, Glaser, et al., 2008), (Webb, 2006), (Mc Minn & Karol, 2010), (Yancey, n.d.) and others. For this reason designers need to be aware of the implications of design decisions on the performance of the building at different times of the year.

Over shadowing is a major issue when considering rooftop-harvesting systems for solar energy and solar heating. This is a relatively easy task to observe via the typical architectural CAD systems in use today. However solar ingress to the building is a far more difficult proposal for the same CAD software, particularly where the view format required might be via floor plans over the whole building (Mc Minn, 2011).

Traditional methods of visualising the potential of over shadowing in architectural models use scale models and testing with a solarscope or heliodon equipment (Mc Minn, 2004) and for daylight factor analysis a mirror sky apparatus is to simulate a overcast sky condition (Cutler et al. 2008)(Mc Minn, 2004). Scale models themselves have been shown to over predict illuminance levels (Mardaljevic, 2001). The daylight factor procedure is based only on the overcast sky and does not predict the levels achieved under non-overcast skies and for varying solar positions (Li, 2007). To model sky conditions the International Commission on Illumination (CIE) describes 15 different sky conditions to simulate skies found around the world.

There are a number of approaches to analysing solar access of architectural models. The analysis outcomes dictate the software stack necessary to achieve these outcomes. For simple over shadowing analysis on the exterior of a building project, most CAD software is more than capable of the simple presentation of the shadows on perspective images. For some software presenting the same shadows on plans, elevations and sections, all best described as orthogonal projections or views, becomes more difficult, as these views are typically used for design and documentation purposes rather than image previews.

McMinn (Mc Minn 2011) and (Mc Minn & Karol 2010) have reported on the difficulties software such as *AutoCAD*, *ArchiCAD* and *SketchUp* have when showing shadows/sun penetration internal to the building in other than perspective views. For this paper, the use of a software stack is considered to be an effective methodology to achieve the desired outcomes and can be readily extended to cover other forms of analysis such as daylight factor for non overcast skies, glare analysis, colour analysis etc.

This paper outlines a methodology where open source software can be combined, thus creating a software stack, to provide the required level of information in a cost efficient and timely manner. The software stack used in this methodology includes a free for use CAD package, open source lighting simulation software and a variety of other packages. For the purposes of this paper, *Goggle SketchUp* (Free) although closed source is included as it is freely available for use. The paper will document a procedure applied to a multi-storey building and scripting to produce a workflow.

1. DISCUSSION

The key to the software selection used in this stack is lighting simulation software. There are three major selection criteria:

- Linkage to the cad model in the form of import/export options;
- Validity of predictions; and
- The capacity to be driven via the command line.

These are key elements in creating the stack. For this purpose the Radiance Lighting Simulation Suite (referred to as *Radiance* in this paper and available via www.radiance-online.org) has been chosen.

1.1. Linkage to CAD models

Depending on the nature of the stack, some use of Graphical User Interface (GUI) software is included. For example for the model creations, CAD programs such as *SketchUp*, *AutoCAD*, *ArchiCAD*, etc. would normally be utilised by the Architectural profession to create the geometric model.

There are many closed/proprietary file formats associated with Computer Aided Design software. For our purposes it is essential that the CAD model be able to be exported via a format, which can be interpreted by our chosen lighting software. For the purposes of this paper, standard formats such as DXF (Autodesk), 3DS (Autodesk), OBJ (Alias|Wavefront Advanced Visualizer) can all be imported into *radiance*. However the DXF format has no material information only layer names so is of limited use. Depending on the route to radiance other export options maybe available, for example:

- Blender: additional export file types may be available such as COLLADA (available from *SketchUp*) and STL. The *BlendMe* plug-in exports directly into Radiance. Limited testing (Blender 2.57b) with the large model case study highlighted that there are still issues in importing significant sizes collada files.
- Direct: there is a plug-in, *su2rad* developed by Thomas Bleicher (<http://code.google.com/p/su2rad>) for *SketchUp*, which will export directly to *Radiance*.

1.2. Validity of Predictions

The Radiance Lighting Simulation Suite has been validated for daylighting and artificial lighting (see Mardaljevic (1995), (2001) and (2000)) predictions and has a number of import options or CAD export options. Reinhart and Andersen (2006, p. 892)) review of the research into validating Radiance concluded that the research presents “*a compelling case, that Radiance is capable of modeling interior illuminances due to daylight for a wide range of sky conditions and complex facade geometries*”.

It is reported by Reinhart and Fitz (2006) in a 2003-4 online survey of 27 countries, 185 designers, engineers and researchers, that 50% of the respondents used tools that either embedded or directly used the Radiance Simulation Suite. That number may have increased since Autodesk’s purchase of the *Ecotect* software and incorporated into its Architectural Suite. *Ecotect* provides a selected view of radiance (Cutler, Sheng, Martin, Glaser & Andersen, 2008). In a similar way, the *BlendMe* plug-in for *Blender* by Mark Pitman (<http://www.ods-engineering.com/blendme>) does the same for Radiance, Energy Plus and Open Foam as a black box analysis engines.

1.3. Capacity to be driven from the Command Line

The final criterion in the list is the requirement for command line access in some other form of non-GUI interface. The radiance suite is often criticised for the lack of a GUI interface (there is a limited version called *trad*, part of the Unix/OSX distribution). Other software such as *Blender* and *Ecotect* provide their own graphical interfaces.

Radiance is a suite of over 50 tools, which can be accessed directly from the command line (Larson & Shakespeare, 1998, p. 3). The radiance process itself uses pipes to interlink tools together to perform complex tasks. For example the main program, *rad*, incorporates features of the typical *make* utilities in selectively compiling objects, rendering and filtering the output images (Larson & Shakespeare, 1998, p. 18). *Trad* mentioned above works on top of the *rad*, providing a convenient method of creating the project files, controlling options and converting image formats.

1.4. Other Software in the Stack

Depending on the desired outcomes a variety of software may be required. For this paper, the outcomes required will be to produce a CD-ROM image containing an html-based presentation of the results/images of the daylighting of the case study building.

To produce the desired results will require:

- Executive summary: Animated thumbnails for different times of the year (hourly results for summer, winter and the solar equinox)
- Multiple web pages hyperlinked
- Overlay labelling on images
- Overlay background images of walls on plans

To automate the process a *BASH* script was developed to increase operational speed and execute the procedure. Other methods could easily be used such as *python*, though in general that knowledge is less

likely to be available to the majority of users even though it could be more elegant, durable and reusable than typical *bash* scripting.

Command line imaging software such as *Imagemagick* is invaluable in many of the operations detailed above such as overlays, adding text, creating thumbnails and animated images. Creating template files and processing with *nawk* for text replacement within text files can handle the creation of web pages. There is still a need for GUI programs such as *gimp* or *PhotoShop* to remove details from overlay masters and to determine offset values to overlaying. Equally, for those unfamiliar with web page construction, software such as *Dreamweaver* can speed the development of the web page templates.

1.5. The Process

The process flowchart is shown in Figure 1. As can be seen, the use of *radiance* via the command line interface allows simple scripting to be used to link the applications into an effective process. In this way *radiance* can be used as a ‘black box’ program to create the required base images in the software stack.

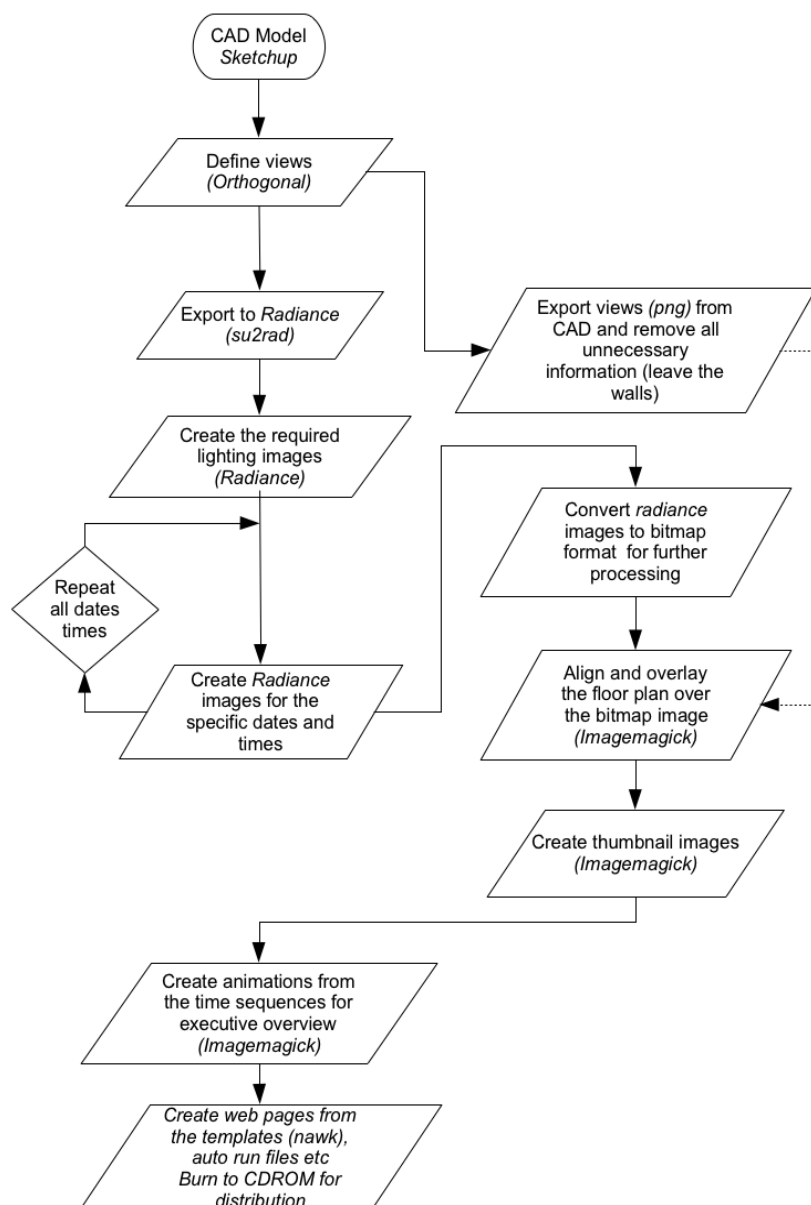


Figure 1. Flow Chart

2. CASE STUDY

The architects of the case study project (Figure 2) provided a large *SketchUp* model for analysis. This model was imported into *Sketchup*. A number of layers were required to be hidden including pure 2D objects such as the vegetation and some 3D objects such as furniture and other buildings. This leaves the project's geometry and materials visible.

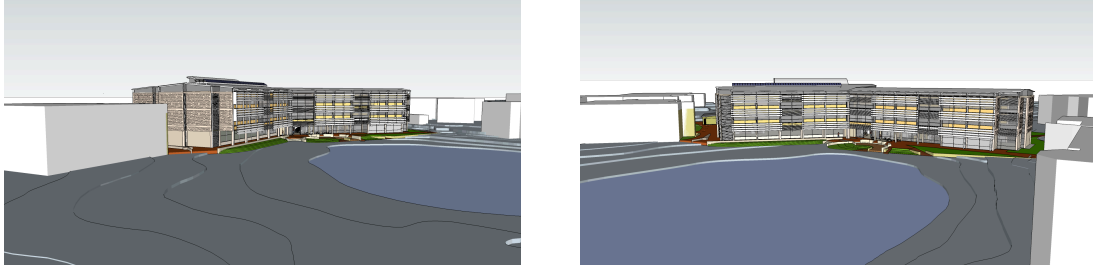


Figure 2. Edith Cowan University (Joondalup Campus) – Building 23, Computing Engineering and Technology, Architects: Ferguson Architects

The next step was to ensure the CAD model specified the correct orientation, and geographical location (neither of which are essential during creation of the CAD model but for this application are essential). A detailed check of the materials used to specify the glazing used in the building was also carried out.

For the analysis, orthographical top view (plans) were created for each floor level and saved as a named view within the *SketchUp* model. By definition, within *SketchUp*, a top view is always from above the object (maximum Z). This is not what is required for this project's analysis, but a good starting point. The analysis requires images for each floor level.

SketchUp does not provide the tools to adjust/specify the height (z) of its orthographical top views, however it is possible to manipulate the height through the *Ruby* programmable interface. This is what is done by *su2rad* plug-in. The various view heights (z) were manipulated via this plug-in.

The *su2rad* plug-in was then used to export the model directly to *radiance*. The export process also mapped the materials, textures and provided a mapping between the existing glazing materials and a more accurate *radiance* glass material definition. It is important to realise that once the export to *radiance* has been completed, it is possible that the rest of the process can be carried out on any operating system that supports the software stack. In this case, the stack can run directly on *Linux* or *Macintosh OSX*. The full version of *Radiance* does not run under *Microsoft windows*.

A preliminary run using *trad* allowed the project to be tested within *radiance* and sample output image created for testing the overlay coordinates.

From this test run the various templates are created and the script tested.

2.1. Hardware

As McMinn (2010) has previously documented, the hardware requirements of the stack are modest. The most time consuming aspect is the *radiance* renderings. Whilst the renders can be carried out on a netbook computer (EeePC 701, 900kHz Intel Celeron processor with 1GB of RAM, USB hard drive, running on a custom Kubuntu 8.10 live CDROM via an SDHC card), the typical desktop computer is adequate for most projects. Where there are significantly complex models, then the *radiance* software has the ability to share the image creation over multiple processors and multiple networked computers (without the need to establish rendering farms). It could be four or five networked computers in the office running at night from a live *Linux* CD!

2.2. Discussion

The major costing in this model is related to execution time given that the software is either open source or free for use and in the case of PC's the operating system – *Linux* is also open source. This can be installed on the system or run via *vmplayer* from VMware.com or *virtualbox* from virtualbox.org. Both *virtualbox* and *vmplayer* are free for use.

From an office point of view having a easy process of exporting the CAD geometry to radiance with as little re-working of the model is crucial to the willingness to use this method of evaluation. The *SketchUp* and *su2rad* plug-in provides the low cost, in terms of time, method of exporting to *radiance*. Once in *radiance* the model can be fine tuned for execution time utilising instances to reduce the size and compilation times of the model. Compilation time is a major issue as every change in time will require a new sky definition and hence a recompilation of the model. Mc Minn & Karol (2010, p. 380) show on a small low powered netbook that the compilation using instances can reduce from over 9 minutes to less than 1 second after the initial compilation of the geometry.



Figure 3. Sample Image from Radiance

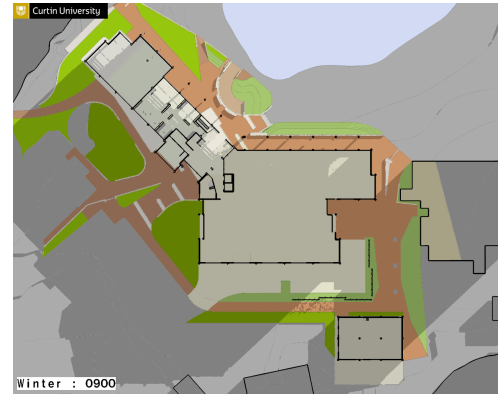


Figure 4. Sample with overlay of walls

The use of scripting techniques, image overlays (see Figures 3 and 4), thumbnails and animation techniques provide a time effect, visually rich and consistent production system. Compilation into an electronic presentation format using standard web browsers and simple html designs provides a low resource cost, simple duplication and rapid visual analysis format well appreciated by the clients – e.g. property clients, architects, engineers, building consultants and council officers. Viewing large number of images with only small variations in parts of the image is not a task many people would find enjoyable. If the number of images are in the hundreds then it is a task which will not be carried out, or if done, prone to error. Animations and executive summaries allow rapid and accurate visual analysis, in the order of minutes. Hyperlinked images provide finding the relative high-resolution details simple.

3. CONCLUSIONS

The use of a software stack of open source software provides the ability to compliment the existing capabilities of commercial software in use by typical architectural offices. Currently the major CAD packages are unable to provide the required analysis in terms of orthogonal top views ad specified (z) heights whilst showing the degree of solar penetration into the project. The software stack by utilising selected software provides significant information to the designer. It can easily be extended, in the instance of using *radiance* to provide glare analysis, daylight factor analysis (using clear sky with sun conditions), illumination levels (total analysis of the combined daylight and artificial lighting), etc.

The presentation technique of using animated images as the executive summary and hyper linked web pages provides rapid analysis of the results and cheap reproduction of the ‘report’ for use within the design team and consultants. With better information on solar loads in the building the designers and engineers are more able to tailor the building to better respond to the climatic conditions.

REFERENCES

- Cutler, Barbara, Sheng, Yu, Martin, Steve, Glaser, Daniel and Andersen, Marilyne (2008) ‘Interactive selection of optimal fenestration materials for schematic architectural daylighting design’, *Automation in Construction*, 17(7), 809-823.
- Larson, Greg Ward and Shakespeare, Rob (1998) *Rendering with Radiance: The Art and Science of Lighting Visualization*, San Francisco, California, Morgan Kaufmann.
- Li, Danny H.W. (2007) ‘Daylight and energy implications for CIE standard skies’, *Energy Conversion and Management*, 48(3), 745-755.

- Mardaljevic, J. (2000) 'Simulation of annual daylighting profiles for internal illuminance', *Lighting Research and Technology*, 32(3), 111-118.
- Mardaljevic, J. (2001) 'The BRE-IDMP dataset: A new benchmark for the validation of illuminance prediction techniques', *Lighting Research and Technology*, 33(2), 117-136.
- Mardaljevic, J. (1995) 'Validation of a lighting simulation program under real sky conditions', *Lighting research & technology*, 27(4), 181-188.
- Mc Minn, T (2004) 'Implementing lighting visualisation using computer simulations', In *38th Annual Conference of the Architectural Science Association ANZAScA and International Building Performance Simulation Association - Australiasia Conference*, Launceston, Tasmania: School of Architecture, University of Tasmania., ANZAScA.
- Mc Minn, T (2011) 'Radiance as a tool for investigating solar penetration in complex buildings', *Architectural Science Review*, 54(2), 101-107.
- Mc Minn, T and Karol, E (2010) 'Extending the use of existing CAD software to enable visualisation of Solar Penetration in Complex Building forms', *Architectural Science Review*, 53(4), 374-383.
- Reinhart, C. and Fitz, A. (2006) 'Findings from a survey on the current use of daylight simulations in building design', *Energy and Buildings*, 38(7), 824-835.
- Reinhart, Christoph F. and Andersen, Marilynne (2006) 'Development and validation of a Radiance model for a translucent panel', *Energy and Buildings*, 38(7), 890-904.
- Webb, Ann R. (2006) 'Considerations for lighting in the built environment: Non-visual effects of light', *Energy and Buildings*, 38(7), 721-727.
- Yancey, Keith J (n.d.) 'The daylighting design process – tools for predicting daylight and techniques for making sound decisions about daylight during design', [online] Available from: http://best2.thebestconference.org/pdfs/023_EE7-1.pdf (Accessed 9 July 2011).