

## **Radiance as a tool for investigating solar penetration in complex buildings**

Terrance Mc Minn (Curtin University of Technology, School of Built Environment,  
Perth, Australia)

Corresponding Author: Tel: +61 89266 7175; Fax: +61 89266 2711; Email:  
t.mcminn@curtin.edu.au

**Abstract:** There are a number of tools currently used by architects for the investigation of solar penetration in buildings. However, complex building forms present unique analysis difficulties. Current 3D computer aided design software can display sun patches and shadow on the exterior of buildings, however, it is difficult to obtain accurate and timely information about the sun patches internal to the building. This paper details a methodology to enhance the capability of commonly used CAD software to show indoor sun patches in complex buildings. The paper demonstrates how multiple software packages, including the Radiance Lighting Simulation Suite, can be used as part of a tool chain to display solar penetration in complex buildings in a timely manner.

**Keywords:** Lighting Visualisation, Software, Solar Penetration

## **Introduction**

Energy efficient design must deal with both heating and cooling seasons and consider the nature of the solar penetration into buildings. In the heating seasons, the use of the building's fabric and internal mass as a 'solar storehouse' requires accurate knowledge of the magnitude and heating potential of the internal areas of the building exposed to the sun, as well as the time of day for the occurrences. In the cooling seasons, knowledge of solar penetration into the building allows better controlling of the shading of openings and glazing areas.

Investigation of the thermal performance of complex buildings requires studies to be performed at various times of the day and seasons. For this, viewing the solar patches over complete floor areas often is the most efficient for the designers. For the purpose of this paper, solar patches refer to the portion/patch of a building element (wall, floor, etc.) that is exposed to the direct solar beam, hence its appearance as a bright, or sun lit patch of material.

There are a variety of tools and techniques available to facilitate this study from solar tables or charts (Phillips, 1987), physical models and solarscopes. Each technique has inherent advantages and disadvantages. In many instances practitioners may not be fully aware of the limitations of individual techniques and processes. For complex building forms computer models are usually the preferred modelling tool. Many architects use computer aided design (CAD) software and prefer solutions which can interact with the building models, due to the cost and time required to re-implement the current design for other modelling methods. There can be considerable costs in time, labour and disruption to the design process to validate design decisions via other techniques. It should be recognised that while 3D CAD

software can show sun shadows, they are rarely the most appropriate design tool for solar penetration issues. Charts such as those in 'Sunshine and Shade' (Phillips, 1987) and the like, can provide the critical design information in a format, which can be more readily integrated in to the design process. The CAD software can then be better used as a checking/validation tool. CAD software often has difficulty in presenting sun patches (or solar penetration) information on internal surfaces in other than perspective views. More often one orthogonal plan view of the sun patches over the full buildings floor plan can be far more meaningful to the designer than 20 perspectives views of individual rooms.

### **Method**

To enhance typical CAD software used by many Architects and designers, as the primary building model, requires the use of external analysis software. Multiple software packages used together as a tool chain to produce the desired diagrams are proposed. The *Radiance Lighting Simulation Suite* software (referred to as *Radiance* in this paper), as proposed in this paper as the simulation tool, has the capacity to create plan views of the solar penetration into complex building forms and the creation of parallel (and other) views at any XYZ location in the model and at any orientation. This provides great flexibility in aligning the view precisely to the requirements of the project. There are different methods to create the plan images in this software that are most desirable for viewing indoor solar patches:

- Parallel plan views at arbitrary heights, for example setting the viewing plane height at 1.2 metres above the chosen floor height; or
- Parallel plan views and the use of clipping planes between the eye point and the object to cut away part of the building to reveal otherwise hidden details,

whilst retaining the geometry in the lighting calculations. This method can also be used in conjunction with the above method.

*Radiance* utilises a physically based rendering engine (Bauer, 2002) used worldwide for lighting analysis for both artificial and daylighting studies. It has been incorporated into other packages, such as *IES Architectural Suite*, *Blender* and Autodesk's *Ecotect*. The simulation engine has undergone a number of independent international validations (Ibarra & Reinhart, 2009:196) and is well accepted. The software is research level open source software and has a steep learning curve compared to modern commercial programs, such as CAD, which tends to provide a one size fits all solution.

The commercial software approach minimises the learning curve by reducing the options available to customise an approach. These software applications are often monolith programs, which rarely allow the user the ability to define their own approach in evaluating a solution in new or different ways to those provided.

*Radiance*, however, encourages the customisation of the approach. This in turn requires a deeper knowledge and experience of the software, as the *Radiance* suite comprises some 50 plus separate programs that may be used in isolation or as part of a chain of processes. Typically this is a difference between monolith programs and research software. Many people cite the fact that as *Radiance* does not have a highly polished graphical user interface, and there is little likelihood of one being developed, as being the major reason not to use the software. However the user community is extremely helpful and there is a large repository of discussions available on the <http://www.radiance-online.org> website.

## **Hardware**

Typically the hardware requirements are not excessive, as the recent computers used in Architectural practices would have sufficient capacity. The *Radiance* software has been developed to execute on the *Unix* operating system. (The Apple Macintosh OSX operating system is sufficiently similar to *Unix* that the software has been compiled for it and used by the prime developer of the software.) There are no current ports of the full *Radiance* software to run natively under Microsoft operating systems. A limited version has been compiled using *Mingw* (Minimalist GNU for Windows – available from <http://www.mingw.org>) or using *Cygwin* (available from <http://www.cygwin.com>).

The approach described in this paper uses a *Linux* system based on *Kubuntu* which runs the full version of *Radiance*, though most *Linux* distributions should be able to compile the source code (see <http://www.radiance-online.org>) if a pre-compiled package is not available. A live CD-ROM containing the *Linux Kubuntu* operating system and the radiance software is available from <http://www.luminance.londonmet.ac.uk/curtin>. It was developed at Curtin University and is used in the teaching program for architects and interior architects, to allow *Linux* to run without installation of any software on the host PC. Computers with adequate memory capacity (over 2GB) can use the free virtualisation software *vmplayer* (available from <http://www.vmware.com>) to enable running *Linux* inside Microsoft *Windows*.

## **Process**

In simplistic terms, creating a tool chain of applications capable of producing the

required information for the analysis of a complex building's solar penetration can be summarised as:

- Development of the CAD model
- Export to Radiance
- Create appropriate views/scenes (i.e. each floor height and location under consideration)
- Create images of scenes over all relevant dates and times, typically the hourly over the complete sunlit times of the summer and winter solstice and the equinox.
- Report generation

The cost in terms of time and customisation of exporting the CAD model into *Radiance* can be significant depending on the availability of appropriate exporters/importers. *Radiance* has a limited range of conversion tools available. They include *3DS2MGF/MGF2RAD* – 3D Studio models (two steps required), *ARCH2RAD* – Architrion text files, *DXF2RAD* (for simplified input with no material information), *OBJ2RAD* – Wavefront and *SU2RAD* – a plug-in for *SketchUp* to export directly to *Radiance*.

In evaluating the performance of an export/import tool, consideration of the following aspects of the conversion are highly important:

- The exporter needs to be able to convert 3D objects into polygons or meshes
- Conversion of curves and polygons via triangulation, typically used in *3DS* exporting creates very large models
- Material properties conversions such as colour, transparency, reflectance, transmittance and textures.

- Removal of any “suns” from the model

Fine-tuning of the conversion may be necessary for glazing systems and any

important colouring and textures. *Radiance* does not provide any graphical

application to modify the text files for the geometry and materials definitions.

*SketchUp* (Google SketchUp – free, or the PRO version) in combination with the

*Su2rad* plug-in are very effective tools to convert 3D CAD models directly to

*Radiance*. The *Su2rad* plug-in has been developed by Thomas Bleicher

(<http://code.google.com/p/su2rad>) and supports development of custom materials

libraries and re-mapping materials to custom definitions at the time of the export.

Additionally, non-triangulation of polygons, exporting textures, the geographic

location, orientation and sun are supported. As considerable work is done by the

plug-in, very large models may take some time to export. For example, a 100MB

*SketchUp* model took approximately 20 minutes on a Apple quad core Mac Pro

computer.

Orthogonal *Radiance* views can be defined in two ways depending on the scene

complexity:

1. Parallel view at an arbitrary height, or
2. Parallel view, e.g. a top view, and then use the clipping planes setting to remove portions of the model between the camera and the viewpoint. Both fore and aft clipping planes are supported. The major difference between *Radiance's* use of clippings plans and that used by typical CAD software, such as *AutoCAD*, *SketchUp* or *ArchiCAD*, is that while the extra geometry is removed from the view by *Radiance*, it remains in tact for the shading calculations. Figures 1 to 3 illustrate difference.

The three images have a sun setting of June 21, 13:00 (Perth, Western Australia – Latitude 31.933 degrees South, Longitude: 115.833 degrees East). Figure 3, shows the podium storey dark, as there is no light entering space via the outer window walls; the length of the overall building shadows are different and only one group of columns are casting shadows in Figure 3 compared to all columns in Figure 2.

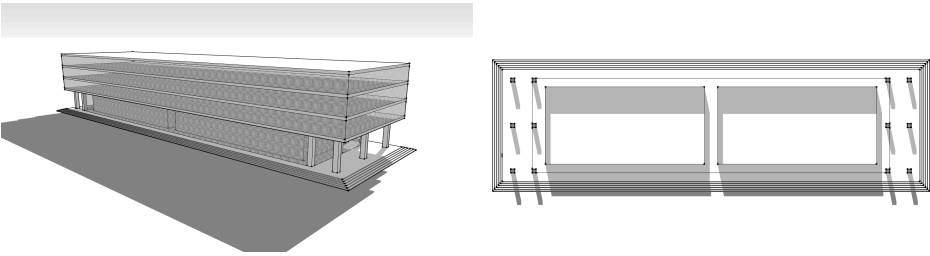


Figure 1: SketchUp Isometric      Figure 2: SketchUp Plan View

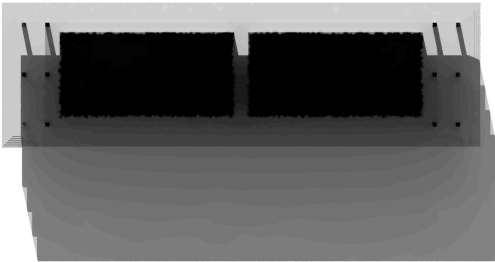


Figure 3: Radiance Plan View - Upper

Having the model imported into *Radiance* and appropriate views (scenes) defined, it is then simply a matter of rendering the scenes for every date and time required for the investigation. Typically this might require hourly simulations for at least the two extreme solar paths (summer and winter solstice) and the equinox or mid season path. On average over the day light hours, this might equate to 10 images per sun



path per scene or 30 images per scene. For complex multi storey buildings, evaluation of multiple views (for example at different floor levels) will result in significant number of images.

Visual analyse of large numbers of images becomes difficult, especially when only small sections of each image changes. Typically, to aid this visual analysis, the images will be enhanced by adding line work for walls and text data such as the season and time of day.

A presentation technique was created using a visually efficient method of presenting the images in a format that allowed rapid assessment and quick selection of relevant images. The technique required the production of an executive summary by way of animated time-lapse images for each season/solar path. The summary page then linked to the season view of hourly thumbnail images, which linked to individual larger images. The complete presentation was designed to be presented via a web browser interface and can be cheaply reproduced on a CDROM.

The web based presentation technique was used, as it provided a cost effective, user friendly and highly efficient method to analyse large numbers of images. Two projects, for different clients and architects, have been completed using this technique. The feedback from both consultants and architects has been very encouraging, even though the solar penetration into the projects was more than expected by the architects.

A basic flowchart of the approach used is reproduced in Figure 4 below. See McMinn & Karol (2010) for a more detailed flowchart and discussion.

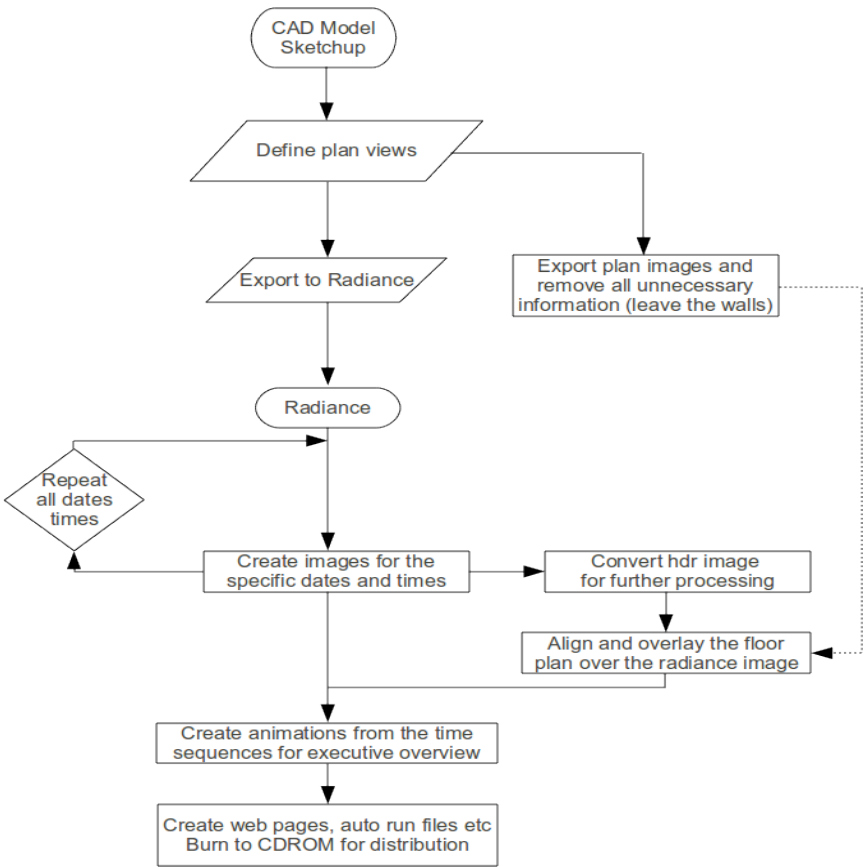


Figure 4: Flowchart

**Software Tools**

The software used in the flowchart (Figure 4 and Table 1) is freely available and can be readily downloaded from the Internet. Table 1 details the various packages used in the tool chain, though other software could be substituted. Most packages are available cross platform, with exception of *Radiance* as described above.

Table 1: Software

Component	Software	Web Reference
CAD system	<i>SketchUp</i>	<a href="http://sketchup.google.com">http://sketchup.google.com</a>
CAD Export plug-in	<i>Su2rad</i>	<a href="http://code.google.com/p/su2rad">http://code.google.com/p/su2rad</a>
Lighting Simulations	<i>Radiance</i>	<a href="http://www.radiance-online.org">http://www.radiance-online.org</a>
Image manipulation – for overlay preparation	<i>Gimp</i>	<a href="http://www.gimp.org">http://www.gimp.org</a>
Command line image manipulation	<i>Imagemagick</i>	<a href="http://www.imagemagick.org">http://www.imagemagick.org</a>
Shell and scripting plus Web page creation	<i>Bash</i> shell	<a href="http://www.gnu.org/software/bash">http://www.gnu.org/software/bash</a>
File based text replacement	<i>Nawk</i>	<a href="http://www.cs.bell-labs.com/who/bwk/index.html">http://www.cs.bell-labs.com/who/bwk/index.html</a>

Assembling a set of tools which can work together to achieve the desired outcomes of producing a set of orthogonal plan views, or indeed any other view types supported by *Radiance*, to depict the sun patches at various points or levels within a building complex is problematic. It requires extensive knowledge of a wide range of applications, often unrelated to the discipline. The software listed in Table 1 is a typical set of free for use software, though there are many other approaches that could be used depending on the skills of the user. This set of applications doesn't require formal programming and in the most part can be implemented with little reliance on anything beyond what are included in the manual pages for each software package.

Figure 4 shows the combination of software tools and a flowchart of the process used in this paper. Table 1 also includes duplicate applications in some areas, such as image manipulation. A combination of applications such as *Gimp* or *PhotoShop*, are used depending on the position in the process. Graphical user interface style programs are used in the area of the one off master creations phase. For example, *Gimp* is used to manipulate, delete background elements from the floor plan images saved from *SketchUp*, where as *ImageMagic* is used in the repetitive phases to composite images, create thumbnails views and create animations.

The choices of applications are narrowed by input and output compatibility and the mode of operation – graphical user interface only or command line accessible. For example when many images are required to be created, some form of automation is essential. An example is the creation of the animations. There are a number of commercial tools available, though in this instance *ImageMagic* can be used. Apart from its extensive repertoire of capabilities it is also a command line set of programs rather than having a graphical user interface, thus suitable for inclusion in an automated scripted application.

To achieve the required level of automation using multiple applications requires some form of scripting. To this end, the standard command shell used in *Linux* or *OSX* is the *Bash* shell, though other command shells could be used. In many respects Microsoft's *cmd* performs a similar function. In general the command shells have many more capabilities than simply executing individual applications. Most shells also provide methods where groups of commands can be assembled into a file (batch or shell script) and then executed as if typed directly on the keyboard. These

script concepts are all that is needed in this application to tie together the various applications into a tool chain. The script can become very sophisticated depending on the scripting capabilities of your chosen command shell.

In this application, the scripting capabilities of the *Bash* shell allow for program flow characteristics such as looping, file checking, variable testing, variable replacement etc. This in turn allows other tools to be employed for example text replacement in files using the *Nawk* software, in combination with template files. The *Bash* script substitutes variable data inside the text files – an advanced search and replace facility and looping to create a custom work flow.

The *Radiance* software predominately uses plain text files for its primary input (though it does use binary files internally). Scripting is used extensively in the *Radiance* tool set. Many of the tools themselves are scripts. In this application, using template files and text replacement, allows the *Radiance* software to be used as a black box simulation system. Other tools are then used to take the created images further by compositing, creating thumbnail images and animations.

### ***Discussion***

The methods proposed in this paper were developed to meet a need to produce images to demonstrate the degree of solar penetration into a proposed shopping and office complex (Figures 5 and 6) and later to determine the solar penetration, if any, for a proposed university building (Figures 7 and 8).

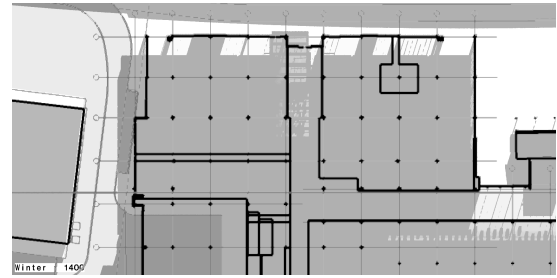
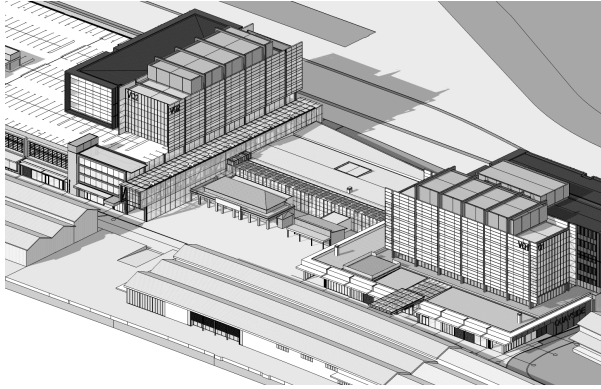
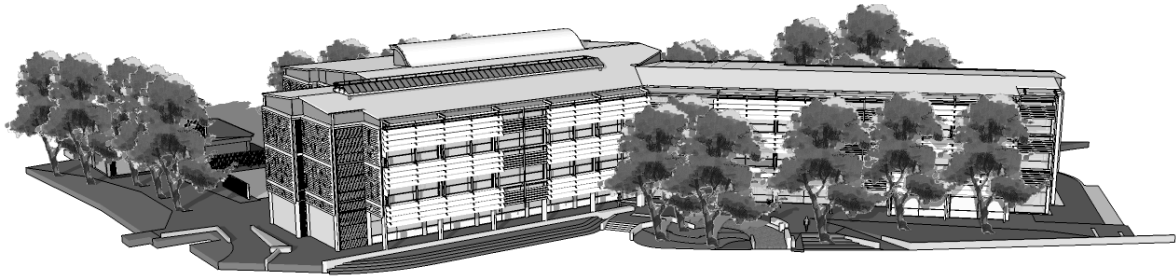


Figure 5: *Project A – Proposed Shopping*     Figure 6: *Project A - Shopping plaza level*

The CAD models provided by the architects were in *SketchUp* format. Initial analysis revolved around attempting to work with *SketchUp* to show the sun penetration inside the complex at different locations. The requirement was to produce orthogonal plan views for ease of analysis. Different methods were investigated to locate the plan view at the required vertical height, including the use of the programmable (*Ruby*) interface, but were unable to achieve the desired outcomes. Additionally, Project A was exported to other common CAD software in use in Western Australia. Similar difficulties were experienced with *AutoCAD*, *ArchiCad* and *Artlantis* rendering software. In these software suites, the use of cutting planes uniformly hid the cut geometry from the shading algorithm (see Figure 2).



*Figure 7: Project B: Edith Cowan University (Joondalup Campus) - Building 23, Computing Engineering and Technology, Architects; Ferguson Architects*



*Figure 8: Project B: Ground Floor*

Having determined that the available CAD/rendering software was unsuitable for the task, a tool chain of applications was devised based around the *Radiance* software. In this respect, *Radiance* was used as a virtual 'black box' simulation engine to produce the desired images.

The production of the images could be automated by scripting elements from the *Radiance* input files. To achieve the goal of automation, the project file and sky files were used as templates and a simple script developed to replace key information

based on the date and hour for the simulation. Overlaying the CAD geometry along with a date and time stamp on each image enhanced visual inspection of the simulation images. Again this was automated in the script by using *Radiance's 'psign'* and *ImageMagick's 'convert'* programs.

With the large number of images produced, it was apparent that traditional paper based reporting would be costly and inefficient, in terms of rapid assimilation of the information. Therefore, the presentation report became a web-based display of images using a combination of animations (executive summary), linked to daily thumbnail images (hourly), and linked to large format hourly images.

The use of scripting techniques reduced the time required to individually run each of the designated time/date combinations, overlay and time stamp every image, create thumbnails and web pages. The script provides automation of the process and brings advantages of speed, repeatability and accuracy.

## **Conclusions**

Current trends in design require designers to be able to identify the scope and locations of solar ingress to buildings for a variety of reasons – from the need to minimise the direct solar beam on the working surface, to the thermal design criteria of heating specific building elements (heat stores) and minimising solar loads for air conditioning. To achieve the desired outcomes of producing a set of orthogonal plan views, depicting the sun patches at various points or levels within a building complex using the currently available CAD software, with the standard menus and graphical interface, was not possible (at least with regards to *AutoCAD*, *ArchiCAD* and *SketchUp*). Therefore, a complex range of applications working together as a tool



chain has been assembled. In addition to the simple image creation problem, the tool chain addresses the reporting analysis issue. Typical presentation of large numbers images with small changes by printed images is costly, time consuming and daunting for the target audience.

I would expect if there is sufficient feedback to the software houses, the CAD programs could be easily 'upgraded' to provide sun penetration and shadow casting in orthogonal views at arbitrary view heights and orientations. There is very little difference in the calculation criteria of positioning the viewing plane at a user specified height versus the minimum height to include the whole building. Recent investigations of the code in the *su2rad* plug-in for *SketchUp* and the published *SketchUp Ruby API*, indicate that with clever programming, it is has been possible to add this feature to *SketchUp* since version 6, yet it has not been done.

In the interim, this paper identifies a set of programs that when used together can provide the required information in a timely and convenient manner. The individual programs provide a wealth of capacity, well beyond the simple usage presented in this paper. For example, the method can easily be expanded to produce iso-contour imagery, glare analysis, area based reporting of light levels, animated fly-through, etc. To be effective and time wise though, requires some preliminary work and the creation of work flows and scripting. Once these have been defined, subsequent generation of the images is straightforward and can be run unattended.

There is a potential for the technique of building tool chains to satisfy other existing deficiencies in the current software packages. Current examples include using CAD programs as the external modelling applications for many simulations, such as thermal analysis. The *IES Architectural Suite* uses this approach. It is an efficient use

of resources but relies on open standards to allow for the interchange of information between applications. This is a major problem with proprietary software. The success of the current application comes from the development of an accessible exporter for the CAD model into a suitable external program. Other applications could include lighting studies, glare analysis and quantitative over shadowing analysis for different seasons. Applications of web based presentation techniques in conjunction with paper reports have the potential to increase the level of understanding of building proposals, of often quite technical areas, by clients and regulators.

## References

- Bauer, C. (2002). *Designer's RADIANCE*. Paper presented at the Scientific Applications Using Radiance, University of Applied Sciences of Western Switzerland (HES-SO), Ecole d'ingénieurs et d'architectes de Fribourg, Switzerland.
- Building Code of Australia*. (2010). Retrieved from [www.abcb.gov.au](http://www.abcb.gov.au).
- Ibarra, D. I., & Reinhart, C. F. (2009). *Daylight Factor Simulations – How close do simulation beginners 'really' get?* Paper presented at the Building Simulation 2009. Retrieved from [www.ibpsa.org/proceedings/BS2009/BS09\\_0196\\_203.pdf](http://www.ibpsa.org/proceedings/BS2009/BS09_0196_203.pdf)
- Mc Minn, T., & 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.
- Phillips, R. O. (1987). *Sunshine and shade in Australasia* (5th ed.). Chatswood, N.S.W: National Building Technology Centre.

## **Software Referenced**

*Blender*, URL <http://www.blender.org>

*Ecotect*, URL <http://www.autodesk.com.au>

*IES Architectural Suite*, URL <http://www.iesve.com/A-NZ>

*Radiance*, URL <http://www.radiance-online.org>

*Kubuntu*, URL <http://www.kubuntu.org>