A Critical Evaluation of Radiance as a Tool for Calculating Radiation View Factors

Sarith Subramaniam[1], Sabine Hoffmann[1]
[1]Department of Civil Engineering, TU Kaiserslautern, Kaiserslautern, Germany

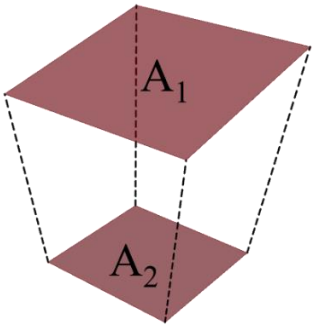
Abstract

The research underlying this paper investigates the feasibility of using Radiance, an open source raytracing software, for calculating view factors. Radiance is intended, and predominantly used, as an engine for lighting simulations. The authors outline the methodology for using Radiance solely to calculate view factors and then provide a numerical critique of its computational speed vis-à-vis standard radiosity-based approach as implemented in another open source tool called View3D. The results from the case studies presented in this paper suggest that while Radiance is unlikely to be effective as a drop-in replacement for radiosity-based software, it can be advantageously leveraged in scenarios where view factors are to be calculated for only specific portions of the overall input geometry.

Introduction

Radiation view factors are used to mathematically express the radiant energy exchange between surfaces. In scientific literature, view factors have also been referred to as angle factors, area factors, geometrical factors, shape factors and configuration factors (Holman 1986; Howell et al. 2010).

For surfaces 1 and 2 shown in Figure 1, the view factor $F_{1\text{-}2}$ can be defined as the fraction of radiant energy leaving surface 1 that reaches surface 2.

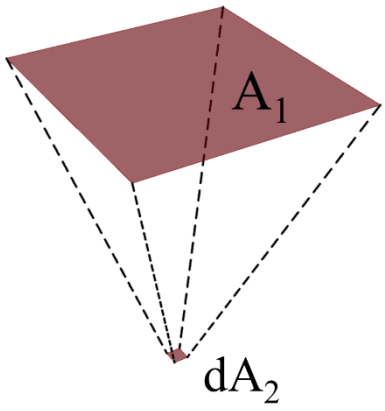


$$A_1F_{1\text{-}2} = A_2F_{2\text{-}1}$$

Figure 1. Radiative exchange between two surfaces of finite areas. A_1 and A_2 represent the areas of surfaces 1 and 2 respectively. The equation in the above image provides the reciprocity relationship between the two surfaces in terms of view factors and areas.

Similarly, the view factor $F_{2\text{-}1}$ can be defined as the fraction of radiant energy leaving surface 2 that reaches surface 1.

View factors can also be defined in the context of radiation from a large finite surface reaching the surface of a differential area. For the finite surface 1 and differential surface 2 shown in Figure 2, the view factor $F_{1\text{-}2}$ can be defined as the fraction of radiant energy leaving surface 1 that reaches the differential area 2.



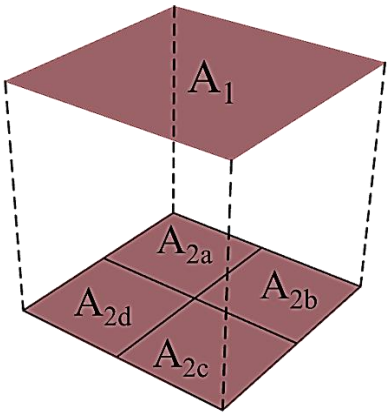
$$A_1F_{1\text{-}d2} = dA_2F_{d2\text{-}1}$$

Figure 2. Radiative exchange between a finite surface A_1 and differential surface dA_2.

As shown through the equations in Figure 1 and 2, view factors adhere to reciprocity relationships. New view factors can be derived from already known view factors through additional algebraic manipulations. One such example is shown in Figure 3, where view factors between two surfaces are expressed in terms of sub-divided surfaces.

The use of view factor calculations is commonplace in building simulations. Some of the prominent applications are listed below:

1. Building energy simulations: View factors are employed to calculate radiative heat transfer between different surfaces in thermal zones and between contextual surfaces outside thermal zones.



$$A_1F_{1\text{-}2} = A_2F_{2\text{-}1}$$

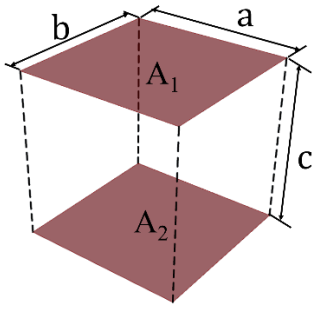
$$A_2 = A_{2a} + A_{2b} + A_{2c} + A_{2d}$$

$$A_1F_{1\text{-}2} = A_{2a}F_{2a\text{-}1} + A_{2b}F_{2b\text{-}1} + A_{2c}F_{2c\text{-}1} + A_{2d}F_{2d\text{-}1}$$

Figure 3. Surface subdivision for calculating view factors by approximating a finite surface to be composed of several smaller surfaces (Howell et al. 2010).

2. Lighting simulations: View factors are used to account for interreflection of luminous flux between surfaces in radiosity-based lighting simulation software (Ashdown 1994).
3. Thermal comfort simulations: They are used to calculate directly incident solar radiation through glazing surface on the human body (Arens et al. 2015; Hoffmann et al. 2012).
4. Photovoltaic (PV) simulations: View factors are used to calculate the amount of diffused radiation collected by PV collectors (Appelbaum and Aronescu 2016).

Analytical solutions exist for the calculation of view factors between surfaces of simple shapes such as polygons, circles and certain three-dimensional surfaces. For example, the mathematical relationship for the calculation of view factor between two identical, parallel, directly opposed rectangles is shown in Figure 4.



$$X = \frac{a}{c};\quad Y = \frac{b}{c}$$

$$F_{1\text{-}2} = \frac{2}{\pi XY}\left\{\ln\left[\frac{(1+X^2)(1+Y^2)}{1+X^2+Y^2}\right]^{1/2} + X\sqrt{1+Y^2}\tan^{-1}\frac{X}{\sqrt{1+Y^2}} + Y\sqrt{1+X^2}\tan^{-1}\frac{Y}{\sqrt{1+X^2}} - X\tan^{-1}X - Y\tan^{-1}Y\right\}$$

Figure 4. The analytical solution for calculating the view factor between two identical, parallel, directly opposed rectangles. $F_{1\text{-}2}$ is the view factor between rectangles A_1 and A_2. (Howell et al. 2010).

A comprehensive list of such analytical solutions can be found in several textbooks on heat transfer. Additionally, there are a few online tools that also facilitate the calculation of view factors (Howell et al. 2010). View factor calculations in building simulations are typically performed with software that implement radiosity-based optimization algorithms such as the Hemi-cube approach or adaptive integration for the calculation of complex geometries (Cohen and Greenberg 1985; Walton 2002).

Whole building energy simulation tools such as EnergyPlus implement view factor calculation algorithms within their workflows. View3D, a standalone program developed by George Walton at NIST in the 1980s, is one of the few freely available tools that is specifically intended for the calculation of view factors (Walton 2002). View3D is distributed along with EnergyPlus and has been recommended by its developers as the preferred tool for calculating view factors for geometries commonly encountered in building simulations.

Motivation for this research

Radiosity-based software are usually not suited for complex geometries. The standard approach towards convergence in such software relies on solving the energy balance between all the surfaces present in the input geometry. So, for an input geometry with N surfaces, the solution for view factors will involve a matrix of (N x N) values.

Figure 5 shows a typical test with View3D where the calculation runtimes were plotted as a function of the number of surfaces in the input geometry. The geometry used for this test, which were building models with progressively higher surfaces and mesh subdivision, required the calculation of view factors for obstructed and unobstructed surfaces. As the plot indicates, increasing the number of surfaces in the input geometry leads to progressively higher runtimes that do not scale linearly.

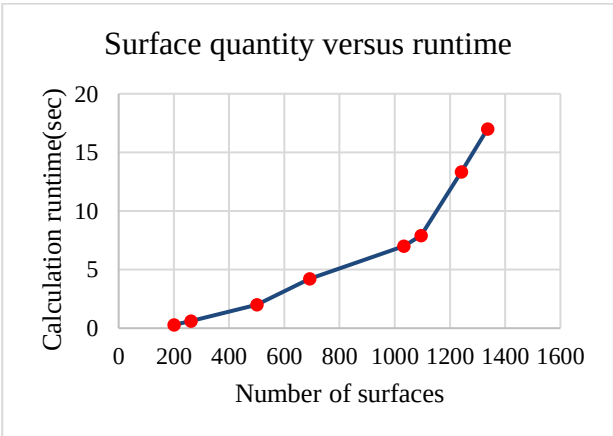


Figure 5.Runtime, in seconds, for calculating view factors with View3D on a single dedicated processor of an Intel Core i7-8700 3.20 GHz machine.

The non-linear increase in the calculation times seen in Figure 5 can be attributed to the fact that View3D employs a combination of Line Integration, Area Integration and Adaptive Integration to calculate view factors based on the input geometry.

The drawback relating to simulation runtimes is inherent to most radiosity-based tools and has been long acknowledged by software developers and researchers (Howell 1969). An alternate approach involves the use of the probabilistic Monte-Carlo raytracing. The use of raytracing to solve view factor problems pertinent to building science has been demonstrated successfully by several studies in the past (Tregenza 1983).

This research focuses on demonstrating and critiquing the possibility of employing Radiance, an open-source raytracing tool, for calculating view factors. Radiance was developed in the mid-1980s at the Lawrence Berkeley National Laboratory by Greg Ward and has been under continued development since then (Ward and Rubinstein 1988; Ward et al. 1989; Ward et al. 1998; Ward and Heckbert 1992; Ward et al. 1988). For the last three decades, the primary research and development on Radiance has focused on lighting and daylighting applications. The following section provides a brief background of Radiance and describes the methodology for employing Radiance for view factor calculations.

Calculating view factors with Radiance

Radiance is essentially a collection of over 100 independent command-line programs that are invoked in customized sequences to perform different types of lighting and daylighting simulations. The core ray-tracing functionality attributed to Radiance is implemented by a few of these programs. These programs, which include rpict and rtrace, have traditionally relied on reverse-raytracing and ambient caching algorithms for performing lighting and daylighting simulations.

For calculating view factors through Radiance, one needs to rely on purely probabilistic Monte-Carlo raytracing. The ability to perform pure Monte-Carlo raytracing was recently introduced in Radiance through the development of a program called rfluxmtx. Rfluxmtx, as per its user manual, is meant to "compute flux matrices for Radiance scene". The flux matrices can be calculated for finite surfaces, light sources with solid-angle representations or differential areas. Differential areas are assigned as “rays” that contain a geometric location as well as a directional vector.

The surfaces or sources whose view factors are to be calculated need to be defined and categorized in terms of “senders” and “receivers”. The terminology of sender and receiver relates to the surfaces from which rays originate and terminate respectively. Only a single surface can be designated as a sender while there can be multiple receivers. The limitation of single sending surface, however, can be circumvented by specifying rays instead of sending surfaces. Figure 6 provides an overview of the manner in which Radiance commands can employed to calculate view factors.

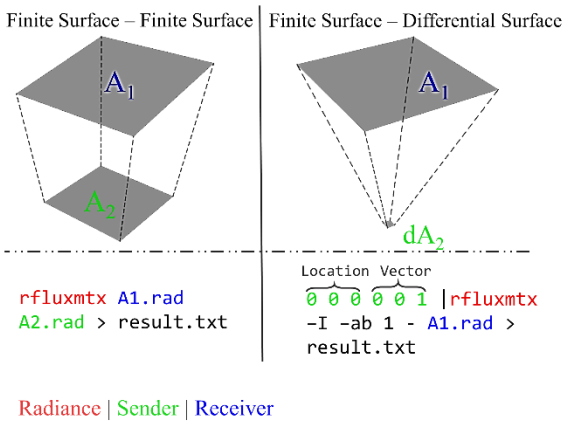


Figure 6. Examples of simple commands to perform view factor calculations with Radiance through rfluxmtx. Finite surfaces can be specified in Radiance by defining their geometry in the form of polygons and other geometric shapes. Differential surfaces are represented by rays.

For this research, the suitability of Radiance for calculating view factors was ascertained by comparing the results generated by rfluxmtx with those calculated through analytical approaches. One such validation is explained through Figures 7-9. Figure 7 shows two identical squares for whom view factors are to be calculated. Figure 8 shows a screen capture of the rfluxmtx command for calculating the view factors for the squares which are stored in a Radiance-compatible geometric format in the files A1.rad and A2.rad.

The three identical numbers shown in the results are on account of Radiance considering three channels for flux-transfer. The three-channel setup, while being critical for photopic lighting calculations, does not serve any specific purpose for view factor calculations.

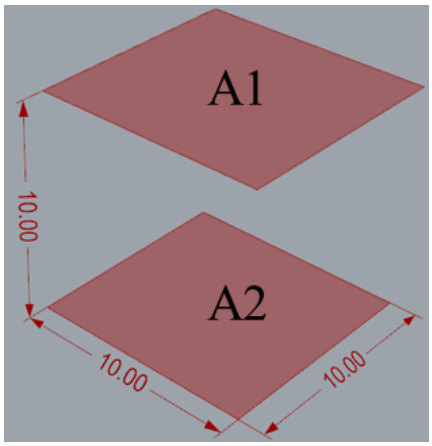


Figure 7. Two identical, parallel and directly opposed squares that were considered for the view factor calculation.

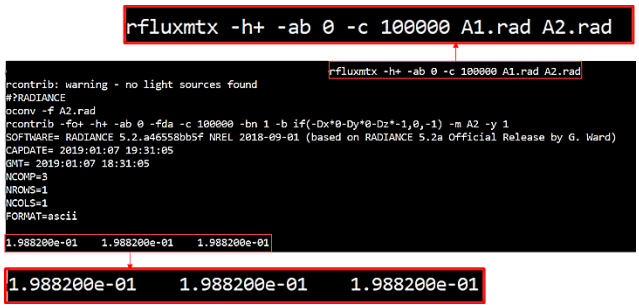


Figure 8. A screen capture of the view factor calculation performed through Radiance. The command-line arguments and the generated results have been magnified for clarity. The values A1.rad and A2.rad in the command-line arguments relate to the squares A1 and A2 in Figure 7.

The convergence of results is achieved by increasing the value of ray-sampling, specified through the -c flag in the command-line. A standard approach for convergence is to progressively increase the sampling parameter till the results in two successive iterations do not vary by a pre-defined tolerance margin. For the purposes of this research, the tolerance was set to 0.005.

Figure 9 shows the value of view factors calculated through the analytical approach. Assuming the analytical solution as the benchmark, the results captured in Figures 8 and 9 demonstrate that the error in the view factor calculated by Radiance is 0.5%.

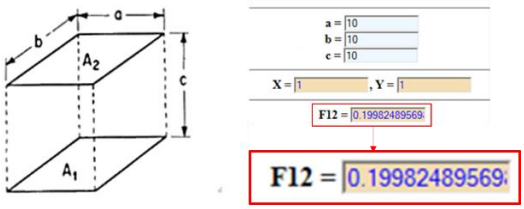


Figure 9. A screen-capture of the view factors calculated through analytical approach with a web-based tool. The dimensions of the parallel and identical rectangles and the distance between them are the same as the values shown in Figure 7 (Howell 2001).

Validation tests like the one described above were performed for calculating view factors of surfaces such as parallel circular discs and perpendicular polygons to ascertain the suitability of Radiance. In all instances, the calculation time was found to be within a few milliseconds.

View factors for surfaces with finite areas by employing surface subdivision

As previously explained, when calculating view factors between finite surfaces, Radiance limits the number of sending surfaces to one. This implies that for an input geometry with N surfaces, only 1xN view factors can be calculated per calculation. One approach to circumvent this issue is to subdivide the surfaces in the model such that their area can be assumed to be differential with respect to the overall input geometry. This approach is demonstrated through Figures 10-12. Figure 10 shows two finite areas whose view factors were calculated by assuming one of them to be constituted of identical subdivisions of differential area. As shown in Figure 11, rays were traced from the centre of the subdivided areas by assigning them as "senders".

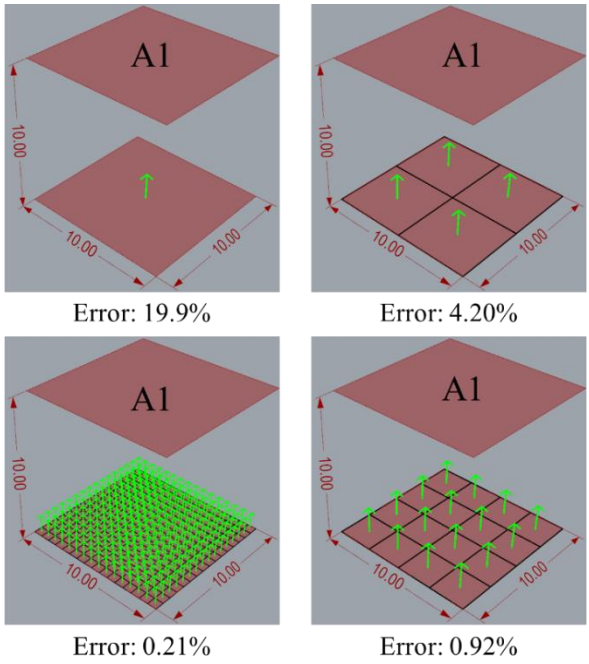


Figure 10. Calculation of view factors by subdividing a surface to identical elements. The error associated with each level of subdivision is shown below the respective image. The image on lower-left corner contains 256 subdivisions.

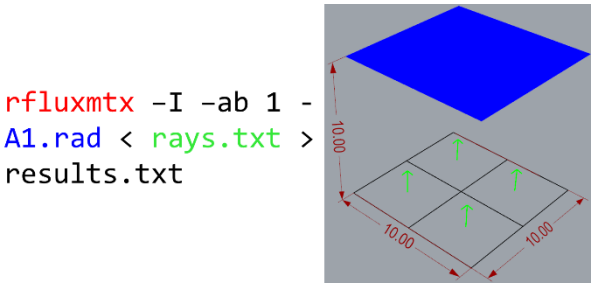


Figure 11. Rays traced from the centre of the subdivided areas are stored in a single file, referred to as rays.txt above.

As shown by the plot in Figure 12, by subdividing a finite area to smaller areas, it is possible to calculate view factors by using rays to represent finite surfaces. Radiance does not impose any practical limitations on the quantity of rays that can be specified as senders. So, this approach

can be employed to calculate view factors for multiple surfaces simultaneously.

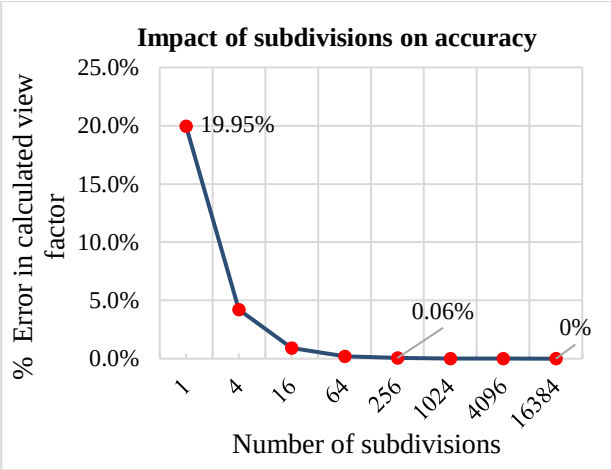


Figure 12. Error in view factor calculated through Radiance plotted against the number of subdivisions considered in the sending surface. As shown in Figure 11, each subdivision accounts for a single ray.

The next two sections investigate the suitability of employing Radiance for calculating view factors for geometries likely to be encountered in building simulations.

Case-study 1: Using Radiance to mimic the functionality of a standard radiosity-based tool

This scenario investigates the use of Radiance for calculating view factors for every surface for geometry representing an indoor space. As shown in Figure 13, the space comprises of 20 mesh faces. So, a full calculation of view factors would imply deriving the values for (20 x 20=) 400 individual view factors.

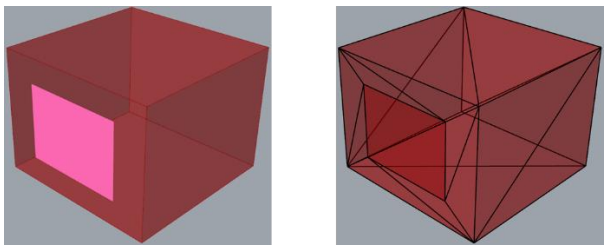


Figure 13. The image on the left shows the representative geometry for a room. The image on the right shows the underlying triangulated mesh. The structure comprises of 20 mesh faces.

The approach employed for calculating the view factors with Radiance involves subdividing the space and tracing individual rays outward from the centre of each mesh face. As explained previously, progressively subdividing the surfaces to trace more rays per surface will eventually lead to a convergence of the calculated view factor values. The subdivisions were generated through a plugin called Grasshopper for Rhino3D. Rhino3D is a modelling software that is commonly used for early stage modelling in architectural design practice. The subdivisions shown in Figure 14 are not rounded to factors of 10 because of the underlying meshing algorithm implemented in Rhino3D.

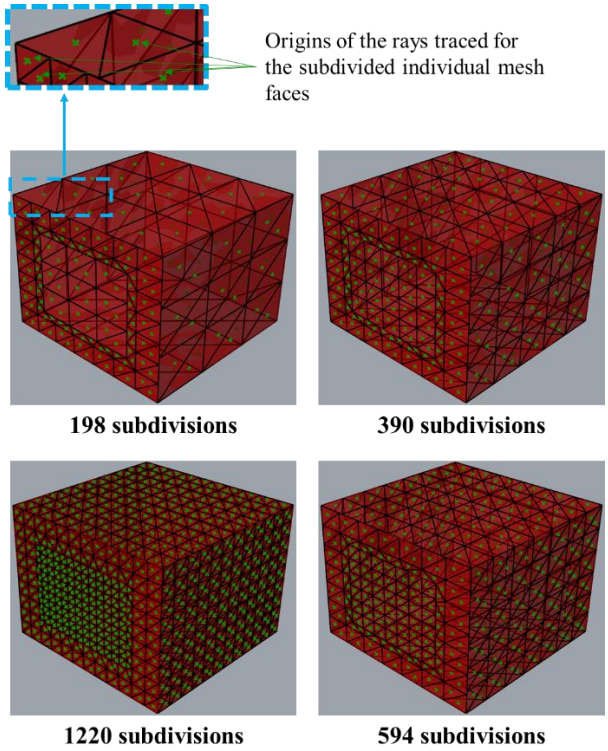


Figure 14. The four images show increasing levels of mesh subdivision for calculating view factors with Radiance by tracing rays from the centre of individual mesh faces. It follows that higher number of subdivisions will yield a more accurate solution at the expense of greater computational effort.

The rationale for subdividing the surfaces as shown in Figure 14, and performing the simulation for each of these cases, was to generate view factors that were within 0.5% of the values calculated by View3D. As described by the developer of View3D, it is meant to be used for calculations where the number of surfaces are less and the geometry is simple. This condition holds true for the geometry considered for this case study.

The time taken by View3D for calculating the view factors for the given geometry was 100 milliseconds. The time taken by Radiance to perform the calculation on a dedicated Intel Core i7-8700 3.2 GHz machine, is provided in Table 1. As indicated by the highlighted cells in the table, for a single processor run, the time taken for calculating view factors within acceptable accuracy was 4815 milliseconds. The runtimes were progressively reduced with an increase in the number of processors used

for the simulation. View3D does not have the functionality to invoke more than one processor at a time.

Table 1. Radiance simulation runtime, in milliseconds, for calculating view factors for all the surfaces in Figure 14. The cells highlighted in yellow indicate instances where convergence was reached and the results where within 0.5% of the results generated with View3D.

Subdivisions \ Processors	1 Proc.	2 Proc.	4 Proc.
198	1308	659	353
390	2541	1312	661
594	4815	2528	1273
1220	12448	6813	3601

Radiance simulations can be performed on multiple processors on Unix-like operating systems such as Linux, Mac OS or Free-BSD. Multi-process simulations are not supported on Windows-based machines. Figure 15 highlights the possibility of decreasing the simulation runtime afforded through multiple processors.

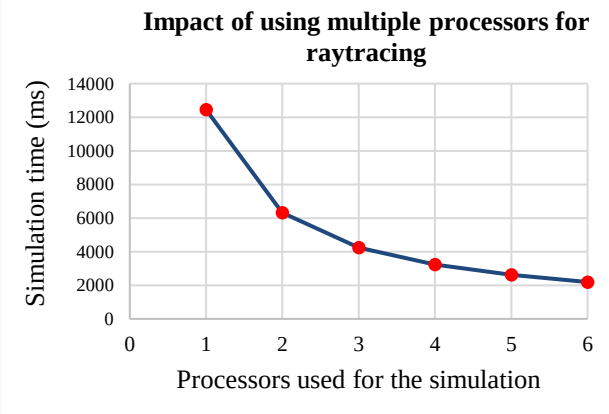


Figure 15. Simulation runtime as a function of the number of processors used for the simulation. All the simulation runtimes are for the model shown in Figure 8 that was subdivided to 1220 mesh faces. The machine used for this test was the same Intel i7-8700 3.2 GHz used for the comparison with View3D.

Based on the simulation runtimes observed for this case-study, it can be inferred that Radiance is unlikely to be a drop-in replacement for conventional Radiosity-based tools like View3D. However, the facility to calculate view factors selectively that is inherent to Radiance can be utilized to reduce the simulation runtimes in instances where view factor results are only required for only certain surfaces and not for the entire geometry. One such use-case is explored in the next case study.

Case-study 2: Calculating view factors for a subset of the total input geometry.

Certain applications involving the use of view factor calculations require the calculations only for a subset of the total geometry being considered. These include thermal comfort calculations, where the emphasis is on the radiation directly or diffusely incident on a human body manikin. The geometry setup for one such instance is shown in Figure 16.

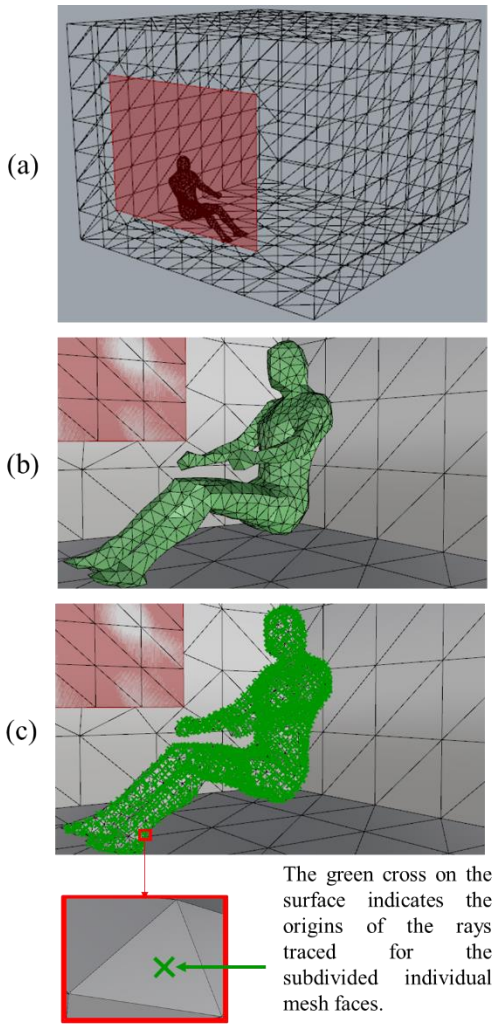


Figure 16. A typical setup for predicting thermal comfort on the human body, represented by a manikin, for locations near glazed surfaces. Image (a) shows the manikin within the context of the surrounding room geometry. Image (b) highlights the mesh faces that constitute the manikin and image(c) outlines the location on the meshes from which rays will be traced outwards through Radiance to calculate view factors.

For the setup shown in Figure 16, view factor values are required to account for radiative transfer between the mesh-faces on the manikin and the glazing polygon shaded in red. The view factor results thus obtained are used to estimate solar load on different parts of the manikin. It follows that while the entire geometry is to be

considered in the view factor calculation, the view factor results are only required for the mesh faces on the manikin. Additionally, the mesh faces that constitute the manikin are much smaller in dimension than rest of the geometry, thereby allowing them to be considered as differential for the purposes of the raytracing calculations.

The variation between the surfaces considered for the view factors calculation in View3D and Radiance is highlighted in Table 2. The total number of considered view factor calculations in the table indicates that in Radiance it is possible to specify the number of surfaces for which the view factors are to be calculated. As discussed earlier, this is because of the categorization of surfaces as "sending" and "receiving". In the present scenario, the manikin mesh faces will be assigned as "sending" surfaces that receive radiation from the glazing. The size of the surfaces constituting the manikin was assumed to be differential when compared with the rest of the contextual geometry of the room. This assumption was based on the observation that the area of the glazing surface ($4.32m^2$) is 1565 times the size of the largest manikin mesh sub-surface ($0.0027\ m^2$) and 5901 times the smallest manikin mesh sub-surface ($0.0007\ m^2$).

For the surfaces considered in the simulations, assuming View3D to be the benchmark, the average error in the 1336 view factors calculated through Radiance was found to be 0.0083 with a standard deviation of 0.0007. The accuracy of the results can be improved further by subdividing the manikin mesh faces in the same manner as shown in Figure 14.

Table 2. A comparison of required and actual view factor parameters between View3D and Radiance. The value of 4519876 is the square of 2126, the total number of surfaces in the model.

	View3D	**Radiance**
Total mesh faces in the model	2126	
Mesh faces in the manikin	1336	
View factors (to be calculated)	1336	
View factors (considered)	4519876	1336

As shown in Table 3, the ability to selectively calculate view factors for only certain aspect of the geometry results in a much lower simulation runtime with Radiance. This runtime can be further improved, as shown previously, by employing multiple processors.

Table 3. Simulation runtimes (in seconds) for calculating the view factors for the model shown in Figure 16. The term N.A. for View3D implies that multi-processing is not supported in View3D.

	Processors		
	1 Proc.	**2 Proc.**	**4 Proc.**
View3D	322.5	N.A.	N.A.
Radiance	14.2	7.8	4.1

Discussion and conclusion

This paper discussed the methodology and applicability of employing Radiance to perform view factor calculations.

Unlike standard radiosity-based tools, the approach for calculating view factors with Radiance requires a pre-identification of surfaces as "sending" and "receiving". This necessitates some extra effort in setting up the model for simulation. Additionally, accurate calculations require the sub-division of surfaces to increase the number of rays that are traced in the Monte-Carlo algorithm implemented by Radiance.

The results from the two case studies presented in this paper indicate that, with Radiance, the computational benefit in the form of lowered runtimes is likely to be gained in instances where the view factors are to be calculated for a subset of the total geometry. For generating view factor results for all the surfaces in the input geometry, as considered in the first case study, the Radiance-based approach was found to be approximately 48 times slower than View3D. It needs to be emphasized, however, that for the input geometry considered for that study, both View3D and Radiance were able to calculate the view factors within 5 seconds.

In the second scenario involving the calculation of view factors for only a subset of total surfaces in the geometry, Radiance was found to be approximately 23 times faster.

The multi-processing feature inherent to Radiance can be employed to curtail long simulation runtimes on compatible operating systems.

References

Appelbaum J. and Aronescu A (2016). View factors of photovoltaic collectors on roof tops. Journal of Renewable and Sustainable Energy 8(2):025302.

Arens E., Hoyt T., Zhou X., Huang L., Zhang H. and Schiavon S. (2015). Modeling the comfort effects of short-wave solar radiation indoors. Building and Environment 88(0):3-9

Ashdown, I. (1994). Radiosity: a programmer's perspective. John Wiley & Sons, Inc.

Cohen, M.F. and Greenberg, D.P. (1985). The hemi-cube: A radiosity solution for complex environments.

Proceedings of the 12th annual conference on Computer graphics and interactive techniques.

Hoffmann S., Jedek C. and Arens E. (2012). Assessing thermal comfort near glass facades with new tools. Center for the Built Environment, University of California at Berkeley.

Holman, J. (1986). Heat transfer. McGraw–Hill Book Company

Howell, J.R. (1969). Application of Monte Carlo to Heat Transfer Problems. In: Irvine TF, Hartnett JP, editors. Advances in Heat Transfer. New York, USA: Elsevier.

Howell, J.R. (2019). A catalog of radiation transfer configuration factors [Internet]. Available from: http://www.thermalradiation.net/indexCat.html

Howell J.R., Menguc M.P. and Siegel R. (2010). Thermal radiation heat transfer. CRC press.

Tregenza, P. (1983). The Monte Carlo method in lighting calculations. Lighting Research & Technology 15(4):163-170.

Walton, G.N. (2002). Calculation of obstructed view factors by adaptive integration. NIST.

Ward, G. and Rubinstein, F.M. (1988). A new technique for computer simulation of illuminated spaces. Journal of the Illuminating Engineering Society 17(1):80-91.

Ward G, Rubinstein F.M. and Grynberg A. (1989). Luminance in computer-aided lighting design. Proceedings of the 1st Building Simulation Conference. Vancouver, Canada.

Ward G., Shakespeare R., Ehrlich C., Mardaljevic J., Phillips E. and Apian-Bennewitz P. (1998). Rendering with radiance: the art and science of lighting visualization. San Francisco, CA, USA: Morgan Kaufmann Publishers

Ward, G.J. and Heckbert, P. (1992) Irradiance gradients. Proceedings of the Third Eurographics Workshop on Rendering.

Ward G.J., Rubinstein F.M. and Clear R.D. (1988). A ray tracing solution for diffuse interreflection. ACM SIGGRAPH Computer Graphics 22(4):85-92.