

## Fast and accurate ray-casting-based view factor estimation method for complex geometries

Sönmez, Furkan Fatih; Ziar, Hesam; Isabella, Olindo; Zeman, Miro

**DOI**

[10.1016/j.solmat.2019.109934](https://doi.org/10.1016/j.solmat.2019.109934)

**Publication date**

2019

**Document Version**

Accepted author manuscript

**Published in**

Solar Energy Materials & Solar Cells

**Citation (APA)**

Sönmez, F. F., Ziar, H., Isabella, O., & Zeman, M. (2019). Fast and accurate ray-casting-based view factor estimation method for complex geometries. *Solar Energy Materials & Solar Cells*, 200, 1-11. Article 109934. <https://doi.org/10.1016/j.solmat.2019.109934>

**Important note**

To cite this publication, please use the final published version (if applicable).  
Please check the document version above.

**Copyright**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights.  
We will remove access to the work immediately and investigate your claim.

# Fast and accurate ray-casting-based view factor estimation method for complex geometries

Furkan Fatih Sonmez, Hesam Ziar, Olindo Isabella, and Miro Zeman

Photovoltaic Materials and Devices Group, Delft University of Technology, 2628CD Delft, the Netherlands

**Abstract**— The concept of view factor has various applications in engineering problems, ranging from heat management to bifacial photovoltaics. Analytical solutions for view factor calculations are difficult to obtain and normally numerical methods are used. For complex geometries, when several surfaces are arbitrarily arranged in a three-dimensional environment, conventional numerical approaches such as Monte Carlo method will take a lot of simulation time. To tackle this challenge, we have developed a simple and yet accurate view factor estimation method based on ray-casting. In our method, the view factor is determined by sending out rays, evenly distributed in all directions from the target surface, and consequently counting the number of rays intercepted by each of the other surfaces present in the environment under study. Then, a simple algebraic procedure enables the estimation of a large number of view factors simultaneously. The results have been compared with exact and numerical solutions, proving that we have devised a fast and accurate view factor estimation method. This can be used to determine the view factors in environments generated via Light Detection And Ranging (LiDAR). The method has the potential to be applied in several scientific researches and engineering studies including heat transfer and solar energy.

**Index Terms**— view factor, shape factor, configuration factor, reflected irradiance, albedo, solar energy, ray casting, computer simulation, sky view factor (SVF), bifacial photovoltaics, building integrated photovoltaics

## I. INTRODUCTION

The view factor (also known as *shape factor* and *configuration factor*)  $F_{1-2}$  is defined as the fraction of radiation leaving area  $A_1$  that is intercepted by area  $A_2$ . In the same way,  $F_{2-1}$  is the fraction of radiation leaving  $A_2$  that is intercepted by area  $A_1$  [1, 2]. The view factor is a geometrical concept dependent only on the size, shape and orientation of the surfaces and the distance between them. It is often used in the field of heat and mass transfer, optics and rendering [3, 4, 5]. In view of such definition, all surfaces must be isothermal, opaque, and Lambertian and the media has no effect (scattering, emission and absorption) on the transfer of radiation between surfaces. Vacuum is such a medium and also other monatomic and most diatomic gases at low and moderate temperatures. In many engineering applications the medium does not affect the radiation heat transfer [2, 6, 7]. Therefore, within the scope of view factor algebra, to calculate the power that is transmitted between two surfaces, only the view factor and the net power of the source is required.

However, to be able to calculate the view factor for every situation, a more complex equation has to be solved. For the

radiation leaving a finite area  $A_1$  that is intercepted by area  $A_2$ , the related view factor  $F_{1-2}$  is defined as [1]:

$$F_{1-2} = \frac{1}{A_1} \int_{A_1} \int_{A_2} \frac{\cos \theta_1 \cos \theta_2}{\pi L^2} dA_2 dA_1, \quad (1)$$

where  $A_2$  is the area of the surface that intercepts the radiation from finite area  $A_1$ ;  $\theta_1$  is the angle between the normal vector to the area  $A_1$  and the line that connects the center of area  $A_1$  to  $A_2$ ;  $\theta_2$  is the angle between the normal vector to the area  $A_2$  and the line that connects the center of area  $A_2$  to  $A_1$ ; and, finally,  $L$  is the distance between the centers of  $A_1$  and  $A_2$ .

The view factors of several two- and three- dimensional configurations can be easily calculated using the corresponding algebraic formulas. However, as Equation (1) involves a double integral, it may turn to be a very challenging mathematical problem in many cases [1]. In such cases, numerical approaches are normally used.

Bopche and Sridharan [8] have tried to estimate the view factor analytically by applying the contour integral technique. Using this technique an estimation of  $\pm 5\%$  was achieved for a specific case. Computational times of several hours are not uncommon using this technique [8]. Sirimanna et. al. [9] have used a numerical method to estimate the view factors for a simple roof enclosure of five surfaces. In order to obtain an error of 1.0%,  $10^8$  iterations were required.

In order to estimate the view factor for more complex problems, the Monte Carlo method is often used. Vujicic et. al. [10] showed that using the Monte Carlo method an error range between 1.18% and 7.67% (with respect to the analytical solution) can be obtained, depending on the size to distance ratio of surfaces. The Monte Carlo method results will be different for each simulation run because of statistical error [11]. Also, it is not uncommon for the Monte Carlo method to result in simulation times of multiple hours [12].

Another method based on probability theory to estimate the view factor is proposed by Baronski et. al. [13] and makes use of the Chebyshev's inequality [14], while Vueghs et. al. [15] make use of the random hemisphere method in order to estimate the view factor from a point to a surface. Despite the low computation time, as they are based upon the probability theory, they are also subjected to statistical error.

This paper uses ray-casting method to find an approach that performs view factor calculation in a quicker and more accurate way. In Section II, we will describe how ray casting method can be used to estimate the view factors. Then,

Section III shows the capability of the introduced approach to estimate the view factor from a differential and finite surface to another finite surface. The estimations are then compared to exact mathematical solutions in terms of accuracy and simulation time as function of the number of rays. Applications of this method in urban energy areas will also be discussed in section IV. Finally, in Section V conclusions and highlights are presented.

## II. METHOD

The term radiation in the abovementioned definition of view factor may be considered to be either a wave or a particle [16]. In this study, to be able to use the ray-casting method, the light is considered as rays, which move from a source in certain directions. Light is then reflected from a surface either in specular or diffuse way, or both [17]. A perfect mirror reflects light in a specular way [18]. However, most materials reflect light diffusely [19]. To be able to use the view factor algebra, we assumed that materials reflect light all directions according to Lambert's cosine law [20, 21]. In this regard, to accurately determine the view factor from  $A_1$  to  $A_2$ , a large number of rays should be cast from  $A_1$  in all directions and be evenly distributed (i.e. the same number of rays for each solid angle).

There are several methods to achieve an even distribution of rays on a sphere, that is to densely pack points on a closed surface. One of them is the so-called Rusin's disco ball method [22], which requires a specific number of points to be packed on a sphere. Another method, proposed by Saff and Kuijlaars [23], allows for any positive integer number to be used for the amount of points to be packed on a sphere, but it can distribute fewer points on a sphere than Rusin's disco ball method. Another method, reported by González, uses in the simulation the so-called Fibonacci lattice method [24], which results in higher packing density on a sphere compared to Saff and Kuijlaars and is still able to allow any positive integer number of points to be put on a sphere. Finally, Boucher [25] created a method (coded in python), which is able to take any positive integer number  $x$  as an input and return an array of  $x$  points on a sphere evenly distributed on its surface. Based on Boucher's code, we wrote an equivalent script in C# language, which was then used for the simulation in this work.

When the surfaces are on a large distance from each other,  $A_1$  may be considered to be a differential surface  $dA_1$  [26]. In this way, such a radiating surface can be considered as a differential surface and thus the simulation has to be done only once to obtain the view factor. If surfaces are too close to each other, the radiating surface may no longer be considered to be a differential surface. For these cases, the radiating surface has to be divided into smaller surfaces, which may in turn be considered to be differential surfaces, depending on their size and distance to the receiving surface (see Section III.3).

Assuming that the  $dA_1$  is a flat plane (e.g. one cell of a photovoltaic module), each emitted ray will be assigned an angle factor according to the cosine of its angle. In other words, each ray  $x_i$  emitted by  $dA_1$  that hits  $A_2$  will be counted and multiplied by its weight factor  $\cos\theta_i$  resulting in the total number of rays  $X$  that hit  $A_2$  corrected according to Lambertian cosine law. The angle  $\theta_i$  refers to the outgoing angle of the  $i$ -

th ray with respect to the normal vector of the emitting area  $dA_1$ . Mathematically, this can be stated as follows:

$$X = \sum_{i=1}^{i=n} x_i \cos \theta_i, \quad (2)$$

where  $n$  is the total integer number of rays that hit  $A_2$ . Also, every ray  $y_j$  cast from  $dA_1$  will be counted and multiplied by its weight factor  $\cos\theta_j$  resulting in the total number of rays  $Y$ , corrected according to Lambertian cosine law in the same way resulting in equation (5):

$$Y = \sum_{j=1}^{j=m} y_j \cos \theta_j, \quad (3)$$

where  $m$  is the total number of rays cast and the angle  $\theta_j$  has the same meaning of the abovementioned angle  $\theta_i$ . Dividing the total number of rays hitting  $A_2$ ,  $X$ , by the total number of rays cast from  $dA_1$ ,  $Y$ , results in the desired view factor from  $dA_1$  to  $A_2$  ( $F_{d1-2}$ ):

$$F_{d1-2} = \frac{X}{Y} = \frac{\sum_{i=1}^{i=n} x_i \cos \theta_i}{\sum_{j=1}^{j=m} y_j \cos \theta_j}. \quad (4)$$

To test equation (4) in a complex geometry, we have considered urban environment. One engineering exemplary case is a photovoltaic (PV) module (as  $A_1$ ) installed on a roof in urban area surrounded by buildings (as  $A_2$ ). In such a situation, most PV modules are placed such that they are visible to a great amount of surface area. The surfaces visible to the PV module may be considered to be on a large distance. Therefore, PV module is treated as  $dA_1$ . Figure 1 shows the position of a PV module (green tilted rectangle) in a location in Delft, the Netherlands, as rendered in Google maps [27]. As each infinitesimal surface contributes to the reflected irradiance on the PV module, the view factor has to be known for all of these surfaces. In order to make this a finite problem, the infinitesimal surfaces will be clustered into small surfaces with a finite area. In order to increase the accuracy of the simulation, the size will be kept as small as possible. However, decreasing the area will increase the number of surfaces inside the simulation domain. Figure 2 shows the visible surfaces to the PV module (in red), where surfaces closer to the PV module are smaller while the surfaces farther from the PV module are larger.

For urban PV purposes, the view factor from all reflecting surfaces to the PV module is needed for assessing the diffuse reflected component as each of these surfaces will contribute to the total reflected irradiance. However, as there may be

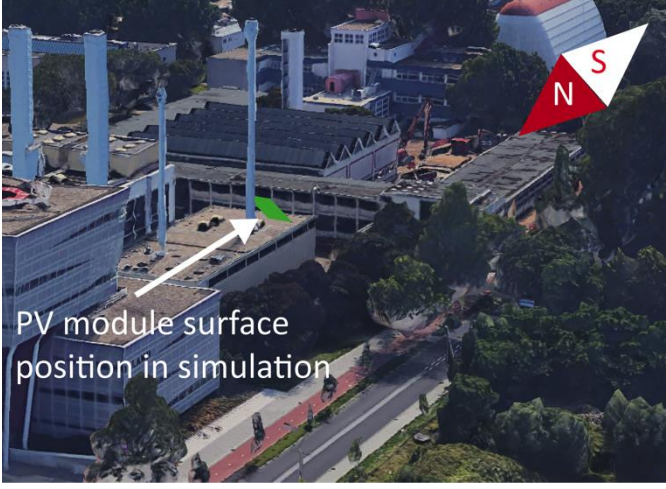


Figure 1- Google maps [27] view of an example simulation location (roof of the thermal station in Delft, Latitude 51.999693N, Longitude 4.368932E) showing an imaginary PV module placed inside the simulation environment (depicted in green).

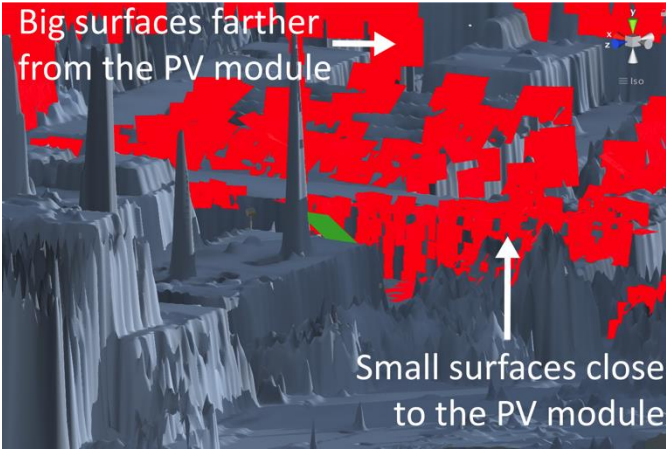


Figure 2 – 3D reconstruction of the thermal station in Delft in the Unity3D Game engine. The imaginary PV module (depicted in green) is where the rays will be cast. The visible surfaces to the PV module are depicted in red, where the surfaces closer to the PV module are smaller and the surfaces farther from the PV module are larger. This 3D surface was created using LiDAR data imported from the AHN height map [28] with a height resolution of 0.5 meters.

several thousands of surfaces visible to the PV module, performing a ray casting simulation for obtaining the view factor from each of these surfaces to the PV module would be an extremely time-demanding task. Alternatively, we may obtain the view factor from the PV module to all reflecting surfaces and then apply the so-called reciprocity rule [1]. As a single ray will never hit multiple surfaces, each ray will have its own hitting surface. By counting the number of ray hits on each surface, individually per each surface (again, using Lambertian cosine law), the numerator of equation (6) is found for each reflecting surface separately. The denominator of equation (6) is equal for each reflecting surface, as it resembles the total number of rays cast during the ray casting simulation. In this way, a single ray casting simulation will result in the view factors from the PV module to every reflecting surface inside of the simulation. From the symmetry of equation (3), the view factor of an area  $A_1$  that intercepts the radiation from a finite area  $A_2$  can be written as follows:

$$F_{2-1} = \frac{1}{A_2} \int_{A_2} \int_{A_1} \frac{\cos \theta_2 \cos \theta_1}{\pi L^2} dA_1 dA_2, \quad (5)$$

Combining equations (1) and (5) yields:

$$A_1 F_{1-2} = A_2 F_{2-1}, \quad (6)$$

which is the reciprocity rule of the view factors. Equation (6) implies that the view factor  $F_{1-2}$  is related to view factor  $F_{2-1}$  by the ratio of surface  $A_1$  and  $A_2$  [1, 2].

Using the reciprocity rule makes it easy to find the view factors from the radiating surfaces to the PV module if the surface area of both the PV module and the surfaces are known. A flowchart showing the processes taken in order to obtain the view factors from the radiating surfaces to the PV module can be seen in *Figure 3*.

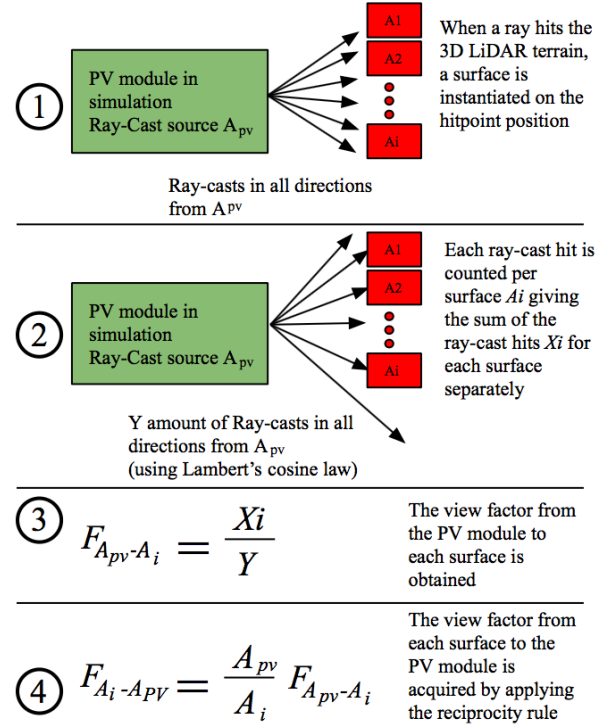


Figure 3 – The sequential processes taken to obtain the view factors from the surfaces to the PV module are here reported. (1) Rays are cast in all directions in order to find the 3D LiDAR-induced terrain visible to the PV module. Surfaces are instantiated on these terrain-hit points. (2) Y number of rays are cast in all directions from the  $A_{pv}$  and each ray hit is counted per surface. (3) The view factor from  $A_{pv}$  to each surface  $A_i$  is calculated. (4) Using the reciprocity rule, the view factor from each surface  $A_i$  to  $A_{pv}$  is obtained.

As a corollary, each ray that does not hit any surface is cast to the sky. By counting rays that do not hit any surface in the same way as done in equation (2), the SVF, which is the view factor from the PV module to the sky dome, can be easily estimated by applying equation (4). In case of blocked horizon, this method eliminates the need of estimating the SVF using a fish-eye lens camera, a time-consuming process that requires field work and image processing.

### III. METHOD VALIDATION

In order to validate the introduced approach of ray casting for the accurate estimation of the view factor between two surfaces, simulations have been made and compared with equivalent mathematical solutions.

#### 1) View factor comparisons with differential cases

View factor estimation for differential surfaces is the foundation of the view factor estimation for finite surfaces, as it is required to estimate the finite cases. Reflective surfaces may be curved or flat. In order to validate the ray-casting model, several results of mathematical solutions of view factors are compared with the results of our computer simulation.

The exact solution of the view factor for a differential sphere to a sphere with radius  $r$  can be calculated, since the geometry is relatively simple. According to Chung et. al. [29], the view factor from a spherical point source, or differential sphere to a sphere can be found from the exact solution of:

$$F_{d1-2} = \frac{1}{2} \left( 1 - \sqrt{1 - K^2} \right), \quad (7)$$

where  $K$  is defined as  $K = r/h$ ,  $h$  is the distance from the differential sphere 1 to the center of sphere 2 and  $r$  is the radius of sphere 2. As an example,  $h$  and  $r$  have been set to 200 and 50, respectively, in the equation (7), resulting in a view factor of 0.015877. When casting  $2 \times 10^5$  rays from sphere 1 in the simulation, 3175 rays are intercepted by sphere 2. This situation as used in the simulation has been illustrated in Figure 4.

Using equation (4), the view factor from the ray-casting simulation can be calculated, yielding a view factor of 0.015875. As the radiating surface is a sphere, Lambertian cosine law is not applied. With a mere 0.013% difference between the simulation and the exact solution, our method basically delivers the same results as the exact solution. As the distance  $h$  becomes larger or the radius  $r$  becomes smaller, the view factor will become smaller, thus decreasing the accuracy. This is explained more in Section III.2.

For the application of urban PV, as a PV module is not a sphere, but it is often considered to be a planar object, our ray-casting approach has to work also for more realistic cases. According to Hamilton and Morgan [30] the exact solution of the view factor from a differential planar surface  $dA_1$  to a finite parallel planar rectangular surface  $A_2$ , where the normal of surface  $dA_1$  passes through the corner of surface  $A_2$  can be found by the following equation:

$$F_{d1-2} = \frac{1}{2\pi} \left( \frac{A}{\sqrt{1+A^2}} \tan^{-1} \left( \frac{B}{\sqrt{1+A^2}} \right) + \frac{B}{\sqrt{1+B^2}} \tan^{-1} \left( \frac{A}{\sqrt{1+B^2}} \right) \right), \quad (8)$$

where  $A = a/c$  and  $B = b/c$  with  $a$  and  $b$  being the sides of the rectangular surface  $A_2$  and  $c$  being the distance between the

differential surface  $dA_1$  and the corner of surface  $A_2$  through which the normal of surface  $dA_1$  passes (see Figure 5).

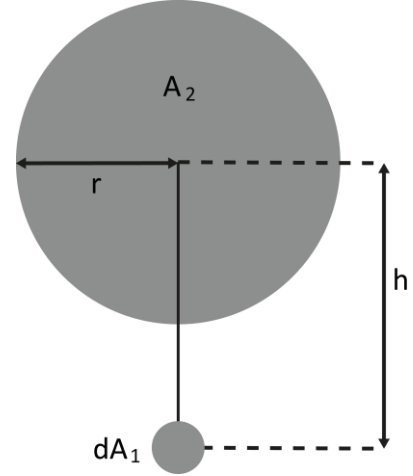


Figure 4 - The view factor for a differential sphere  $dA_1$  to sphere  $A_2$ , where  $h$  is the distance between differential sphere 1 and the center of sphere 2, which has a radius  $r$ .

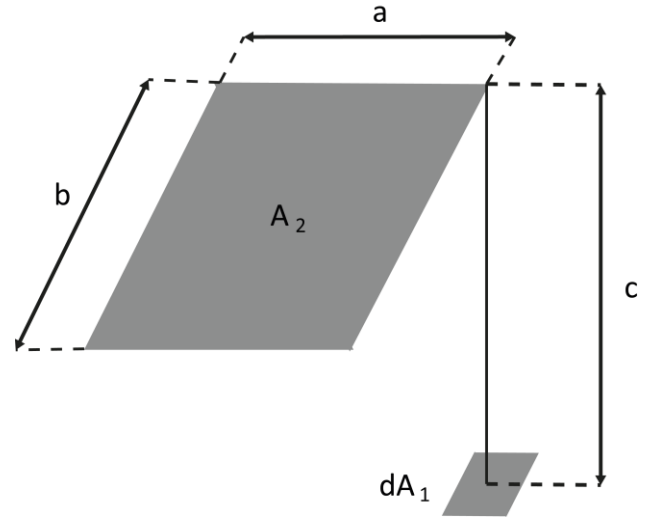


Figure 5 - The view factor from  $dA_1$  to  $A_2$ , where  $dA_1$  is the radiating differential surface and  $A_2$  is a finite receiving surface.

To compare the results of the simulation to the exact value of  $F_{d1-2}$ , the values 300, 200 and 100 are chosen for  $a$ ,  $b$  and  $c$ , respectively, in both the equation and the simulation. Using then equation (10), the exact view factor  $F_{d1-2}$  is 0.21758. According to the carried-out simulation, when using the same exact geometry and parameters, 10878 rays hit the receiving surface  $A_2$  with respect to a total of  $2 \times 10^5$  emitted rays, yielding a view factor of 0.21756. The difference between the exact value of the view factor and the estimated view factor using ray casting has a negligible difference equal to 0.009%.

The difference in percentage will be different with respect to the distance between the surfaces and the length and width of the receiving surface. As the distance between surfaces increases and/or as the length and width of the surfaces decreases, the view factor becomes smaller. As the view factor becomes smaller, the difference in percentage for the same amount of ray casts will become larger because the simulation is only accurate up to a certain decimal point. The sphere



example has a difference in percentage of 0.013%, while the rectangular surface example has a difference in percentage of 0.009%, which would suggest that the second surface example is more accurate. For the same amount of casted rays, however, the sphere example is accurate up to 5 decimals, while the surface example is accurate up to 4 decimals.

### 2) Accuracy of the simulation with respect to amount of the rays cast

Depending on the scenario for which the simulation is used, different amounts of casted rays will be sufficient.

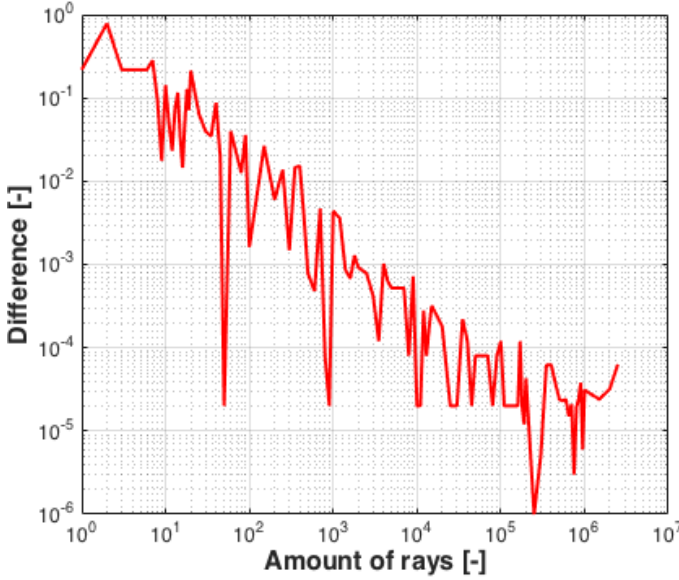


Figure 6 – The difference between the exact mathematical solution value and the simulated value for certain ray cast amounts for the case of a differential surface to a finite surface as seen in figure 5.

Figure 6 gives the difference between the view factor obtained from the exact mathematical solution and the simulation result for a range of different total number of rays cast for the case of a differential surface to a finite surface as shown in Figure 5. Clearly, as the number of rays increases, the accuracy of the simulation increases as it approaches the real view factor value. Table 1 gives estimation bounds with regard to the total number of casted rays. In order to achieve a view factor accurate within 2 decimals, 1000 rays have to be cast in total. For this study case, ten times the inverse estimation bound gives the required number of rays.

Table 1 – Required casted rays to achieve a view factor estimation within certain bounds for the case of a differential surface to a finite surface as seen in Figure 5.

| VF Estimation bounds | Required ray casts |
|----------------------|--------------------|
| $\pm 10^{-01}$       | $10^2$             |
| $\pm 10^{-02}$       | $10^3$             |
| $\pm 10^{-03}$       | $10^4$             |
| $\pm 10^{-04}$       | $10^5$             |
| $\pm 10^{-05}$       | $10^6$             |

### 3) View factor comparisons for the finite case

The ray casting simulation proved to be usable to estimate the view factor from a differential surface to a finite surface. However, in several cases, radiating surfaces may not be

considered to be differential. In order to evaluate whether the proposed model can estimate the view factors from a finite surface to another finite surface, the simulation procedure is repeated for different positions on the radiating finite surface, where after the different view factors are averaged to obtain the view factor from the entire radiating finite surface to the receiving finite surface. This makes the estimation for finite cases much more time consuming, but more useful and accurate.

To validate our ray-casting approach for the finite case, several results of mathematical solutions for view factors are compared with the results of the simulation. Again, before a comparison can be made, the geometry of both cases has to be the same.

The mathematical exact solutions for finite cases quickly become more complex as compared to the differential cases. For some relatively simple geometries the exact solutions are available. An example of such a case is the view factor for two identical, parallel, directly opposed rectangles. An illustration of the case is shown in Figure 7.

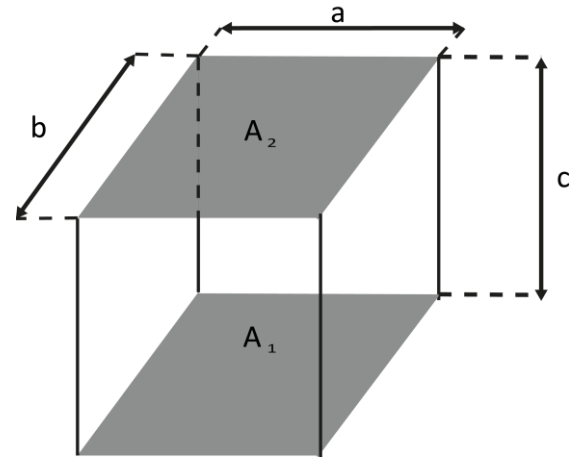


Figure 7 – The view factor for two identical, parallel, directly opposed rectangles  $A_1$  and  $A_2$ .

In Figure 7,  $a$  and  $b$  are the length and width of the rectangles and  $c$  is the distance between them. According to Hamilton and Morgan [30] the exact solution for the view factor from a finite planar surface to an opposing identical finite parallel planar rectangle can be found by the following exact solution:

$$F_{1-2} = \frac{2}{\pi AB} \left( \ln \left( \sqrt{\frac{(1+A^2)(1+B^2)}{1+A^2+B^2}} \right) + A\sqrt{1+B^2} \tan^{-1} \left( \frac{A}{\sqrt{1+B^2}} \right) + \tan^{-1} \left( \frac{B}{\sqrt{1+A^2}} \right) - A \tan^{-1} A - B \tan^{-1} B \right) \quad (9)$$

where  $A = a/c$  and  $B = b/c$ . When taking 1 for  $a$ ,  $b$  and  $c$ , the resulting view factor  $F_{1-2}$  is 0.199825.

Since the radiating surface is now a finite surface as well, apart from the number of rays casted in the simulation, also

the distribution of the radiating points on the finite surface is now required. We suggest to distribute the points from which the rays will be cast evenly over the surface of the radiating finite surface, as shown in *Figure 8*. When casting  $10^5$  rays from 9 different points ( $9 \times 10^5$  rays in total), the resulting view factor is 0.1685016. The difference in percentage between the exact mathematical solution and the simulation result is 17.0085%, which is a rather large difference. Instead of casting a large number of rays from a single point, we can also cast a smaller number of rays from more points distributed on the finite surface. When casting from  $95 \times 95 = 9025$  different points (evenly distributed over the surface), where only 100 rays are cast per point, again  $9 \times 10^5$  rays in total, the estimated view factor is 0.1980125. The difference between the exact mathematical solution and the new simulation result is now 0.911176%, which is a very large improvement considering that the total number of casted rays is the same for both approaches. When increasing the number of points as opposed to increasing the number of rays, the view factor becomes even more accurate, as  $151 \times 151 = 22801$  ray-cast points result in an estimated view factor of 0.1985698, which is even closer to the exact value of the view factor with a difference in percentage of 0.630129%.

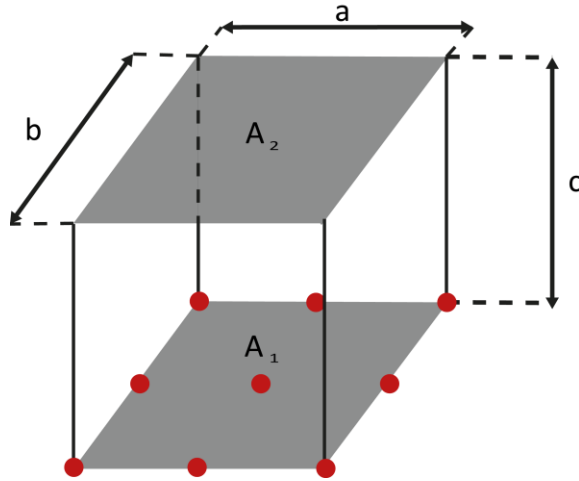


Figure 8 – The points on the radiating surface  $A_1$  (in red) from which the simulations are performed.

#### 4) Ray-cast ratio

While the number of points from which the rays are casted is important for the simulation, the number of casted rays from a point is still important as it determines the accuracy of each point. As the total number of casted rays is a multiplication of the total number of radiating points and the casted rays from each of such points, the total number of rays cast increases exponentially for an increasing number of points on a side. If there are only a few rays casted from each point, the number of points does not matter, as the simulation will be rather inaccurate. Similarly, if a large number of rays is cast from a few points, again the simulation will be inaccurate.

As a figure of merit (FoM), the ratio between the total number of casted rays and the difference between the exact solution and the simulated result is found to determine the optimal number of rays cast from each point. The FoM has

been plotted against the total number of casted rays and reported in *Figure 9*.

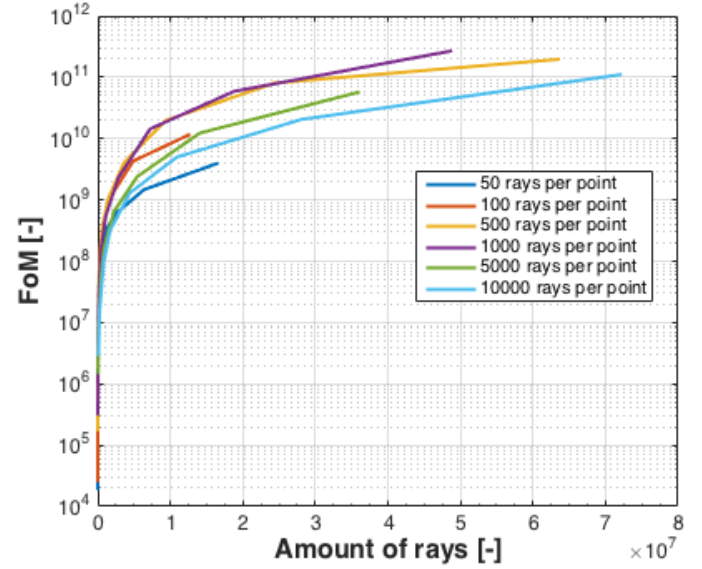


Figure 9 – The Figure of Merit (FoM) is the ratio between the total number of casted rays and the difference between the exact solution and the simulated result. Here, it is plotted against the total number of ray-casts. The abovementioned difference refers, in this case, to the difference between the exact mathematical value obtained from equation 9 and the simulation result. As the ratio increases, the accuracy increases.

It becomes clear that the greatest accuracy per ray-cast is found for 500 or 1000 rays cast per point, where 1000 rays seem to perform slightly better in the greater amounts of ray-cast. When using 1000 rays at 48841 points on the radiating surface, the total number of rays cast is  $4.8 \times 10^7$ . These parameters result in the estimated simulation result for the view factor of 0.199644, whereas the exact solution equation gives 0.199825 for the case of *Figure 8*. The resulting difference in percentage is 0.0906203%.

*Table 2* shows the rays required to estimate within certain decimal points of accuracy limit for the case study shown in *Figure 7*. A comparison between *Tables 1* and *2* reveals that ten to a thousand bigger number of rays are required to achieve a similar accuracy bound for the finite case compared to the differential case.

Table 2 – Required ray casts to achieve a view factor estimation within certain bounds for the finite case of 1000 rays per point as seen in *Figure 8*.

| VF Estimation bounds | Required ray casts |
|----------------------|--------------------|
| $\pm 10^{-01}$       | $10^3$             |
| $\pm 10^{-01}$       | $10^4$             |
| $\pm 10^{-02}$       | $10^5$             |
| $\pm 10^{-03}$       | $10^6$             |
| $\pm 10^{-03}$       | $10^7$             |

#### 5) Simulation time

The main advantage of the ray-casting method is the reduction of simulation time as thousands of view factors are determined simultaneously. Still, millions of rays are casted in order to achieve a high accuracy. Each ray-cast requires computational time. In this work, an entry-level computer with an Intel Core i5 2.7 GHz processor has been able to simulate a

maximum of approximately 3000 rays per second. *Table 3* shows the time required to estimate within certain decimal points.

In order to achieve a view factor accuracy of 3 decimals, the required amount of simulation time is then approximately between 5.5 minutes and 55 minutes. In order to achieve this simulation time, a single processor core was used. Multi-core processing would reduce the simulation time drastically [31]. However, if the simulation had to be done for each surrounding surface separately, the simulation time would have to be multiplied by the number of surfaces instantiated in the simulation, which would make the simulation impractical for many cases.

Table 3 – Required ray casts and the corresponding simulation time to achieve a view factor estimation within certain bounds for the finite case of 1000 rays per point as seen in Figure 8.

| VF Estimation bounds | Required ray casts | Simulation time (s)  |
|----------------------|--------------------|----------------------|
| $\pm 10^{-01}$       | $10^3$             | $3.3 \times 10^{-1}$ |
| $\pm 10^{-01}$       | $10^4$             | $3.3 \times 10^0$    |
| $\pm 10^{-02}$       | $10^5$             | $3.3 \times 10^1$    |
| $\pm 10^{-03}$       | $10^6$             | $3.3 \times 10^2$    |
| $\pm 10^{-04}$       | $10^7$             | $3.3 \times 10^3$    |

#### IV. EXTENSION TO MORE COMPLEX GEOMETRIES (EXAMPLE OF PV MODULES IN URBAN ENVIRONMENT)

The surroundings of a real PV array may consist of various surfaces of different geometries. Each of the surfaces will have a different view factor to the PV module, which means that each of the surfaces will reflect light to the PV modules in different intensities. The quantities of these surfaces can be very high (specially in urban areas), depending mostly on the size of each surface. Our proposed ray-casting method is especially usable when estimating multiple view factors simultaneously. Since each ray can only hit one surface and the total number of casted rays is the same for all of the surfaces, multiple view factors can be estimated using only a single set of ray-casts.

The Unity3D game engine was used for the simulation as it allows the import of LiDAR data obtained from the Actueel Hoogtebestand Nederland (AHN) [28] and their transformation into polygons. These are able to intercept rays and are used to position also the PV module as they form the simulation environment. Such a virtual environment is similar to the real world with a spatial resolution of 0.5 meters. The surfaces are instantiated onto the 3D simulated terrain, which is visible to the PV module.

Referring to *Figure 3*, when estimating the view factors, all the surfaces surrounding a PV module and visible to it, are defined. A first batch of rays is casted in all directions to determine the surfaces visible to the PV module. Small rectangular surfaces are instantiated on each location where a ray hits the environment. The view factors will be determined from each of these surfaces to the PV module. The size of the instantiated surfaces can be constant or dynamic. The sizes as well as the distance are the variables, which affect the view factor. A small sized surface or a large distance will decrease the view factor, making the reflection from the surface less significant. Keeping the amount of instantiated surfaces low will improve the performance of the simulation in terms of

time. Instantiating large surfaces will decrease the number of surfaces. A ratio can be applied to define the size of the instantiated surface based on its significance. Based on the experience obtained from various simulations, the suggested ratio for an urban environment is to keep the size of the instantiated surfaces constant (e.g.  $2 \times 2 \text{ m}^2$ ) up to a distance of 30 meters and to increase the size of the instantiated surface by 0.5 meter per side for every 1 meters of distance after that. The size of the instantiated surfaces will increase as the distance between the surface and the PV module increases, making the surfaces that are farther away more significant as they will receive more rays even if they are far away, while keeping the amount of surfaces as low as possible. Reciprocity rule

Our ray-casting method determines the view factor from the PV module to each of the surrounding surfaces, which are seen in *Figure 10*. However, the desired view factor is from each surrounding surface to the PV module as it is the direction in which the radiation is reflected. When the areas of both the radiating surface as well as the receiving surfaces are known, the view factor from each surface to the PV module can be determined using equation (6) [26]. This makes it extremely easy and fast to estimate thousands of view factors accurately using only a single simulation run.

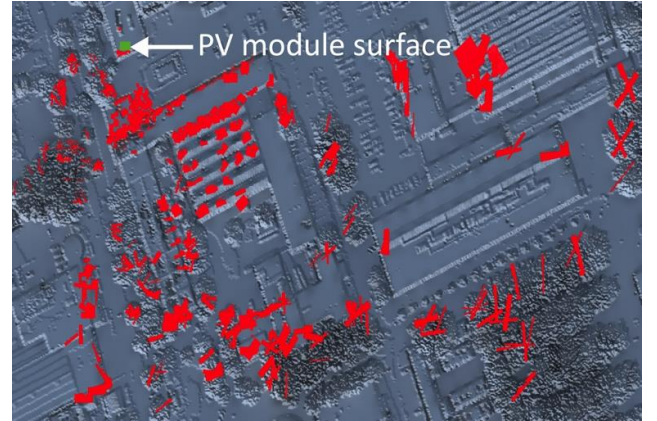


Figure 10 – A Unity3D simulation top view, showing the imaginary PV module (in green) and the surfaces visible to the PV module (in red). Each of these surfaces contributes to the total reflected irradiance on the PV module (i.e. to the albedo component of the irradiance).

##### 1) Application in calculation of albedo component in urban areas

The irradiation received by a PV module has three components, consisting of (1) the direct component, which is the irradiance radiated directly from the sun, (2) the diffuse component, which is the irradiance radiated from the whole sky dome, and (3) the albedo component, which is the irradiance radiated as a result of reflection from the surrounding surfaces [32]. The amount of irradiance reflected from a uniformly illuminated surface is given by:

$$I^{\text{reflected}} = I^{\text{incident}}_{\text{surface}} R, \quad (10)$$

where reflectance ( $R$ ) is the fraction of the incident irradiance which is reflected [33].  $I^{\text{incident}}_{\text{surface}}$  and  $I^{\text{reflected}}$  are the amount



of radiation received by and reflected from that surface, respectively. The amount of irradiance that a PV module will receive from the reflective surface ( $I_{PV}^{received}$ ) depends on the geometrical positioning of the PV module, reflective surface and the environment. Consequently, the amount of irradiance radiated from all reflective surfaces to the PV module is given by:

$$I_{PV}^{received} = \sum_{K=1}^{K=N} I_K^{incident} R_K F_{K-PV}, \quad (11)$$

where  $F_{K-PV}$  is the view factor from the surface  $K$  to the PV module.  $N$  is the total number of instantiated reflected surfaces in the simulation. Each of these surfaces has its own incident irradiance, reflectance and view factor. The last is found using the ray-casting method.

Note that the application of our method for urban PV (in the form of equation (11)) does not give information about the specular reflection from the surrounding, such as glare effect [34], since we consider all surfaces to be diffuse reflectors in our model. Also, the chance of receiving specular reflection for PV module in urban environment is lower than diffuse reflection because most of the urban environment materials (specially the roofs) does not cause glare. Besides, if specular reflection happens, the duration of specular reflection is very low because Sun position is changing but reflector and receiver (PV module in this case) are fixed. Moreover, in the case of Netherlands, as most of the time sky is cloudy or overcast, materials reflect light diffusely as there is no direct component from the Sun.

## 2) Application in remote SVF estimation for urban areas

Sky view factor (SVF) is a unit-less quantity that represents the ratio at a point in space between the visible sky and a hemisphere centered over the studied location [35]. SVF lies between zero and one.  $SVF = 0$  means the entire sky is blocked from view by obstacles and when the horizon is free:  $SVF = 1$ . SVF is a useful value in urban heat-island studies [36] and also urban PV system modelling [37].

SVF is a special case of view factor. A direct application of our proposed method is the remote estimation of SVF. Using LiDAR data, seven locations in the TU Delft campus were chosen for this investigation. On-site SVF measurements were taken using a hori-catcher device [38] and the results were compared to the simulation based on our ray-casting approach. Figure 11 shows the seven locations and their corresponding sky view from a height of 0.6 m above the ground. There, we use the tool of sky grids as formulated by Steyn [39] and further refined by Calcabrini et. al. [40] to estimate the measured SVF. To simulate the SVF at each location, rays were casted from a single point (i.e. the hori-catcher location) in all directions according to Lambertian cosine law in the same way that was done for the view factor estimation method described before. However, instead of counting the number of rays hitting a surface, the number of non-hitting rays (i.e. the number of rays that are “lost” to the sky  $X$ ) are counted.

As it can be seen in Table 4, our ray-casting approach is capable of accurately estimating the SVF in urban areas with complex geometries. With an average error for the seven different cases of only 3.82%, our ray-casting method results to be easy-to-implement, cost-effective, and time-saving for SVF estimation in urban studies.

Table 4 – Result of comparison for SVF estimation and measurement

| Location no. | Simulated SVF | Measured SVF | Error (%) |
|--------------|---------------|--------------|-----------|
| 1            | 0.724         | 0.7031       | 2.97      |
| 2            | 0.8333        | 0.8236       | 1.18      |
| 3            | 0.541         | 0.527        | 2.66      |
| 4            | 0.2613        | 0.2542       | 2.79      |
| 5            | 0.3313        | 0.3096       | 7.01      |
| 6            | 0.521         | 0.4992       | 4.37      |
| 7            | 0.6891        | 0.6513       | 5.80      |

We observe that the simulated SVF is slightly bigger than the measured counterpart. We believe that this biased error is mainly due to the fact that LiDAR data is limited by its grid resolution, which is 0.5 meters [28]. This results in many details in the surroundings to be left out, which in reality will have some effect on the SVF. Also, the walls of buildings and other objects are slightly inclined, while in reality the walls are perfectly vertical. This may also result in a slightly higher simulated SVF as compared to the measured SVF. For the SVF simulations,  $10^5$  rays were cast, requiring a simulation time of 5 seconds. Using more casted rays and accepting a somewhat longer simulation time, even more precise results are expected.

## V. CONCLUSION

In this paper, a simulation approach was presented for view factor calculations in complex geometries or urban environment. The ray-casting method only requires the environment around the surface under study (e.g. a PV module). The three-dimensional environment can be obtained from LiDAR data and formed into polygons which are able to intercept rays. The rays are cast from a point in all directions according to the Fibonacci lattice method. This method allows an even distribution of an arbitrary number of points onto a sphere. This way, no random number generator has to be used in the simulation. The effects of distance, angle and size are incorporated in the method automatically. When comparing the result of a view factor simulation to examples of exact solutions, very small differences were observed. As the ray-casting method is able to estimate multiple view factors, the simulation time is decreased considerably compared to performing the simulation for each surface. This makes our approach a viable method to estimate the albedo component of the irradiance on a PV module. That is, it enables more accurate energy yield estimation of building integrated PV (BIPV) modules or bifacial PV modules, since the albedo component may contribute significantly to the irradiance on these types of PV modules. Also, the ray-casting model is usable for the estimation of the sky view factor, which can be used in urban energy and urban climate studies.

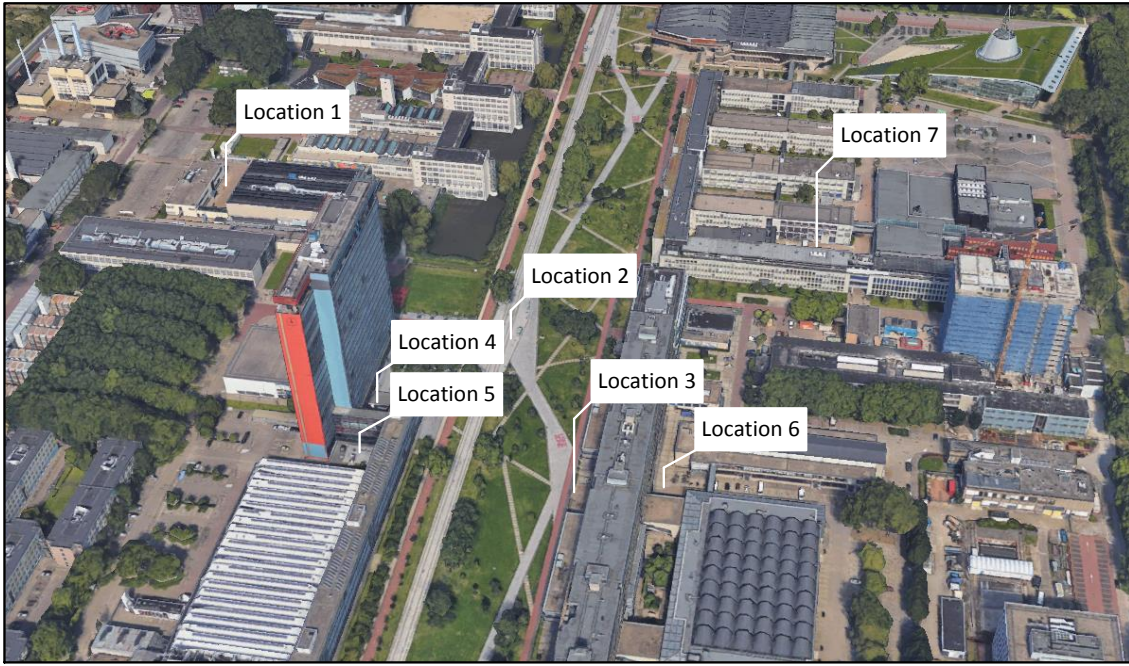


Figure 11– (a) Locations in the TU Delft campus where the hori-catching measurement were done; (b to h) corresponding horizons (on the right) and sky grids (on the left) of measurement locations.

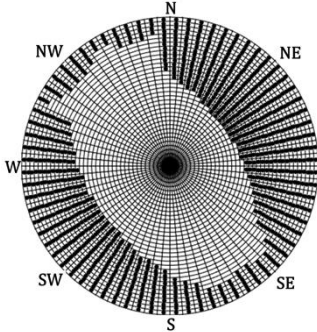


Figure 11 (b), Location no. 1.

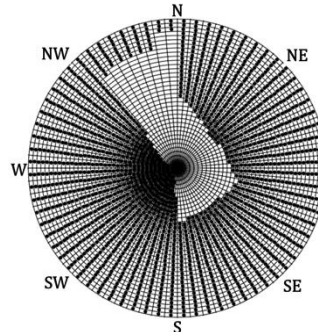
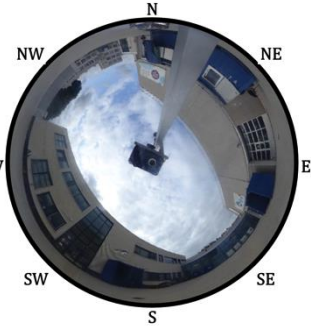


Figure 11 (e), Location no. 4.

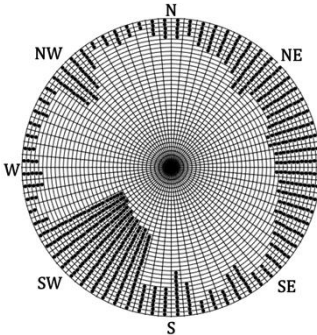
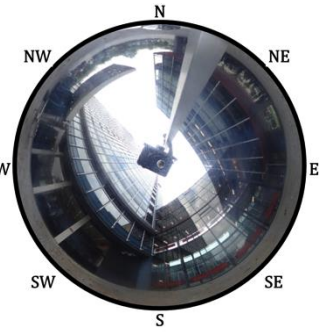


Figure 11 (c), Location no. 2.

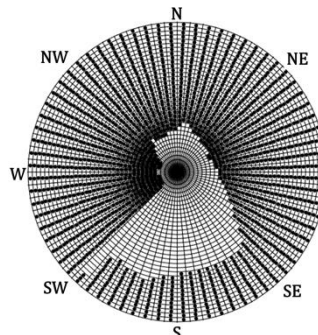
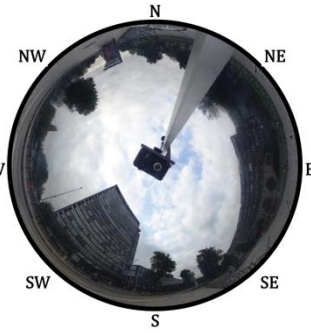


Figure 11 (f), Location no. 5.

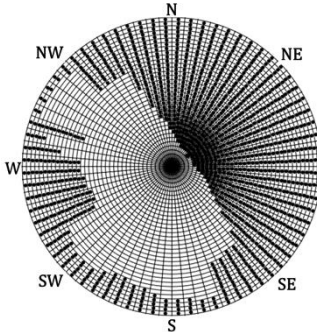
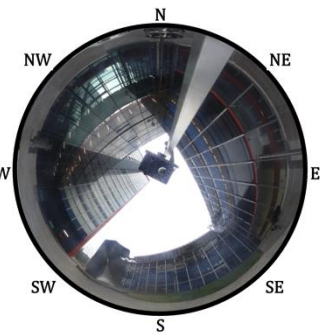


Figure 11 (d), Location no. 3.

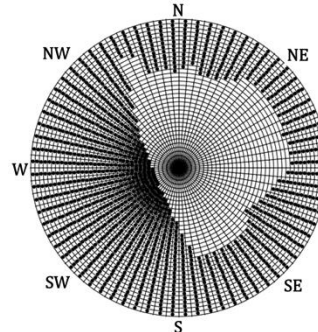
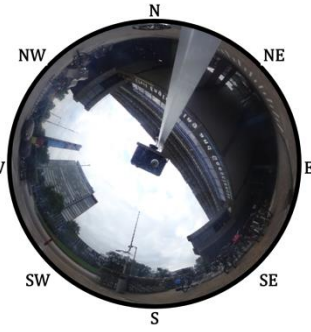
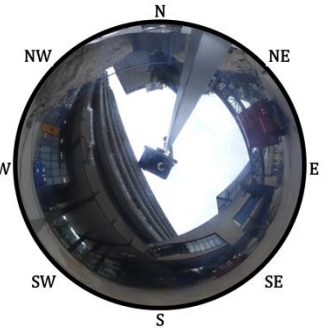


Figure 11 (g), Location no. 6.





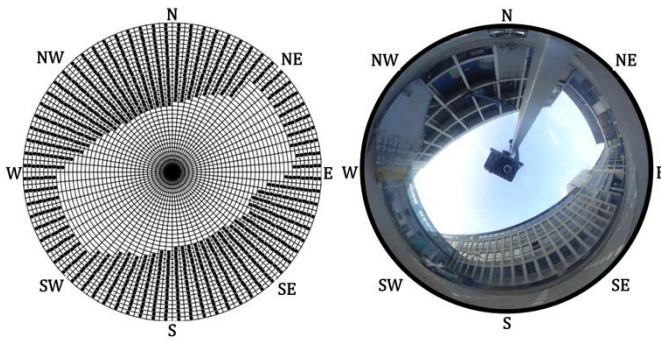


Figure 11 (h), Location no. 7.

#### ACKNOWLEDGMENT

The work was supported by Netherlands Enterprise Agency (RVO) via the project: PVISION with Grant No. TEUE116236.

#### REFERENCES

1. A.F. Mills, *Basic heat and mass transfer*. Vol. 2. 1999: Prentice hall Upper Saddle River.
2. Howell, J.R., M.P. Menguc, and R. Siegel, *Thermal radiation heat transfer*. 2015: CRC press.
3. M.K. Gupta, et al., *Methods for Evaluation of Radiation View Factor: A Review*. Materials Today: Proceedings, 2017. **4**(2): p. 1236-1243.
4. K. Paturski and E. Wolf, *Progress in optics*. 1989: SI.
5. F. Sillion and C. Puech. *A general two-pass method integrating specular and diffuse reflection*. in *ACM SIGGRAPH Computer Graphics*. 1989. ACM.
6. T. L. Bergman, F. P. Incropera, D. P. DeWitt, and A. S. Lavine, *Fundamentals of heat and mass transfer*. John Wiley & Sons, 2011.
7. Klobučar, L., seminar report on thermal radiation heat transfer between surfaces, Department of physics, University of Ljubljana, Feb. 2016.
8. S.B. Bopche and A. Sridharan, *Determination of view factors by contour integral technique*. Annals of Nuclear Energy, 2009. **36**(11-12): p. 1681-1688.
9. M.P.G. Sirimanna, A.T. Perera, and R.A. Attalage. *Numerical estimation of view factors for simple roof enclosures with five surfaces*. in *IASTED International Conference on Engineering and Applied Science (EAS2012)*, Colombo, Sri Lanka. 2012.
10. M. Vujičić, N. Lavery, and S. Brown, *Numerical sensitivity and view factor calculation using the Monte Carlo method*. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2006. **220**(5): p. 697-702.
11. M. Modest, *Radiative Heat Transfer*, Academic, San Diego, CA (2003). Google Scholar.
12. M. Mirhosseini and A. Saboonchi, *View factor calculation using the Monte Carlo method for a 3D strip element to circular cylinder*. International Communications in Heat and Mass Transfer, 2011. **38**(6): p. 821-826.
13. G.V. Baranoski, J.G. Rokne, and G. Xu, *Applying the exponential Chebyshev inequality to the nondeterministic computation of form factors*. Journal of Quantitative Spectroscopy and Radiative Transfer, 2001. **69**(4): p. 447-467.
14. Durrett, R., *Probability: theory and examples*. Vol. 49. 2019: Cambridge university press.
15. P. Vaeeghs, et al. *Random hemisphere method for radiation ray tracing computations*. in *5th European Thermal-Sciences Conference*. 2008.
16. R.M. Eisberg, *Quantum physics of atoms, molecules, solids, nuclei, and*. 1985.
17. G. Paasch, J. Lekner, *Theory of reflection of electromagnetic and particle waves*. Developments in Electromagnetic Theory and Application; 3. Martinus Nijhoff Publishers, Dordrecht/Boston/Lancaster, 1987, 279 S. Dfl. 175.00, US \$78.50, ISBN 90-247-3418-5. Crystal Research and Technology, 1987. **22**(12): p. 1536-1536.
18. A. Gjurchinovski, *Reflection of light from a uniformly moving mirror*. American journal of physics, 2004. **72**(10): p. 1316-1324.
19. M.F. Cohen and D.P. Greenberg. *The hemi-cube: A radiosity solution for complex environments*. in *ACM SIGGRAPH Computer Graphics*. 1985. ACM.
20. J.H. Lambert, *Photometria sive de mensura et gradibus luminis, colorum et umbrae*. 1760: Klett.
21. S. Juds, *Photoelectric sensors and controls: selection and application*. Vol. 63. 1988: CRC Press.
22. E.W. Weisstein, *CRC concise encyclopedia of mathematics*. 2002: Chapman and Hall/CRC.
23. E.B. Saff and A.B. Kuijlaars, *Distributing many points on a sphere*. The mathematical intelligencer, 1997. **19**(1): p. 5-11.
24. Á. González, *Measurement of areas on a sphere using Fibonacci and latitude-longitude lattices*. Mathematical Geosciences, 2010. **42**(1): p. 49.
25. P. Boucher, *Points on a Sphere* 2006 22-July-2018]; Available from: <http://www.softimageblog.com/archives/115>.
26. Y.A. Cengel, et al., *Fundamentals of thermal-fluid sciences*. 2008: McGraw-Hill New York.
27. Google, *Google maps*, . [cited 2018 Online; accessed 15-August-2018].
28. PDOK, *Ahn3 downloads*,. [cited 2018 Online; accessed 15-August-2018].
29. B.T. Chung and P. Sumitra, *Radiation Shape Factors from Planar Point Sources*. Journal of Heat Transfer, 1972. **94**(3): p. 328.
30. D.C. Hamilton and W. Morgan, *Radiant-interchange configuration factors*. 1952.
31. S.S. Ghuman, *Comparison of Single-Core and Multi-Core Processor*. 2016, Advanced research in computer science and software engineering.
32. A. Smets, et al., *Solar Energy: The physics and engineering of photovoltaic conversion, technologies and systems*. UIT Cambridge Limited, 2016.
33. J. Coakley, *Reflectance and albedo, surface*. 2003, Academic. p. 1914-1923.
34. Schregle, R., C. Renken, and S. Wittkopf, *Spatio-Temporal Visualisation of Reflections from Building Integrated Photovoltaics*. Buildings, 2018. **8**(8): p. 101.
35. Oke, T.R., *The energetic basis of the urban heat island*. Quarterly Journal of the Royal Meteorological Society, 1982. **108**(455): p. 1-24.
36. Cheung, H., D. Coles, and G. Levermore, *Urban heat island analysis of Greater Manchester, UK using sky view factor analysis*. Building Services Engineering Research and Technology, 2016. **37**(1): p. 5-17.
37. Isabella, O., et al. *Advanced modelling of E/UIPV systems from location to load*. in *IEEE 7th World Conference on Photovoltaic Energy Conversion*, 2018.
38. *Meteonorm Horicatcher*. [cited 2018 Online accessed: 11-December-2018].

39. D. Steyn, *The calculation of view factors from fisheye-lens photographs: Research note*. 1980.
40. Calabrini, A., et al., A simplified skyline-based method for estimating the annual solar energy potential in urban environments. *Nature Energy* 4, 2019: p. 206-215.