Solar Energy 198 (2020) 623–642

Contents lists available at ScienceDirect

Solar Energy

journal homepage: www.elsevier.com/locate/solener

A review on light transport algorithms and simulation tools to model daylighting inside buildings

Mohammed Ayoub

Architectural Engineering and Environmental Design Department Arab Academy for Science, Technology and Maritime Transport, Abou Keer, P.O. Box 1029, Alexandria, Egypt

ARTICLE INFO

Keywords:
Daylight simulation
Light transport algorithm
Radiosity
Raytracing
Bi-directional algorithms

ABSTRACT

The behavior of daylight inside built settings, though complex, is well understood; and over past years, designers and researchers investigated different tools and methodologies to predict daylighting performance inside buildings. This research presents a comprehensive review on the gradual evolution of light transport algorithms that drove the progress of simulation tools, focusing on the used procedures and algorithms. It draws a broad timeline diagram of light transport algorithms, built upon the literature survey to reveal the knowledge gaps and highlight the future perspectives. Also, this review offers a detailed description of 50 commonly used simulation tools, in terms of the used light transport algorithm, analysis types, accuracy, computation, integration with other programs and licensing. This should help young practitioners of different backgrounds to comprehend these topics, providing a better implementation and integration of current tools as well.

1. Introduction

Daylighting is being regarded by the building design community as an essential concept. Architects and designers significantly benefit from different methodologies to model daylighting in and around buildings to support their decisions during different stages of design (Tregenza and Mardaljevic, 2018). Over past decades, there have been many attempts to predict daylighting, which include diagrams, protractors, calculations, rules-of-thumb and scale models under natural or artificial skies (Aizlewood and Littlefair, 1996; de Boer and Erhorn, 1998; Kenny and Lewis, 1995; McNicholl and Lewis, 1994). Daylight Factor (DF) (Trotter, 1911) is an one of the earliest methods that is still being widely used, as it does not need much computations. It splits daylight into: Sky Component (SC), External Reflected Component (ERC) and Internal Reflected Component (IRC). Frühling (1928) presented *Lumen Method* based on empirical formula to calculate DF, but ignored ERC. Dresler (1954) extended this method by incorporating IRC, but required intensive calculations. Thus, Hopkinson et al. (1954) originated formulas for IRC, based on the *Split-Flux* principle to account for ERC due to ground and external obstructions. Then, Tregenza (1989) modified *split-flux* for vertical obstructions, and Bryan and Clear (1981) developed formulas to obtain SC for overcast and clear skies. Also, Muneer and Angus (1994) presented a method to calculate SC with variable sky luminance distribution. Still, *Split-Flux* showed limits with complex geometries and multiple interreflections, and since it was basically originated for overcast sky, it neglected the changing weather conditions (Littlefair, 1989).

As time went by, daylight modeling tasks increased in both number and sophistication. This led to the emergence of more refined methods, including analytical formulas (Athienitis and Tzempelikos, 2002) machine learning (Kazanasmaz et al., 2009) and physically-based simulations (Ward, 1994). The latter extended earlier conventional practices and revolutionized how daylight modeling is being conducted. However, in their early years, simulation tools were time-consuming and computationally demanding. A previous survey on the use daylight simulation showed the reluctance of many architects to integrate such tools into their practices (Hattrup, 1990). The first decade of the new millennium has witnessed a notable increase in the use of such tools, coupled with further progress of computer processing power (Parkhurst et al., 2006), supported by tools' biased solutions (Crane, 2006) and further simplifications. These advances made simulation tools deliver more accurate results in acceptable timeframes. A later web-based survey revealed the growing interest by practitioners to incorporate daylighting concepts and tools into their work (Reinhart and Fitz, 2006). Successive developments of computer architectures and specialized hardware originated more even more advanced simulation tools (Schardl, 2016).

Daylight simulations provide designers with the ability to compare and optimize design alternatives to promote visual and thermal comforts for the occupants (Nasrollahi and Shokri, 2016), while exploiting

E-mail address: dr.ayoub@aast.edu.

https://doi.org/10.1016/j.solener.2020.02.018
Received 18 December 2019; Received in revised form 23 January 2020; Accepted 4 February 2020

energy-efficient techniques (Amasyali and El-Gohary, 2018). Daylight simulation tools were supported by sophisticated Light Transport Algorithms (LTAs) that offer unprecedented abilities and interactive solutions. Currently, many simulation tools, along with presentation and 3D modeling courses, are commonly being introduced to the students in several schools of Architecture (Reinhart et al., 2006). Overall, the purposes of daylighting studies largely differ in complexity and scale from simple evaluation of internal spaces, complex façade fenestrations, to urban contexts. However, daylight simulations are based on a common set of models, accounting for sky models, building models and daylight models.

1.1. Sky models

The sky celestial hemisphere is modeled based on the received direct sunlight and diffuse daylight (Murdoch, 1985). However, external illuminance values are rarely obtained, compared to the routinely measured global and diffuse irradiances (Alshaibani, 1996) or the monitored sky luminance distribution (Mardaljevic, 2000). Among measurements that are collected by weather stations, solar irradiances are widely-accessible for many locations around the world (Crawley, 2007), as a part of larger meteorological data. This is why the majority of simulation tools depend on luminous efficacy models to convert such irradiances into sky luminance values (Igawa et al., 1999; Muneer and Kinghorn, 1998; Perez et al., 1990; Seo, 2018). Weather datasets come in standardized file formats, such as EnergyPlus Weather (EPW) and DAYSIM Weather Format (WEA), providing typical single-years that represent historical weather conditions over extended periods (Moazami et al., 2019). Typically, EPW includes hourly records of different weather elements, such as temperature, humidity, precipitation, cloud cover, solar irradiances, along with wind velocity and direction (Levermore and Parkinson, 2006). Of all those elements, only two are of concern to daylight modeling: direct normal and diffuse horizontal irradiances.

1.2. Building models

Previous models predict the spatiotemporal settings of external luminous conditions. The geometrical representations of spaces are generated using different approaches of 3D modeling. Computer Aided-Drafting (CAD) is basically a static approach that relies on coordinate-based system to create models. However, editing models using CAD is time-consuming. A more dynamic approach, Building Information Modeling (BIM), integrates building information with the design process. It encapsulates building spatial relationships, geographic information and building components into a single model. It also facilitates the coordination among different team members to make informed decisions. Recently, Algorithmic Design approach generates numerical and geometrical associations among building components, allowing to create and manipulate more complex forms. Apart from such modeling approaches, there is no standard definition of the amount of information daylight models should be supported by. However, for daylight models to be considered well-represented, they should include any surrounding obstructions, exterior ground, façade openings and internal furniture (Ayoub, 2019a; Ng, 2001; Reinhart, 2011; Reinhart and Walkenhorst, 2001). Also, they should involve shading or light redirecting devices, if any (Ashikhmin and Shirley, 2000; Lafortune et al., 1997; Ward, 1992), along with light scattering models, which can be acquired by either analytical or data-driven models (Guarnera et al., 2016; Ulbricht et al., 2006).

1.3. Daylight models

The behavior of daylight in buildings, though complex, is well understood (Tregenza and Mardaljevic, 2018); and many researchers and practitioners depend on *Climate-Based Daylight Modeling (CBDM)* (Mardaljevic, 2000; Reinhart and Herkel, 2000) with *Daylight Coefficient (DC)* method (Tregenza and Waters, 1983). This entails the use of physically-based simulation tools to obtain illuminance values at sensor grids in built environments, considering the fluctuations of the sky's luminance distributions as derived from representative weather data of the study location. *CBDM* depends on façade openings, building geometry, surrounding context and definitions of materials. Still, the building design community was confronted with problems related to modelling the behaviour of light transport through complex fenestration systems (CFS), laser cut panels, light shelves and microprisms (McNeil et al., 2017). To avoid the added computations to account for direct and indirect components, more efficient methods were developed to extend the conventional *CBDM*, such as the 3-, 5-, 4- and 6-Phase methods.

The 3-Phase method (Saxena et al., 2010) splits the raytracing aspect of the DC into 3 phases, each is simulated in a separate matrix without recalculating the others. Daylight (D), Transmission (T) and View (V), in addition to the external Sky matrix (S). (D) matrix relates subdivided sky patches to incident directions on a window. (T) matrix defines the outgoing angular and intensity distribution of reflected and transmitted flux from CFS as a function of incident angle by *Bidirectional Scattering Distribution Function* (BSDF). (V) matrix describes the outgoing directions from the window to the interior space. (S) matrix assigns luminance values to patches representing the sky directions. The resultant illumination can be then attained by multiplying those matrices, for each time step (McNeil et al., 2017). This allows to focus the simulations only on the phase that is being investigated. It was validated against external irradiance, internal illuminances and direct sunlight with shadings (McNeil et al., 2013). The 5-Phase method follows the procedure of the earlier, but adds accuracy to model complex glazing or CFS. It also advances the representation of direct sunlight component (Brembilla and Mardaljevic, 2019) by keeping the interreflected sunlight, and replacing the direct sunlight component with adjustments that can be simulated separately. It was tested against External irradiance and internal illuminance and clear glazing (Lee et al., 2018). The 4-Phase method (Ward, 2015) builds on the 3-PM, but adds a new Facade matrix (F) to consider the luminous flux transfer due to non-coplanar shadings. It was validated against Internal illuminance and luminance field measurements (Wang et al., 2018, 2017, 2016). The 6-Phase method (Geisler-Moroder et al., 2017) is similar to the 5-Phase, but adds the (F) matrix to the procedure. It was tested against Internal illuminance and luminance field measurements (Wang et al., 2018, 2017, 2016). For more details on *CBDM* approaches, the readers are referred to (Ayoub, 2019b).

2. Previous comparison, validation and review studies

Daylighting is a parallel domain to Architecture, and architects have long been advancing their concepts and practices through daylight modeling techniques. Over past years, the related research has seen paralleled improvements, where many studies have focused on studying the capabilities of simulation tools, supported by the progress of computer technology. However, simulation tools must be tested and validated for practical applications (Ward, 1994). Wong and Wong (2017) recognized the importance to validate simulation tools against measurements of built reality, experimental tests or other validated tools. The landscape of previous research efforts to compare and validate daylight simulation tools cannot be fully reviewed here due to the research limits. But for the purpose of this review, it can be briefly overviewed and categorized into three lines of research: comparisons, validations and reviews.

2.1. Comparison studies

The first line of research entails performing comparative experiments on two or more tools, along with their supporting LTAs. Among these attempts is a study conducted by Tsangrassoulis et al. (1996), who developed a model to predict daylighting performance and validated it against experimental data and simulation results, attained from *radiosity* and *raytracing* methods. In another study, Tsangrassoulis and Bourdakis (2003) compared daylight levels in an atria under different conditions using *radiosity* and *raytracing* methods. Ibarra and Reinhart (2009) evaluated the daylight predictions of a classroom using Ecotect's *split-flux* and Radiance's *raytracing*. Versage et al. (2010) examined the internal illuminances due to North and South facing openings by EnergyPlus' *split-flux* and DAYSIM's *raytracing*. Yoon et al. (2014a, 2014b) compared the internal illuminances due to openings of different orientations via EnergyPlus's DElight integrated *radiosity* and Detailed's *split-flux*. Those attempts, in addition to many others, show how models are applied to different situations encountered by building designers and the type of data they work with. Comparing simulation tools might seem a simple task, but in practice, the results are significantly affected by many factors, such as the selected sky models that differ from the actual condition (Kim and Chung, 2011), and the uncertainties that stem from continuously changing sky luminance distribution (Tregenza, 2017). Another issue is the estimation of optical properties of materials, including surface reflectance, window transmittance (Ng et al., 2001), furniture reflectance (Li et al., 2004) and the impact of the nearby context (Mardaljevic, 2004). Still, the range of expected errors in the results are still difficult to be estimated (Maamari et al., 2006). Thus, LTA must reflect user requirements, although no single tool can meet all designers' needs (Davoodi et al., 2014).

2.2. Validation studies

The second line of research is about validating daylight simulation tools and their LTAs against built realities, controlled laboratory conditions and standardized datasets. Reproducing built realities via scale models will likely introduce uncertainties that are associated with measurement of geometries and photometric properties of materials (Maamari and Fontoynont, 2003). Also, evaluating simulation tools under laboratory settings may ignore normal building conditions (Galasiu and Atif, 2002). To overcome such inconsistencies, standardized comparison methods were proposed, such as Cornell Box (Goral et al., 1984) that was commonly used to visually determine the accuracy of early rendering tools (Ulbricht et al., 2006). The British Building Research Establishment (BRE) created BRE-IDMP validation dataset, which comprises sky luminance measured at 145 patches, besides direct normal, internal and vertical illuminance values for different glazing (Mardaljevic, 2001). To the author's best knowledge, Mardaljevic was the only researcher that ever used the BRE-IDMB dataset to validate Radiance (Mardaljevic, 2000, 1997, 1995). Also, the Commission Internationale de l'Éclairage (CIE) proposed a variety of test cases of different conditions to validate the accuracy of simulation tools (CIE, 2006). Among the tested tools against CIE test cases are Radiance (Donn et al., 2007; Geisler-Moroder and Dür, 2008; Osborne, 2012), 3ds Max (Kensek and Suk, 2011; Osborne, 2012), DIALux (Mangkuto, 2016), LightScape (Maamari et al., 2006), Relux (Maamari et al., 2006), VELUX Daylight Visualizer (Labayrade et al., 2010), AGi32 and Ecotect (Kensek and Suk, 2011). The National Research Council in Canada created another set of test cases that includes the measured indoor and outdoor illuminances, with direct and diffuse irradiances. They were used to validate 3ds Max and DAYSIM (Reinhart and Pierre-Felix, 2009). Apart from such test cases, Radiance and its modified version, DAYSIM, were validated against measurements of real room and physical model (Jarvis and Donn, 1997), clear glazing and shadings (Reinhart and Walkenhorst, 2001), measurements of goniophotometer (Schregle and Wienold, 2004), internal illuminance (Reinhart et al., 2006), translucent panels (Reinhart and Andersen, 2006) and 3-Phase Method to model complex fenestration systems (McNeil and Lee, 2012). Noback et al. (2016) also compared simulations with the Radiance *Photon Map* against actual measurements. They also discussed the problems arising with such experimental approaches.

2.3. Review studies

The third line of research involves conducting review studies on the simulation tools and their LTAs. Carroll (1999) presented an early review on the abilities of simulation tools, such as the used algorithms, accuracy and ease of use. However, the study did not include detailed comparisons, let alone many of the reviewed tools are now obsolete, compared to today's tools that require much higher computational power. Roy (2000) performed a review on the history of lighting calculation and their tools, comparing Adeline, LightScape, Microstation and RadioRay. Based on this comparison, there was no ideal tool could be concluded, since they have their advantages and disadvantages. Bryan and Autif (2002) compared the commonly used simulation tools, including Radiance, LightScape, LumenMicro, and FormZ RadioZity in terms of procedure, viewpoints, memory, numeric output, realistic rendering, animation, glare analysis, and ease of use. It was concluded that simulation tools should offer user-friendly interface, accurate results, and be capable of producing photo-realistically rendering. Ulbricht et al. (2006) identified different approaches to verify photo-realistic simulation tools, focusing on the evaluation of LTA by comparing the rendered image to real scene measurements and analytic solutions. The review summarized the existing verification methods, classified by the used device, such as visual comparisons, radiometers, incident color meters, and scanners. Kota and Haberl (2009) reviewed the development of daylight calculation methods, algorithms and the supporting tools. The study outlined the attempts to integrate thermal simulation with daylighting. However, many tools showed limitations in daylighting calculation, implying the need for additional development in the used LTAs.

Interestingly, Ochoa et al. (2012, 2010) summarized LTAs, validations, tools' inputs, outputs, and the integration with the design process. Different aspects of daylight simulation tools were also discussed such as accuracy, number of parameters to consider, and computational effort and time. Davoodi et al. (2014) also evaluated simulation tools, including DIALux, VELUX Daylight Visualizer, Radiance, DAYSIM, and Radiance's Evalglare. It was found that no single tool can meet all designers' needs, summarizing the strength and limits of such tools. Oh and Haberl (2016a, 2016b, 2016c) presented significant reviews on the analysis methods of building design, considering energy simulation, solar energy analysis, daylighting studies and LTAs. Such reviews explained the analysis methods of thermal envelope loads, solar photovoltaic, solar thermal, passive solar and daylight simulation tools. Then, Danks et al. (2016) discussed the regulations and metrics related to studying the impacts of visible light and thermal energy in buildings. They developed quantitative criteria for design and construction, which can help architects realizing the impact of reflections buildings have on the surrounding context. Jakica (2018) conducted an interdisciplinary review on simulation tools, analyzing their used algorithms, accuracy, speed and integration with the design process. Recently, Ayoub (2019b) presented a broad review on the growing fundamental directions to predict the amount of daylight inside buildings, putting the focus on tracing sky models, weather datasets, building geometry and daylight calculation methods. The review included a detailed description of prevalent simulation tools, focusing on their evaluation in terms of usability, interoperability, exchangeability and simulation capabilities.

A major review is conducted almost every 2–3 years on research efforts and methodologies to model daylighting in buildings using different simulation tools. This trend is not only expected to continue, but will also accelerate, revealing newly developed research trends. This is particularly important, as such reviews provide an overall picture of the related topics and draw new paths for more focused future studies, supporting professionals in their practices. In general, the majority of those reviews highlighted the relevant procedures and algorithms. Despite their significance, most of them are missing the explicit consideration of the essential principles of LTAs, such as their aspect, accuracy and integration with simulation tools, from the architects' viewpoint. Not to mention that the landscape of scholarly literature that focuses on employing different LTAs to model daylighting inside buildings continues to grow. Such accumulating knowledge may hinder architects from integrating those techniques into their practices due to the lack of essential knowledge. To address this gap, a review is found imperative to trace the ever-growing developments of LTAs that are constantly driving the progress of daylight simulation tools.

3. Research aim, scope and methodology

This review aims to help architects of different expertise and backgrounds understanding the essential concepts of LTAs that are used to model internal daylighting conditions. Building upon the previous reviews, this work not only highlights early significant research that led to the development of notable physically-based LTAs, but also targets the scholarly literature that was published after 2011. Moreover, it presents a detailed description of 50 common simulation tools, in terms of the used LTAs, analysis types, accuracy, computation, integration with other programs and licensing. Finally, a broad timeline is drawn according to the reviewed literature to reveal recent trends and future perspectives in the related research domain. This should clear the ambiguity of unfamiliar terms and technicalities, leading to a better implementation of current daylight simulation tools as well. This review is conducted with the architects' nature in mind; thus, it only considers significant breakthroughs that are associated with the development of physically-accurate simulation tools and their LTAs to model daylighting in buildings. The domain of computer graphics largely extends beyond such borders. Thus, this work will not reflect the attempts to develop LTAs for photorealistic representations or artistic renderings that suit other fields of applications, such as animation, gaming and movie-making. They are out of scope, and cannot fit the extent of the review's specified target.

The methodological procedure is realized through several steps (Fig. 1). It starts by investigating the related scholarly literature against the targeted scope. A keyword search is conducted on databases of journal and conference proceedings using Google Scholar and Egyptian Knowledge Bank, facilitating broader searches across many sources. The resulting scholarly literature is then filtered to recognize the aim of each study, identifying the relevant candidate articles within the scope. The selection process also occurs within each article, where articles that were cited by such refined literature are screened as well. Those that passed the scope criteria are then identified as candidate articles. Later, all the selected articles are categorized in accordance with the topical structure of the entire review, as follows: Section 1 provides a brief introduction on the methodologies to predict daylighting performance inside buildings, including sky models, building models and daylight models. Section 2 outlines the related studies that focused on comparing, validating and reviewing different studies that considered the use of simulation tools and their LTAs. Section 3 defines the aim, scope and the methodological procedure to conduct this review. Section 4 offers an overview on the family of LTAs, including *Finite Element-Based Radiosity*, *Instant Radiosity*, *Monte-Carlo-Based Raytracing*, *Bi-Directional Path Tracing*, *Metropolis Light Transport* and *Photon Mapping*, preceded by a brief description of reflection and transmission models, in addition to the rendering equation. Section 5 reviews the applications of different hardware architectures to reveal the achieved breakthroughs that benefited from the growing computational power. Section 6 offers a description of 50 widely-used daylight simulation tools, in terms of the criteria mentioned above. Finally, conclusions of the review findings are discussed in Sections 7 and 8.

4. Light transport algorithms

4.1. Reflection and transmission models

LTAs are basically derived by the progress of computer graphics, animation, gaming and movie-making. They model the propagation of light in spaces and its interactions, such as absorption, reflection and refraction, based on surfaces' properties (Ashikhmin and Shirley, 2000; Glassner, 1995; Phong, 1975; Ward, 1992). These properties mathematically describe how the light is scattered by a surface through the *Bidirectional Scattering Distribution Function (BSDF)* (Asmail, 1991). In practice, *BSDF* is split into the reflected and transmitted components that are defined by the *Bidirectional Reflectance Distribution Functions (BRDF)* for the reflected scatter off surfaces (Nicodemus, 1965), and the *Bidirectional Transmittance Distribution Function (BTDF)* for the transmitted scatter through surfaces (Heckbert, 1991), respectively. A vital issue is related to how to represent those functions in LTAs. A *BSDF* matrix contains structured material definitions in eXtensible Markup Language (XML) file format. This dataset isn't typically generated by simulation tools, but instead either by specialized tools, such as TracePro, Radiance's genBSDF and Lawrence Berkeley National Laboratory's Window (Mitchell et al., 2006), or by measured data of CFS using goniophotometers or spectrophotometers (Guarnera et al., 2016).

Besides direct lights, such interactions were thought to constitute most of the total illumination, but a space's ambient illumination comes from object-to-object interreflections as well (Reinhart, 2018). Thus, internal surfaces should be treated as potential light sources, especially when they cause significant specular reflections. These concepts are vital in daylighting studies, since they explain the behaviour of complex light transport paths in built settings. For example, the light scattering properties of CFS are often irregular in a way that deflect light. Thus, they must be accounted for by LTAs, which is challenging task. Their value stems from the ability to indirectly illuminate deep internal spaces, where direct daylight cannot reach. This results in spatio-temporally localized highlights (caustics) that may be perceived as glare; which are computationally expensive to simulate. Some LTAs are more efficient at this than others in terms of bias and noise (Whittle et al., 2017), as will be discussed next. The initial idea of *Global Illumination (GI)* led to recognizing various optical effects. LTAs receive a 3D model of a given space, along with its material properties. Then, the behavior of light is modeled to acquire integral quantities, such as illuminance and irradiance values, and directionally resolved luminance distributions (IES, 2012; Van Den Wymelenberg and Inanici, 2016,

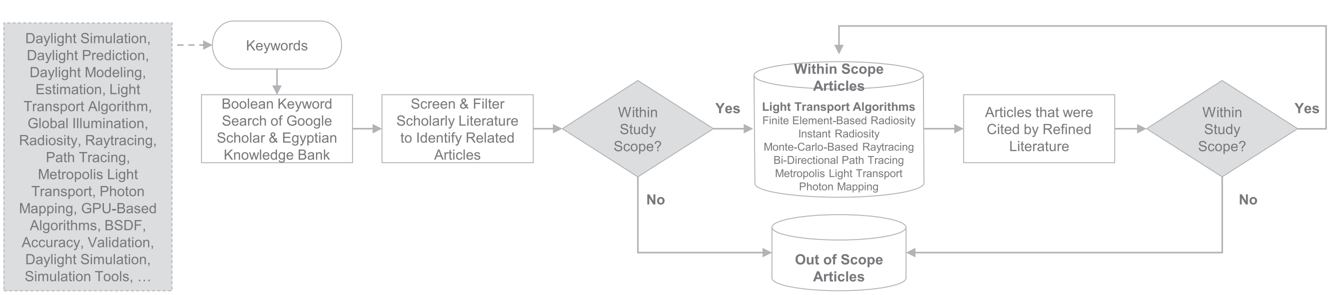

Fig. 1. Steps of selecting and categorizing scholarly literature related to the scope of the research.

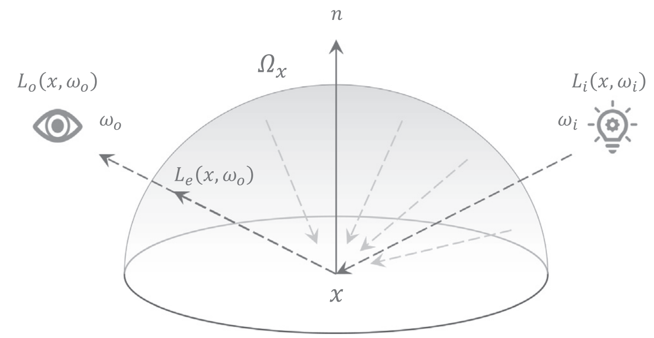

Fig. 2. The geometry of the rendering equation.

2014). Thus, precise light scattering models are prerequisite to accurate simulations. Physically, LTAs aim at approximating the light transport integral equation, called the *rendering equation*, at any given point in a space (Kajiya, 1986) based on the law of conservation of energy, representing three principles of optics: global illumination, equivalence and direction. The *rendering equation* (Fig. 2) states that at a given point (x) on a surface, the intensity of outward light (L_o), directed towards the viewer along solid angle (ω_o) $\in \Omega_x$, is the sum of the light emitted (L_e) by and reflected (L_r) off that surface, as expressed in Eq. (1):

$$L_o(x,\omega_o) = L_e(x,\omega_o) + L_r(x,\omega_o) \tag{1}$$

The light reflected (L_r) itself corresponds to an integral over a unit hemisphere of *BRDF* from (ω_i) to (ω_o) of the incoming light (L_i) from all possible directions (ω_i), multiplied by the chances of light to bounce towards the viewer and by the irradiance factor of outward light over the surface normal to that point, defined by Eq. (2):

$$L_o(x,\omega_o) = L_e(x,\omega_o) + \int_{\Omega_x} f_r(x,\omega_i \to \omega_o) L_i(x,\omega_i)(\omega_i \bullet n)\, d\omega_i \tag{2}$$

Still, this equation does not state how to solve light transport problem, as it simply sets the theoretical basis from which various LTAs are derived. Having infinite high-dimension integrals over discontinuous functions, the rendering equation is infinitely recursive without closed form solution. This makes solving it a challenging task, except in cases that are too simple to be of practical importance, where analytical solutions can be derived by a series of expansions (Kollig and Keller, 2000). In practice, the equation is too complex to solve analytically;

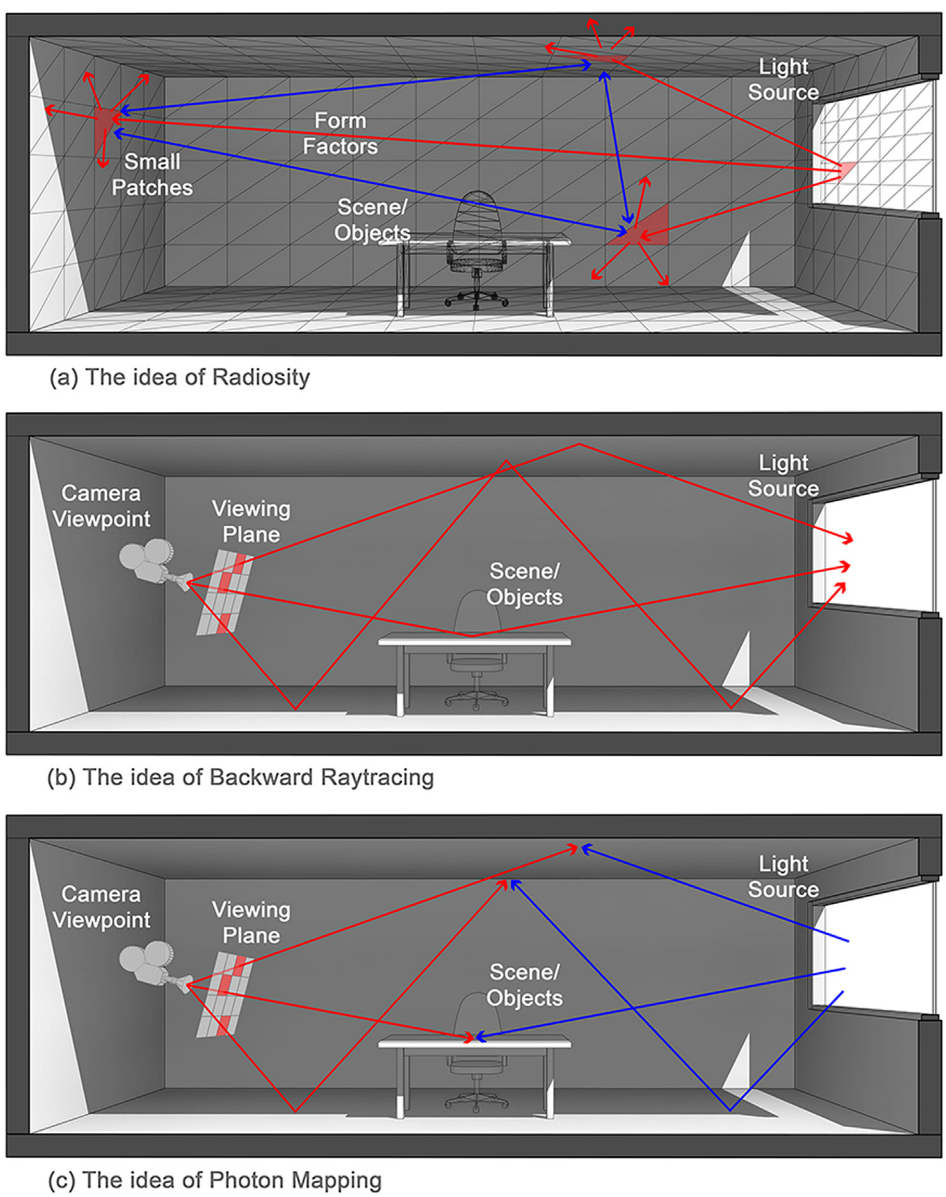

Fig. 3. The basic concepts behind three widely-used light transport algorithms: (a) Radiosity, (b) Backward Raytracing (c) and Photon Mapping.

thus, LTAs are developed as means to approximately solve it, irrespective of the computational effort, since it is scalable in terms of iterations, number of traced rays, etc., depending on the desired accuracy. As such, relatively excessive computation can be avoided, still with erroneous solutions. One of the biggest problems was the limited computational resources due to technological limitations during the early years of LTAs. As computers become more capable and less expensive, more demanding algorithms are created. LTAs should handle complex geometries and different material definitions within acceptable timeframes (Veach and Guibas, 1997). Some algorithms that ignore certain optical effects, while seemingly fast, still exhibit errors. Statistically, an error is the difference between the estimated result and the actual value, described as systematic or random error. In simulation, there is a common confusion that is always associated with the description of LTAs' behavior and function of errors in terms of bias and noise (Arvo et al., 1994). Systematic error (bias) is not determined by chance, and it is attributed to inaccuracy; yet, it cannot be eliminated by simply averaging a larger sample size (Veach and Guibas, 1997). In contrast, random error (noise) is defined by the imprecision of the solutions due to randomness, and represented as random variations among samples (Hachisuka, 2013). Still, it can be eliminated by generating more samples (Peddie, 2019).

Apart from the errors of LTAs, the purpose of conducting lighting simulations can be categorized into photorealistic representations and physically-based visualizations (Ochoa et al., 2012). The computer graphics and game-developer communities have been focusing on exploring photorealism to produce realistic and artistic renderings. However, designers cannot rely on representations as a sole approach of decision-making. The building design community has been oriented towards physically-accurate photometric simulations, which is the focus of this review, to acquire quantitative estimations under different conditions during the different stages of design. This approach is particularly vital to daylighting studies, where the abilities of LTAs heavily influence the accuracy of building performance simulation. Form the core of the rendering equation, there has been other classifications of the approaches LTAs are taking to solve light transport problems (Lesev, 2010; Ochoa et al., 2012), according to the type of scene-based rendering; view dependency (view-dependent and view-independent); and accuracy (biased and unbiased). The next subsections will review different LTAs (Fig. 3) as per view dependency, including the view-independent *Finite Element-based Radiosity* and *Instant Radiosity*, in addition to other view-dependent algorithms, such as *Monte-Carlo-based Raytracing*, *Bi-Directional Path Tracing*, *Metropolis Light Transport* and *Photon Mapping*.

4.2. View-independent algorithms

4.2.1. Finite element-based radiosity

As previously explained, light interactions with surfaces are governed by material properties. Complex materials yield both specular and diffuse behaviors, but the latter dominates most building materials (Reinhart, 2018); and their lower frequencies vary slowly among surfaces (Vardis, 2016). Such factors motivated the designers to adopt *Finite Element Method (FEM)*, or *Radiosity*, solving the problems of light transport in built environments. *Radiosity* relates the received illuminance by surfaces to uniform reflected luminance, forcing all surfaces to be Lambertian; ideal diffuse reflectors of constant luminances. Thus, all internal surfaces are divided into finite elements, or small patches, as additional illumination sources, received directly by light sources and indirectly by reflection from patches (Watt, 2000). In this sense, *radiosity* performs calculations for all patches in the space, not only the visible ones. While this is not the case for many other LTAs, this is particularly useful, since different viewpoints do not require additional computations, making *radiosity* a view-independent algorithm. For added accuracy, surfaces of high luminances compared to their neighbors are detected and divided into smaller patches. For each pair of patches, a form factor that denotes the amount of light transport interactions between patches is calculated and stored in the memory with the directional information of incoming and outgoing light. Then, these form factors are used to solve the *radiosity* linear system of rendering equations. The mathematical forms of light interreflections with Lambertian surfaces by *radiative flux transfer* are credited to Yamauti (1926), who presented finite differences to solve discrete forms of these equations. Although theoretical, these notions were used in practice by Higbie (1934), who suggested techniques to calculate reflected luminances off surfaces, which were then utilized by Moon and Spencer (1948) to create the first radiosity-based rendering (Ashdown, 1996). O'Brien (1955) developed the *radiosity* algorithm via network approach and *FEM*. While relied on hand calculations then, *radiosity* was ahead of its time, and it took almost 25 years until it was adopted by the building design community. However, *radiosity* ideas have been exploited to analyze *radiative heat transfer* between surfaces, based on the theory of energy balance (Goral et al., 1984).

Modest (1982) exploited the original *Radiosity* concept to create an algorithm for daylight calculation in buildings. Goral et al. (1984) formulated *radiosity* for accurate renderings due to natural or artificial light, yielding the famous *Cornell Box* (Goral et al., 1984), which became commonly used test of the accuracy of rendering tools. Cohen et al. (1988) also developed *Progressive Radiosity* as a computationally-efficient algorithm based on the approach of progressive refinement. At that moment, those advances were designated for light transport in ideally diffuse spaces, ignoring the effect of complex materials. Thus, research efforts advanced *radiosity* to consider more capabilities, such as non-diffuse effects and directions (Immel et al., 1986), few specular reflections and translucent surfaces (Wallace et al., 1987), multiple specular reflections, refractions and planar mirrored surfaces (Sillion and Puech, 1989), along with light interreflections with diffuse and non-diffuse surfaces (Rushmeier and Torrance, 1990). Since view-independent interreflections require smaller patches and heavy computation, hierarchical methods were formulated, included discontinuities and refined patches for efficient calculations. Hanrahan et al. (1991) suggested *Hierarchical Radiosity* that subdivides large polygonal patches into smaller ones to reduce the computational effort. Lischinski et al. (1992) integrated diffuse situations by discontinuities as boundaries among patches. Then, they combined *hierarchical radiosity* and discontinuity for more accurate and efficient results (Lischinski et al., 1993). Gortler et al. (1993) formulated *Wavelet Radiosity* as a hierarchical method, based on theory of wavelets (Yserentant, 1986). While Smits et al. (1994) suggested accelerating *hierarchical radiosity* by volumetric clustering of objects, Garland et al. (2001) extended this approach by incorporating surface regions. Tregenza (1994) also proposed a recursive subdivision of surfaces into triangular patches to consider hidden surfaces and specular reflections.

4.2.2. Instant radiosity and other hybrid methods

A diversification of the original algorithm is two-pass *Instant Radiosity* (Keller, 1997) that received much recognition due to its ability to handle difficult situations. Unlike *patch-based radiosity*, *pixel-based* methods can be parallelized, where each pixel is processed independently. *Instant Radiosity* combines *radiosity* with *raytracing* for accelerated calculations; but it only addresses diffuse or near-diffuse *BRDF* materials. The first pass traces light paths, while virtual point lights are generated and deposited at the vertices of such paths. Few point lights are used to approximate indirect illumination in the second pass, making this method biased; but it can be adapted to increase accuracy. Segovia et al. (2006) developed *Bi-Directional Instant Radiosity* as an extension to *instant radiosity*, among many other algorithms that are generated and combined to find virtual point lights relevant to the virtual plane of the viewer. Using Graphics Processing Unit (GPU), this method provides fast interactive frame rates for complex materials and many light sources (Ritschel et al., 2012). In other paper, Segovia et al. (2007) presented *Metropolis Instant Radiosity* to represent incoming radiances as sets of virtual point lights; and their responses are calculated and sampled for efficient renderings using GPUs. Another variation is the *Incremental Instant Radiosity*, providing real-time renderings using GPUs as well (Laine et al., 2007). *Multidimensional Lightcuts* as an additional hybrid method was developed by Walter et al. (2006) as an extension to *Lightcuts* (Walter et al., 2005), directly derived from *Instant Radiosity*. Those methods can produce comparable results to unbiased algorithms in less time (Dachsbacher et al., 2014). New extensions were developed to reduce bias and noise using GPUs, supporting wider ranges of materials, such as *Bi-Directional Lightcuts* (Walter et al., 2012) and *Progressive Lightcuts* (Davidovič et al., 2012). Yet, they are still considered biased, showing limitations in certain situations, such as the behaviour of CFS.

4.3. View-dependent algorithms

4.3.1. Monte-Carlo-based raytracing

Light sources emit light rays through internal space until they are occluded by surfaces, whose optical properties of absorption, reflection, refraction or fluorescence define the light propagation. The reflected or refracted rays bounce off other surfaces before reaching the eye. Rather than *radiosity's* object-to-object interactions, *Raytracing* is founded to imitate the natural behavior of light, based on tracing light rays from camera viewpoint through viewing plane of many pixels to the receiving surfaces (Kuchkuda, 1988). The physical basis for such notion is that the radiance remains invariant along the unoccluded ray. The algorithm triggers different light scattering models that simulate the interaction of light with surfaces of different optical properties (Kota and Haberl, 2009; Reinhart, 2018), approximating the resulting illuminance and color of each pixel. Rays continue to bounce off the space until reaching light sources, or achieving termination criteria. The tracing is then ended, and pixels' information is updated. In practice, these are sampled separately with directed rays.

The idea of *raytracing* was pioneered by Appel (1968), with Goldstein and Nagel (1971), who solved hidden surfaces issues, while proposing mathematical representations for the production of renderings. In this early algorithm, called *Ray Casting*, rays are cast from the eye into the space through the pixels of a virtual plane, one ray per pixel. It solved the light intersection with different geometries, defining their visibility based on the materials without recursively recasting additional rays, yet limiting the ability to handle complex materials. Thus, Whitted (1980) extended *ray casting* to include specular reflection and refraction. In *Recursive Raytracing*, rays are cast recursively to determine the occluding surfaces, generating new ray types, such as reflected or refracted, depending on surfaces' materials. Reflected rays travel in mirrored directions, while refracted rays penetrate surfaces, bending their directions. Still, the required computational effort limits the ability to determine ray directions. Thus, Cook et al. (1984) proposed *Distributed Raytracing*, which incorporates analytic functions of to distribute multiple rays in various domains, including fuzzy conditions to feasibly solve problems of prior methods, in addition to motion blur, depth of field, and most importantly, translucency and glossy reflections. Due to the exertion to focus on the rays that only contribute to the results, those progresses still rely on the computationally expensive brute force to intersect each ray with each surface. Therefore, many researchers suggested simplifications by enclosing objects with bounding volumes before ray-surface intersections (Arvo and Kirk, 1987; Kay and Kajiya, 1986; Weghorst and Greenberg, 1984).

It may be surprising to backwardly trace light from the viewer to surfaces, instead of forwardly following natural ray directions. But *Backward Raytracing* or simply *raytracing*, a term first used by Arvo (1986), exhibits more efficiency in many cases, except for specular reflections and CFS. Not all the emitted rays reach the viewpoint, even indirectly, where reflected and refracted rays can get complicated, as they travel in unpredicted directions, yielding redundant computation of unseen rays (Kuchkuda, 1988). This is why *raytracing* shortcuts the tracing process by starting with the eye. Still, many optical systems benefit from *forward raytracing*, such as light transport in lenses, CFS and caustics (Jakica, 2018). Although *raytracing* involves efficient computation of specular reflections, it requires generating many rays that overwhelm the calculations to handle diffuse interreflections, unlike *radiosity*. Thus, early *raytracing* methods use a constant ambient term to avoid this complication. Arvo's *backward raytracing* provided a solution for diffuse reflection that has not been tackled before. Likewise, Ward et al. (1988) and Ward and Rubinstein (1988) suggested efficient calculations for diffuse reflection by replacing the ambient term without influencing specular reflections. In classic caching, irradiance is not computed for every pixel, but rather for some pixels according to a criterion based on the average distance to other surfaces (Ritschel et al., 2012). The *Ambient Cache* was one of the breakthroughs in Radiance that is based on the observation that the computationally expensive indirect-diffuse scattering is characterized by low frequency and view-independent, thus cacheable. The efficient implementation of the *ambient cache* allowed to re-use most of the raytracing results in image generation.

Kajiya (1986) pioneered the rendering equation that is vital for many algorithms. He also introduced the idea of *Path Tracing* by distributing paths stochastically from the eye through the viewing plane using *Monte-Carlo* methods (Dutre et al., 1993; Spanier and Gelbard, 1969), which became fundamental constituent for many subsequent algorithms. Unlike *raytracing*, *path tracing* does not just trace rays to light sources at each surface intersection point. It rather recursively emits and traces rays that continue to bounce the space until hitting light sources, reaching bounce limit, or getting absorbed by a surface. The information of color and illuminance of each pixel is collected during every contributing ray-surface interaction. This way, the algorithm is inefficiently used for direct illumination, causing variance in the form of noisy results; owing to its probabilistic nature (Dutre et al., 2006). Also, *path tracing* relies on surface's *BRDF* for directional sampling of specular reflections; but for diffuse situations, the rendering is guided by incoming radiances. In practice, this radiance is not always available; thus, assumptions or rigorous pre-samplings must be made. Sampling incoming radiance distribution is effective for diffuse reflections, where many techniques combine samples from distributions for variance reduction of *Monte-Carlo* methods, such as importance sampling, caching and interpolation. Further developments that were added the algorithm, considering its working technique, and the use of dedicated hardware for real-time and interactive rendering, will be reviewed in the following section. In addition, for daylight simulations, more recent improvements in raytracing has been applied on CBDM, as reflected by the reviewed standards introducing climate-based performance metrics (Brembilla et al., 2019; Geisler-Moroder et al., 2017; McNeil and Lee, 2012; Subramaniam, 2018; Wang et al., 2018).

4.3.2. Bi-directional path tracing

Apart from prior uni-directional methods, a two-pass algorithm that combines forward and backward path tracing, known as *Bi-Directional Path Tracing (BDPT)*, was developed by Lafortune and Willems (1993) along with Veach and Guibas (1994). *BDPT* traces rays from a light source and from a viewpoint at the same time, where both bounce the space for certain steps. Then, those rays are combined by another deterministic ray; and the contribution at each respective pixel is obtained. *BDPT* may seem slower and more expensive per iteration, but it outperforms the uni-directional methods by ensuring that light will reach the eye from the added tracing. Moreover, its larger number of sampling techniques provides higher accuracy, especially in situations that involve indirect and inhomogeneous illumination, such as reflective shading devices, redirecting systems and CFS, balancing the added computation. Veach and Guibas (1995) developed *Multiple Importance Sampling (MIS)* for effective *BDPT* in difficult spaces, where new sampling paths are traced, based on successful tracing of light from the sources to the camera. Similarly, Kollig and Keller (2000) modified

BDPT by implementing the deterministic *quasi-Monte-Carlo* method using low-discrepancy sequence, known as *quasi-random sequence*, which is more efficient than other *BDPT* algorithms due to its faster convergence of sampling space with less clustering. Still, this comes at the cost of low periods (visible patterns), compared to pseudorandom generators. The quality of *Monte-Carlo raytracing* relies greatly on the quality of its random number generator. Thus, *quasi-Monte-Carlo* attracted much attention, although it is still not considered as a mainstream by the research community.

Both Hachisuka et al. (2012) and Georgiev et al. (2012) integrated *Vertex Connection and Merging* method into *BDPT* and *Photon Mapping* (Jensen, 2001, 1997, 1996, 1995) to handle wider ranges of lights, allowing to develop other methods that use both algorithms to trace and combine paths using *MIS*. Kaplanyan and Dachsbacher (2013) regulated light-carrying paths to improve the unbiased *BDPT* by introducing effective bias in difficult situations. Vorba et al. (2014) addressed previous issue by effectively sampling radiance distribution in difficult spaces during the rendering instead of the tracing pass. Jones and Reinhart (2016) introduced *Progressive Path Tracing* for real-time prediction of glare, and contrast with rendered images. A diversification of *BDPT* was incorporated into *IRay's* unbiased render engine (Keller et al., 2017), which was integrated in *Nvidia's Mental Ray* since 2010 (Reinhart and Breton, 2009). *Arnold* renderer (Georgiev et al., 2018) is an unbiased engine that uses an optimized variation of *BDPT*. It was originated by Solid Angle in 1997, then was acquired by Autodesk in 2016.

4.3.3. Metropolis light transport

BDPT can handle caustics well, yet it generates additional computationally excessive rays per pixel to avoid noisy results. In situations where many rays occlude surfaces of low reflectance, they do not contribute accurately to the results. Thus, Veach and Guibas (1997) tackled noise errors by developing *Metropolis Light Transport (MLT)* for efficient performance using fewer samples, based on a variation of the *Monte-Carlo method*, called *Metropolis Sampling*. It obtains samples from radiance distribution, in case direct sampling is complicated, specifically, when most of significant lighting contributions are indirect, and follow convoluted paths, such as Veach's famous "light source through a jar door" scene. *MLT* is a powerful LTA, but also is difficult to implement as well, since it introduces significant bias that must be adequately compensated. *MLT* stores the rays from light source to viewpoint and traces the mutating ones in difficult spaces. *MLT* was integrated into *IRay* and research-oriented *Mitsuba* renderer (Mitsuba, 2010). *MLT* has seen further progresses, including *Primary Sample Space Metropolis Light Transport* (Kelemen et al., 2002), *Energy Redistribution Path Tracing* (Cline et al., 2005), *Manifold Exploration* (Jakob and Marschner, 2012), *Gradient-Domain Metropolis Light Transport* (Lehtinen et al., 2013) *Multiplexed Metropolis Light Transport* (Hachisuka et al., 2014) and *Matrix Bi-Directional Path Tracing* (Chaitanya et al., 2018).

4.3.4. Photon mapping

Photon Mapping is an efficient alternative to *Monte-Carlo* methods. It is a versatile algorithm, originated by Jensen (1995), who added further refinements to the algorithm (2001, 1997, 1996). Unlike *radiosity*, *photon mapping* decouples the geometrical representation of spaces from their light contributions without patches, and stores it in a spatial data structure, called photon map. This allows solving the terms of the rendering equation separately using other methods, making this algorithm flexible. *Photon mapping* is a two-pass method. In the first pass, a set of photons is emitted forwardly into a space from each light source, where the emission model is based on the source type and intensity. Instead of generating new photons at photon-surface intersections, the behavior of photons, whether absorbed, reflected, or refracted, is determined probabilistically based on the surface optical properties, using an unbiased *Monte-Carlo* technique called *Russian Roulette* (Arvo and Kirk, 1990). It terminates photons that negligibly contribute to the results, whether they ultimately contribute to the sensor position, which is not known at this point or not, saving computational resources. This is a statistical workaround, under the assumption that the photon would be traced forever. The accuracy can be improved by re-emitting more photons, or by boosting the energy of unterminated ones, making this approach consistent. If a photon is absorbed, the tracing ends; but if it is reflected or refracted, the material's *BRDF* defines its new direction probability (Grobe et al., 2015; McNeil and Lee, 2012). At each photon-surface interaction, light contribution due to photon's position, incoming power and incident direction is stored in the photon map and in the memory for later usage. As point clouds in 3D space, photon maps can handle non-uniform distribution of photons, unlike *illumination maps* (Arvo, 1986) or *texture maps* (Collins, 1995) of larger memory that are also bound to the geometry. Spatial data structures, such as *octrees*, *kd-tree*, *r-trees*, *hashing*, etc. are employed. Those contributions are stored at diffuse surfaces for absorbed photons, yet for reflected or refracted photons, contributions can be stored many times along their paths. The view-independent photon map just represents the diffusely reflected and transmitted portion of light, as all specular reflection and transmission is view dependent.

The second pass sends rays backwardly from the eye to the space to determine the nearest intersecting surfaces. The stored contributions are then used to approximate the illuminance of every pixel due to the nearest photons to the corresponding intersections. For added accuracy, *Density Estimation* is used to determine a specified number of photons around an arbitrary neighborhood at an intersection of the photon map. This is where the rendering equation is split into the terms of direct illumination, specular reflection and indirect illumination (diffuse reflection and caustics), providing not only flexibility to solve each term separately, but also efficiency by choosing the best sampling strategy for each component. Direct illumination is calculated similar to path tracing; specular reflection is handled using raytracing; diffuse reflection is obtained from photon map, while caustics are modeled from a dedicated photon map of higher density that contains photons caused by reflection and refraction rays. Still, precomputed photon mapping eliminates the need for the computationally expensive *density estimation* during image generation (Christensen, 1999; Grobe, 2019a). Besides its efficiency with caustics modelled based on geometric fenestration models, photon mapping is also an efficient means to sample data-driven models and CBDM, which are two important fields of development in the recent years (Grobe, 2019a, 2019b).

Photon mapping, in contrast to other *Monte-Carlo* methods, is biased, where certain bias errors cause blurry results to reduce high-frequency noise; and unfortunately, high gradient illuminance in caustics, which, unlike noise, is needed to be preserved, as an indicator of potential glare in a daylighting context (Schregle, 2003). Yet, this does not yield significant error to the rendering equation on average, as it provides consistent results. The accuracy can be adjusted by increasing the number of photons, but it is limited by the maximum number of photons that can be stored in the photon map and in the memory (Hachisuka et al., 2008; Vardis, 2016). To address this limit, Havran et al. (2005) introduced *Reverse Photon Mapping* that conducts backward raytracing in the first pass, where the *kd-tree* is stored on the intersection points, whereas in the second pass, photons are traced forwardly from light sources. The data structure is used to find the nearby points that photons contribute to, and that information is sent back to the origins of the corresponding rays. This could reduce complexity, especially when many rays are emitted in the first pass. This is particularly important in a daylighting context, where the main light source, the sun, is located outside the internal space of interest, and a small fraction of photons passes through the CFS. Herzog et al. (2007) also suggested storing the visible points from the viewpoint and finding photon contributions to them. Still, bias is not removed, as this depends on *Density Estimation* to define photons near intersections. Hachisuka et al. (2008) created *Progressive Photon Mapping* to extend the original method by combining multiple smaller photon maps to approximate the larger one that cannot fit the memory, allowing to use larger number of photons. Some out-of-core approaches to *photon mapping* addresses this problem specifically in the context of CBDM (Schregle et al., 2016). This considerably removes noise by combining and averaging successive results, yielding seemingly unbiased results; while at the same time progressively reducing the number of photons that contribute to the *density estimation*, reducing the bias. Knaus and Zwicker (2011) conducted a formal statistical analysis of *progressive photon mapping* and arrived at a recommended parametrization that optimally trades off bias and noise. This was incorporated into *Radiance* (Ward and Shakespeare, 1998; Ward, 1994) to simulate the behavior of daylight redirecting systems (Schregle et al., 2015). Hachisuka and Jensen (2009) introduced *Stochastic Progressive Photon Mapping* to handle more complex spaces almost without noisy results. Due to the advantages of this method over the original *photon mapping*, it was widely-adopted after its inception. In another paper, Hachisuka and Jensen (2011) suggested an effective method using *Metropolis Sampling* to find photon contributions due to visible points from the viewpoint. Likewise, Chen et al. (2011) proposed distributing additional photons targeting erroneous parts of resulting images. Vorba (2011) presented *Bi-directional Photon Mapping*, based on *MIS* to improve the biased original algorithm.

5. Implementations of different hardware architectures

The building design community has been profoundly concerned with building performance simulation, as the gradually achieved breakthroughs in the related research exploited the growing computational power of newer and more affordable generations of Central Processing Unit (CPU) (Moore, 1965). Simulations became faster and more feasible, and designers got motivated by the desire for robust results. While they embrace those algorithms, but with some skepticism about their accuracy, their development was driven exclusively within the computer graphics and game-developer communities. This led to the emergence of *raytracing* methods that took advantage of certain computer architectures such as multiprocessors with shared memory (Badouel and Priol, 1992; Green and Paddon, 1989; Muuss, 1987), clusters of computers (Kato, 2002; Parker et al., 1999; Wald et al., 2003, 2001) and render farms.

Still, such systems are hardly used at this time. But they inspired tackling more difficult light transport situations using sophisticated, physically-based LTAs that benefited from the available computational resources, instead of cutting calculation time. This directly relates to Jim Blinn's law: as technology advances, rendering time remains constant; transistors continued to shrink, until reaching a limit that the quantum effects would make them unreliable (Waldrop, 2016). Thus, chip manufacturers were forced to develop multi-core processors rather than faster single-cores (Boix, 2015; Koomey et al., 2011). The post-Moore era (Moore, 1965) was accompanied by attaining accurate LTAs (Maamari, 2004; Reinhart and Fitz, 2006), focusing towards the emerging Parallelism and Interactivity. Parallelism entails executing sets of simultaneous instructions, or threads, in which many scientific fields took advantage of, such as weather modeling, data science, machine learning, and building performance simulation. The following research advanced to exploit Specialized Raytracing Hardware and GPUs, providing unprecedented capabilities and more interactive results.

Humphreys and Ananian (1996) proposed *TigerSHARK* accelerated *raytracing* hardware using digital signal processor. Schmittler et al. (2002) created *SaarCOR* modular *raytracing* hardware for real-time simulations. Dietrich et al. (2003) presented *OpenRT* programming interface for *raytracer*, featuring interactive frame rates. Woop et al. (2005) introduced *Ray Processing Unit* for parallel computations, providing real-time raytracing for dynamic situations. Govindaraju et al. (2008) developed *Copernicus raytracing* acceleration hardware based on multi-core architecture. Also, Spjut et al. (2009) designed *TRaX* multi-core processor for real-time *raytracing* using Multiple Instructions, Multiple Data (MIMD) architecture. Nah et al. (2014) introduced *RayCore*, the first dedicated real-time raytracing hardware for mobile devices. Despite being notable attempts, none of these architectures got widely commercialized.

In 2010, Imagination Technologies acquired Caustic Graphics and introduced *Caustic Professional* cards to integrate *raytracing* architectures into traditional GPUs (Keller et al., 2013). Alternatively, GPUs rapidly became capable sources of parallel threads for large data processing due to the improved performance and cost efficiency over CPUs and other dedicated hardware (McNeil and Lee, 2012; Schardl, 2016). GPUs were also used for heavy simulation tasks, such as acoustics, aerodynamics, solar irradiances and daylighting, because of their Single Instruction, Multiple Data (SIMD) programming model, where the same instruction is executed simultaneously on multiple data-paths, providing larger speeds by parallelism. Pixel-based methods such as *raytracing* can be easily parallelized because of the independency of the traced paths and the recursive nature of ray-surface interactions. Yet, this involves different data with different instructions, requiring Single Instruction, Multiple Thread (SIMT) programming model (Jones and Reinhart, 2015). Therefore, to apply raytracing on GPUs, programming language models, such as *Nvidia's Compute Unified Device Architecture (CUDA)* and *Khronos Group's Open Computing Language (OpenCL)*, can be used to enable GPUs' virtual instruction sets and parallel computational elements (Aila and Laine, 2009; Wang et al., 2009).

The rise of raytracing frameworks such as Nvidia's *OptiX* (Parker et al., 2010) and Intel's *OSPRay* (Wald et al., 2017), in addition to other libraries such as Intel's *Embree* (Wald et al., 2014) and AMD's *FireRays* (AMD, 2015), enabled the expansion of hardware-optimized raytracing engines. *Radiance* was modified to perform *progressive path tracing* to detect potential ray-surface intersections (Jones and Reinhart, 2016), and for interactive visualizations of results as well (Andersen et al., 2013) on GPUs using *OptiX*. In addition, data processing can be partitioned and distributed over multiple GPUs, yielding increased parallelism (Navarro et al., 2014). By the end of 2018, Nvidia and Microsoft's new DirectX Raytracing (Microsoft, 2018) announced the *RTX* developer library (Nvidia, 2018) of faster real-time raytracing for developers and professionals, supported by Nvidia's *Volta* and subsequent *Turing* microarchitectures. The latter including dedicated artificial intelligence and accelerated raytracing processors. Based on *Turing*, Nvidia also introduced *GeForce RTX 20* as consumer-level GPUs (Nvidia, 2019) that support real-time *raytracing*. AMD also announced *Radeon* GPUs, based on AMD's Navi microarchitecture that supports real-time *raytracing* as well (AMD, 2017). Moreover, Intel's *Xe* microarchitecture was revealed, which ranges from integrated chips on CPUs to dedicated GPUs for developers (Intel, 2018).

6. Daylight simulation tools

Apart from different LTAs, there is a larger number of simulation tools used for scientific and professional applications, each supports one or more LTAs to model and analyze the physical behavior of light under different conditions. Their diverse capabilities made architects uncertain to choose the "best" one of valid calculations. This section provides detailed descriptions of 50 prevalent tools **(**Fig. 4**)** (Davoodi et al., 2014; IBPSA, 2019; Iversen et al., 2013; Jakica, 2018; Kaempf et al., 2016; Kirimtat et al., 2016; Wong and Wong, 2017), considering their capabilities to support LTAs, analysis types, accuracy, computation, integration with other programs and licensing **(**Table 1**)**. 70% of the reviewed tools are still being released, while 38% were first released after 2006; and only 12% by 2012. 14% of tools are actually simulation engines, such as *Radiance*, *DAYSIM*, *DOE-2* and *EnergyPlus* that were not easy to use. Their complicated settings and features, while valued by researchers, discourage being utilized for casual use by new practitioners. Thus, many newly-developed tools interface such engines, without including many of their complicated features. **(**Table 1**)** reveals that 54% of tools are actually *Radiance-based*, confirming the results of

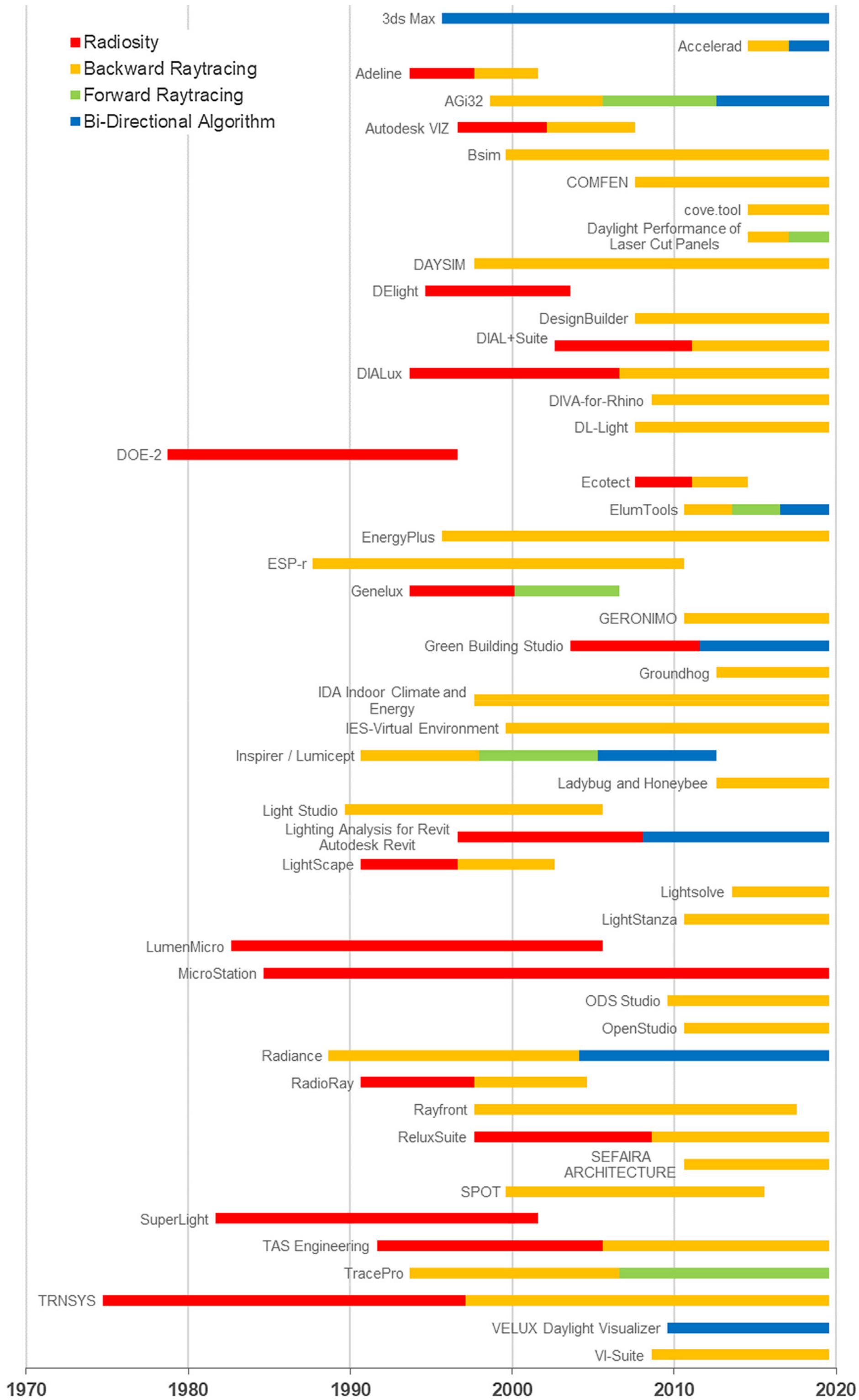

Fig. 4. A summary of 50 simulation tools in terms of the used light transport algorithm.

previous survey by Reinhart and Fitz (2006); and the reason being this is that *Radiance* has undergone extensive validation, as previously mentioned. While all the reviewed tools perform photometric simulations, only 66%, 62% and 54% of them support visualization, radiometric, and glare analyses, respectively. The integration between 3D modeling and simulation tools saw growing interests from tool developers and researchers, whether standalone (70%), plugins (42%) or web-based applications (6%).

7. Discussion

Considering the reviewed LTAs, *Radiosity* provides more accurate results than split-flux (Yoon et al., 2014a). It can calculate diffuse interreflections between surfaces for several views without recalculation. Still, its early developments showed limits when used for daylight simulations in buildings, since it could not effectively model specular reflections or transparency. It can handle simple spaces, but the

Table 1
A summary of 50 simulation tools capabilities.

Simulation tool	Simulation engine	Developer	First/Last releases (September 2019)	Light transport algorithm	Analysis	Accuracy	Computation	Integration with other programs	Licensing	Web site
3ds Max	ART	Autodesk	1996-till now	BDA	Visualization Photometric (Illum., Lum.)	Unbiased Biased	CPU (support multi. cores)	Standalone	Rental (Commercial) Educational	https://www.autodesk.com/products/3ds-max/overview
Accelerad	Radiance and OptiX	Jones, N. at MIT Sustainable Design Lab	2015-till now	BRT BDA	Visualization Photometric (Illum., Lum.) Radiometric (Irrad., Rad.)Glare	Unbiased Biased	GPU	Standalone	Open Source	https://nljones.github.io/mit_thesis.html
Adeline	Radiance and SuperLite	Fraunhofer-Institute für Bauphysik	1994–2002	BRT FEM	Visualization Photometric (Illum., Lum.) Glare	Unbiased Biased	CPU	Standalone	Retired	http://www.iea-adeline.de/index.html
AGi32	Internal Engine	Lighting Analysts, Inc.	1999-till now	FRT BRT BDA FEM	Visualization Photometric (Illum., Lum.) Glare	Unbiased Biased	CPU	Standalone	Rental (Commercial)	http://www.agi32.com
Autodesk VIZ	Internal Engine	Autodesk	1997–2008	BRT FEM	Visualization Photometric (Illum., Lum.)	Unbiased Biased	CPU (support multi. cores)	Standalone	Retired	N/A
Bsim	Radiance and DAYSIM	BSI, Aalborg University Copenhagen	2000-till now	BRT	Photometric (Illum., Lum.)	Unbiased	CPU	Standalone	Rental (Commercial) Educational	https://sbi.dk/bsim/Pages/BSim_Building_Simulation.aspx
COMFEN	Radiance,DAYSIM and EnergyPlus	Windows and Daylighting, LBNL	2008-till now	BRT	Photometric (Illum., Lum.) Glare	Unbiased	CPU	Standalone	Public	https://windows.lbl.gov/software/comfen
cove.tool	EnergyPlus	Pattern r + d	2015-till now	BRT	Visualization Photometric (Illum., Lum.) Radiometric (Irrad., Rad.) Glare	Unbiased	CPU Cloud-Based	Plugin (Revit, SketchUp 3D, Rhinoceros and Grasshopper)	Rental (Commercial) Educational	https://www.covetool.com/
Daylight Performance of Laser Cut Panels	Radiance and EnergyPlus	CARBSE, CEPT University and LBNL	2015-till now	FRT BRT	Photometric (Illum., Lum.)	Unbiased	CPU	Standalone	N/A	http://carbse.org/research-tools/
DAYSIM	Internal Engine and Radiance	Reinhart, C. F. and National Research Council	1998-till now	BRT	Photometric (Illum., Lum.) Radiometric (Irrad., Rad.) Glare	Biased	CPU (support multi. cores)	Standalone	Open Source	https://daysim.ning.com
DElight	DOE-2, EnergyPlus and SuperLite	Hitchcock, R. J.	1995–2004	FEM	Photometric (Illum., Lum.) Radiometric (Irrad., Rad.)	Biased	CPU	Standalone	Retired	N/A
DesignBuilder	Radiance, DAYSIM and EnergyPlus	DesignBuilder Software Ltd	2008-till now	BRT	Visualization Photometric (Illum., Lum.) Radiometric (Irrad., Rad.) Glare	Unbiased	CPU (support multi. cores)	Web-Based	Proprietary (Commercial) Educational	https://designbuilder.co.uk/

(*continued on next page*)

Table 1 (*continued*)

Simulation tool	Simulation engine	Developer	First/Last releases (September 2019)	Light transport algorithm	Analysis	Accuracy	Computation	Integration with other programs	Licensing	Web site
DIAL + Suite	Radiance and DAYSIM	Estia, LESO-PB - EPFL	2003-till now	BRT FEM	Photometric (Illum., Lum.) Radiometric (Irrad., Rad.)	Unbiased Biased	CPU	Standalone	Proprietary (Commercial)	https://www.dialplus.ch/home
DIALux	Internal Engine	DIAL GmbH	1994-till now	BRT FEM	Visualization Photometric (Illum., Lum.)	Unbiased	CPU (support multi. cores)	Standalone	Public	https://www.dial.de/en/home/
DIVA-for-Rhino	Radiance and DAYSIM	Solemma, Harvard University	2009-till now	BRT	Visualization Photometric (Illum., Lum.) Radiometric (Irrad., Rad.) Glare	Unbiased	CPU (support multi. cores)	Plugin (Rhinoceros 3D and Grasshopper)	Proprietary (Commercial) Educational	http://www.solemma.net/Diva.html
DL-Light	Radiance	DE LUMINAE	2008-till now	BRT	Photometric (Illum., Lum.) Radiometric (Irrad., Rad.)	Unbiased	CPU	Plugin (SketchUp 3D)	Proprietary (Commercial) Educational	https://deluminaelab.com/docs/dl-light-en/
DOE-2	Internal Engine	LBNL and US DOE	1979–1997	FEM	Photometric (Illum., Lum.) Radiometric (Irrad., Rad.) Glare	Biased	CPU	Standalone	Retired	http://doe2.com
Ecotect	Radiance	Autodesk	2008–2015	BRT FEM	Photometric (Illum., Lum.) Radiometric (Irrad., Rad.)	Unbiased Biased	CPU	Standalone	Retired	https://knowledge.autodesk.com/support/ecotect-analysis
ElumTools	Internal Engine	Lighting Analysts, Inc.	2011-till now	FRT BRT BDA FEM	Visualization Photometric (Illum., Lum.)	Unbiased Biased	CPU	Plugin (Revit)	Rental (Commercial)	http://lightinganalysts.com/software-products/elumtools/overview/
EnergyPlus	Internal Engine	LBNL and US DOE	1996-till now	BRT	Photometric (Illum., Lum.) Radiometric (Irrad., Rad.) Glare	Unbiased	CPU	Standalone	Open Source	https://energyplus.net/
ESP-r	Internal Engine and Radiance	University of Strathclyde's Energy Sys. Research Unit	1988–2011	BRT	Photometric (Illum., Lum.) Radiometric (Irrad., Rad.) Glare	Unbiased	CPU	Standalone	Open Source	http://www.esru.strath.ac.uk/Programs/ESP-r.htm
Genelux	Internal Engine	l'Ecole des Travaux Publics de l'Etat	1994–2007	FRT FEM	Visualization Photometric (Illum., Lum.) Radiometric (Irrad., Rad.) Glare	Biased	CPU	Standalone	Retired	N/A
GERONIMO	Radiance	LESO-PB - EPFL	2011-till now	BRT	Visualization Photometric (Illum., Lum.) Glare	Unbiased	CPU	Standalone Plugin (N/S)	Public	https://www.epfl.ch/labs/leso/transfer/software/geronimo/

(*continued on next page*)

Table 1 (*continued*)

Simulation tool	Simulation engine	Developer	First/Last releases (September 2019)	Light transport algorithm	Analysis	Accuracy	Computation	Integration with other programs	Licensing	Web site
Green Building Studio	DOE-2	Autodesk	2004-till now	BDA FEM	Photometric (Illum., Lum.) Radiometric (Irrad., Rad.)	Biased	Cloud-Based	Standalone Plugin (Revit)	N/S	https://gbs.autodesk.com/GBS/
Groundhog	Radiance	Molina, G., Vera, S. and Bustamante, W.	2013-till now	BRT	Visualization Photometric (Illum., Lum.) Radiometric (Irrad., Rad.) Glare	Unbiased	CPU	Plugin (SketchUp 3D)	Open Source	https://groundhoglighting.gitbook.io/groundhog-bible/
IDA Indoor Climate and Energy	Radiance and DAYSIM	EQUA	1998-till now	BRT	Photometric (Illum., Lum.)	Unbiased	CPU	Standalone	Proprietary (Commercial) Educational	https://www.equa.se/en/ida-ice
IES-Virtual Environment	Radiance,DAYSIM and EnergyPlus	Integrated Environmental Solutions Ltd	2000-till now	BRT	Visualization Photometric (Illum., Lum.) Radiometric (Irrad., Rad.)	Unbiased	CPU (support multi. cores) Cloud-Based	Standalone Plugin (SketchUp, Revit and Vectorworks)	Rental (Commercial) Educational	https://www.iesve.com/software/virtual-environment
Inspirer / Lumicept	Internal Engine	Integra	1991–2013	FRT BRT BDA FEM	Visualization Photometric (Illum., Lum.) Radiometric (Irrad., Rad.) Glare	Unbiased	CPU (support multi. cores)	Standalone	N/S	https://integra.jp/en/products/lumicept
Ladybug and Honeybee	Radiance,DAYSIM and EnergyPlus	Ladybug Tools	2013-till now	BRT	Visualization Photometric (Illum., Lum.) Radiometric (Irrad., Rad.) Glare	Unbiased	CPU (support multi. cores)	Plugin (Rhinoceros 3D and Grasshopper)	Open Source	https://www.ladybug.tools/
Light Studio	Radiance	Die Lichtplaner	1990–2006	BRT	Visualization Photometric (Illum., Lum.)	Unbiased	CPU (support multi. cores)	Plugin (3ds Max)	Retired	https://www.lichtplaner.com/
Lighting Analysis for Revit Autodesk Revit	A360	Autodesk	1997-till now	BDA FEM	Visualization Photometric (Illum., Lum.) Radiometric (Irrad., Rad.) Glare	Biased	Cloud-Based	Plugin (Revit)	Rental (Commercial) Educational	https://www.autodesk.com/products/revit/overview
LightScape	Internal Engine	Autodesk	1991–2003	BRT FEM	Visualization Photometric (Illum., Lum.) Radiometric (Irrad., Rad.) Glare	Unbiased Biased	CPU (support multi. cores)	Standalone	Retired	N/A
Lightsolve	Radiance and OptiX	LIPID EPFL	2014-till now	BRT	Visualization Photometric (Illum., Lum.) Glare	Unbiased	CPU (support multi. cores) GPU	Plugin (N/S)	Public	http://lightsolve.epfl.ch/
LightStanza	Radiance and DAYSIM	Light Foundry	2011-till now	BRT	Photometric (Illum., Lum.) Glare	Unbiased	Cloud-Based	Web-Based	Rental (Commercial) Educational	http://lightstanza.com/

(*continued on next page*)

Table 1 (*continued*)

Simulation tool	Simulation engine	Developer	First/Last releases (September 2019)	Light transport algorithm	Analysis	Accuracy	Computation	Integration with other programs	Licensing	Web site
LumenMicro	Internal Engine	Lighting Technologies	1983–2006	FEM	Photometric (Illum., Lum.)	Biased	CPU	Standalone	Retired	N/A
MicroStation	Internal Engine	Bentley Systems	1985-till now	FEM	Visualization Photometric (Illum., Lum.) Radiometric (Irrad., Rad.)	Biased	CPU (support multi. cores)	Standalone	Proprietary (Commercial)	https://bentley.com
ODS Studio	Radiance and EnergyPlus	ODS Engineering	2010-till now	BRT	Visualization Photometric (Illum., Lum.) Radiometric (Irrad., Rad.)	Unbiased	CPU (support multi. cores)	Plugin (Blender 3D)	Rental (Commercial)	https://www.ods-engineering.com/tools/ods-studio/
OpenStudio	Radiance,DAYSIM and EnergyPlus	Alliance for Sustainable Energy, ANL, NREL, LBNL	2011-till now	BRT	Visualization Photometric (Illum., Lum.) Radiometric (Irrad., Rad.) Glare	Unbiased	CPU (support multi. cores) Cloud-Based	Standalone Plugin (N/S)	Open Source	https://www.openstudio.net/
Radiance	Internal Engine	Ward, G. J. at LBNL	1989-till now	BRT BDA	Visualization Photometric (Illum., Lum.) Radiometric (Irrad., Rad.) Glare	Unbiased Biased (Photon Mapping)	CPU (support multi. cores)	Standalone Plugin (Many CAD and BIM tools)	Open Source	https://www.radiance-online.org
RadioRay	Internal Engine	Autodesk	1991–2005	BRT FEM	Visualization Photometric (Illum., Lum.)	Unbiased Biased	CPU (support multi. cores)	Plugin (Kinetix (Autodesk)	Retired	N/A
Rayfront	Radiance	Georg Mischler	1998–2018	BRT	Visualization Photometric (Illum., Lum.) Radiometric (Irrad., Rad.) Glare	Unbiased	CPU	Standalone Plugin (AutoCAD and 3Dsolar)	N/A	https://www.schorsch.com/en/software/rayfront/
ReluxSuite	Internal Engine and Radiance	Relux Informatik AG	1998-till now	BRT FEM	Visualization Photometric (Illum., Lum.) Radiometric (Irrad., Rad.)	Unbiased Biased	CPU	Standalone Plugin (Revit)	Public	https://reluxnet.relux.com/en/
SEFAIRA ARCHITECTURE	Radiance,DAYSIM and EnergyPlus	SEFAIRA	2011-till now	BRT	Visualization Photometric (Illum., Lum.) Radiometric (Irrad., Rad.)	Unbiased	Cloud-Based	Plugin (Revit and SketchUp 3D) Web-Based	Public Rental (Commercial)	https://sefaira.com/sefaira-architecture/
SPOT	Radiance and DAYSIM	Architectural Energy Corporation	2000–2016	BRT	Visualization Photometric (Illum., Lum.) Radiometric (Irrad., Rad.) Glare	Unbiased	CPU	Standalone	Retired	http://www.daylightinginnovations.com/
SuperLight	Internal Engine	IEA Task 21 and LBNL	1982–2002	FEM	Photometric (Illum., Lum.) Glare	Biased	CPU	Standalone	Retired	N/A

(*continued on next page*)

Table 1 (*continued*)

Simulation tool	Simulation engine	Developer	First/Last releases (September 2019)	Light transport algorithm	Analysis	Accuracy	Computation	Integration with other programs	Licensing	Web site
TAS Engineering	Internal Engine	Environmental Design Solutions Ltd	1992-till now	BRT FEM	Visualization Photometric (Illum., Lum.)	Biased	CPU (support multi. cores)	Standalone	Proprietary (Commercial)	https://www.edsl.net/tas-engineering/
TracePro	Internal Engine	Lambda Research Corporation	1994-till now	FRT BRT	Visualization Photometric (Illum., Lum.) Radiometric (Irrad., Rad.)	Unbiased	CPU (support multi. cores)	Standalone Plugin (Many CAD tools)	Proprietary (Commercial)	https://www.lambdares.com/tracepro/
TRNSYS	Internal Engine	Thermal Energy System Specialists	1975-till now	BRT FEM	Photometric (Illum., Lum.) Radiometric (Irrad., Rad.) Glare	Unbiased Biased	CPU	Standalone	Proprietary (Commercial)	http://www.trnsys.com/
VELUX Daylight Visualizer	Internal Engine	VELUX, LUXION	2010-till now	BDA	Visualization Photometric (Illum., Lum.)	Biased	CPU	Standalone	Public	https://www.velux.com/article/2016/daylight-visualizer
VI-Suite	Radiance and EnergyPlus	Southall, R. at University of Brighton	2009-till now	BRT	Visualization Photometric (Illum., Lum.) Radiometric (Irrad., Rad.) Glare	Unbiased	CPU	Plugin (Blender 3D)	Open Source	http://arts.brighton.ac.uk/projects/vi-suite

FEM: Finite Element Method (Radiosity); BRT: Backward Raytracing; FRT: Forward Raytracing; BDA: Bi-Directional Algorithm; N/A: Not Available; N/S: Not Specified.

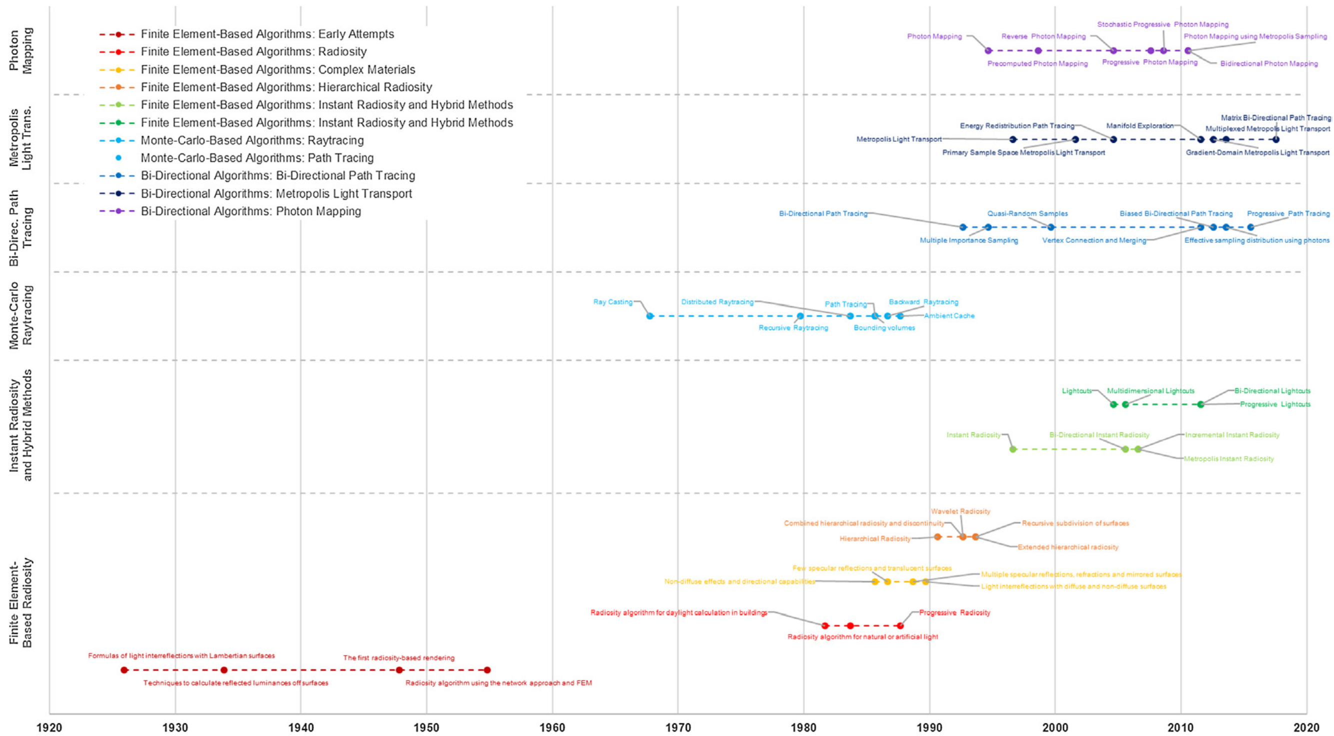

Fig. 5. A comprehensive timeline chart of light transport algorithms.

required computational power increases significantly with unoptimized implementations (Wallace et al., 1987). Still, an additional benefit of this LTA is that this solution is view-independent, enabling some of the first interactive photorealistic walkthroughs. Enhancements were added to support complex reflections as post processing, assuming specular reflections constitute a small portion of internal light distribution (Cohen and Wallace, 1993). However, those progresses came at the expense of more memory.

Raytracing and *Path Tracing* are accurate, unbiased, physically-based algorithms, owing to their ability to handle various optical properties and complex spaces (Oh and Haberl, 2016c; Reinhart, 2011; Watt, 2000). As they usually entail extensive computations, some materials would require generating more rays recursively to model difficult ray-surface interactions. Unlike *radiosity*, considering multiple viewing positions would still need more computations for every new position. In situations that involve specular reflections, and if the light comes from small areas, such as CFS, *forward raytracing* is more efficient. But in internal spaces of diffuse reflections, *path tracing* is more suitable (Vardis, 2016), including other *bi-directional* LTAs. However, some effects are difficult to be handled by *path tracing* methods, such as subsurface scattering, fluorescence, iridescence and spectral rendering; yet, such cases are not of main concern to model daylight inside buildings. Some CFSs exhibit considerable spectral effects, and while there is a need to investigate such complex systems, most LTAs only operate in the simplified RGB domain; although the SIMD capabilities of modern CPUs and GPUs could accommodate this demand.

Photon Mapping can handle difficult situations, such as light tubes and CFS, faster than other unbiased, yet memory-efficient, *Monte-Carlo* methods; and if the used memory is not of concern, *photon mapping* is more effective, producing less noticeable errors. As it combines the strengths from different methods, *photon mapping* can model light refraction through transparent materials, diffuse interreflection between objects, subsurface scattering in translucent materials, besides complex caustics (Jensen et al., 2001). Still, these simulations are prone to some degree of inherent bias and noise, unless more sophisticated variants are used, such as *progressive photon mapping*. Photon map can also be reused to calculate DC with a variety of sky subdivisions for different viewpoints without additional calculations (Grobe et al., 2015; Schregle et al., 2016). Moreover, since there is no patent on photon mapping (Jensen and Christensen, 2000), it got widely-integrated into many existing render engines.

Unbiased LTAs refer to rendering methods that do not introduce systematic errors to the approximation. They converge to physically-accurate solutions on average, considering multiple iterations, where the difference between simulated and actual results values is negligible (Jakica, 2018). However, their probabilistic nature makes them computationally inefficient and slow in some situations. In contrast, biased LTAs utilize simple assumptions to improve computational efficiency and time (Crane, 2006). The error of unbiased LTAs is manifested as noise, while that of biased LTAs is due to both noise and bias (Hachisuka, 2013). However, this does not necessarily mean that biased LTAs are inaccurate, as they can converge to the correct solutions if they are consistent (Crane, 2006; Jakica, 2018). In other words, by increasing the computational time, approximation error can approach zero. Also, with the same sample size, some biased LTAs can yield lower variance error than unbiased (Hachisuka, 2013). Besides, biased and consistent LTAs can handle situations that unbiased algorithms cannot, for instance density estimation, allows for efficiently handling the problem of insufficient techniques (Keller, 2013). Still, biased LTAs exhibit blurry solutions to eliminate high-frequency noise. Nonetheless, for real-time simulations, bias can be accepted to achieve accelerated computations (Crane, 2006). As explained in previous sections, *Path Tracing, Bi-Directional Path Tracing* and *Metropolis Light Transport* are unbiased and consistent algorithms. On the other hand, *Photon Mapping* is a biased and consistent algorithm, while *Radiosity* is biased and inconsistent (Crane, 2006; Hachisuka, 2013).

GPU-based simulation tools have been gaining popularity over the past years due to the expansion of hardware-optimized raytracing engines. Also, high-end GPUs can efficiently handle the computation of parallel identical problems, but get inefficient when large amounts of data have to be moved. There are reasons why, for example, Accelerad (Jones and Reinhart, 2016) supports a small subset of Radiance models and functionalities. Although there is a growing number of simulation tools that support multicore and multi-processor computations

(Table 1), in practice, only few LTAs are employed on anything but such architectures. Distributed memory systems (clusters) are supported but are not used currently that much. A recent application of a LTA on special hardware includes the use of CBDM 5-Phase method to control sun-shades (Wu et al., 2019).

According to the literature survey, the following detailed chart demonstrates the timeline of LTAs, with regard to their different groupings (Fig. 5).

8. Conclusion

As an active area of research, the exploration in the field of LTAs continues to grow. LTAs have come a long way since the inception of the first ray casting algorithm in 1968. Within acceptable timeframes, modern LTAs can now facilitate physically-accurate solutions to model difficult light transport situations inside buildings, supporting a wide range of material types and scene complexities. This was achieved by extensively investigating the foundations that sculptured photometric simulations. In such domains, this review traces back the gradual evolutions that drove the advancement of simulation tools. A good understanding of fundamental issues related to modern LTAs is the key to well-recognized simulation tools. Thus, the accompanied summary of 50 widespread simulation tools overviews the legacy and current tools, along with their abilities in terms of the used LTA, analysis types, accuracy, computation, integration with other programs and licensing. This should help architects of different backgrounds comprehending the essential daylight-related topics, guiding their decisions on suitable tools to use in their practices.

References

Aila, T., Laine, S., 2009. Understanding the efficiency of ray traversal on GPUs. Proc. High-Perform. Graph. 145–149.

Aizlewood, M.E., Littlefair, P.J., 1996. Daylight prediction methods: a survey of their use. Conf. Pap. CIBSE Natl. Light. Conf. 126–140.

Alshaibani, K.A., 1996. Prediction of Interior Daylight Under Clear Sky Conditions. Ph.D. University of Sheffield, South Yorkshire, United Kingdom.

Amasyali, K., El-Gohary, N.M., 2018. A review of data-driven building energy consumption prediction studies. Renew. Sustain. Energy Rev. 81, 1192–1205.

AMD, 2017. AMD Unveils Expanding Set of High-Performance Products and Technologies Propelling Next Phase of Growth [WWW Document]. URL https://www.amd.com/en/press-releases/amd-unveils-expanding-2017may16 (accessed 6.15.19).

AMD, 2015. FireRays SDK [WWW Document]. URL http://developer.amd.com/wordpress/media/2015/07/157205-A_AMD_FireRays_Document_Original_44796.pdf (accessed 6.10.19).

Andersen, M., Guillemin, A., Ámundadóttir, M.L., Rockcastle, S.F., 2013. Beyond illumination: an interactive simulation framework for non-visual and perceptual aspects of daylighting performance. In: Proceedings of the 13th Conference of International Building Performance Simulation Association (BS2013). Chambery, France, pp. 2749–2756.

Appel, A., 1968. Some techniques for shading machine renderings of solids, in: Proceedings of the ACM Spring Joint Computer Conference. Massachusetts, United States, pp. 37–45.

Arvo, J., 1986. Backward ray tracing. Dev. Ray Tracing. SIGGRAPH '86 course notes 12, 259–263.

Arvo, J., Kirk, D., 1990. Particle transport and image synthesis. Comput. Graph. (ACM) 24, 53–66.

Arvo, J., Kirk, D., 1987. Fast ray tracing by ray classification. ACM Siggraph Comput. Graph. 21, 55–64. https://doi.org/10.1145/37402.37409.

Arvo, J., Torrance, K., Smits, B., 1994. A framework for the analysis of error in global illumination algorithms. In: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques. ACM, pp. 75–84.

Ashdown, I., 1996. Lighting for architects. Comput. Graph. (ACM) 19, 38–46.

Ashikhmin, M., Shirley, P., 2000. An anisotropic phong BRDF model. J. Graph. Tools 5, 25–32.

Asmail, C., 1991. Bidirectional scattering distribution function (BSDF): a systematized bibliography. J. Res. Natl. Inst. Stand. Technol. 96, 215–223.

Athienitis, A.K., Tzempelikos, A., 2002. A methodology for simulation of daylight room illuminance distribution and light dimming for a room with a controlled shading device. Sol. Energy 72, 271–281.

Ayoub, M., 2019a. A multivariate regression to predict daylighting and energy consumption of residential buildings within hybrid settlements in hot-desert climates. Indoor Built Environ. 28, 848–866 https://doi.org/326X18798164.

Ayoub, M., 2019b. 100 years of daylighting: a chronological review of daylight prediction and calculation methods. Sol. Energy 194C, 360–390.

Badouel, D., Priol, T., 1992. An efficient parallel ray tracing scheme for highly parallel architectures. Adv. Comput. Graph. Hardware 93–106.

Boix, C., 2015. Technological progress. Political Order Inequality 92–127. https://doi.org/10.1017/cbo9781316105504.004.

Brembilla, E., Chi, D.A., Hopfe, C.J., Mardaljevic, J., 2019. Evaluation of climate-based daylighting techniques for complex fenestration and shading systems. Energy Build. 203, 109454.

Brembilla, E., Mardaljevic, J., 2019. Climate-Based Daylight Modelling for compliance verification: benchmarking multiple state-of-the-art methods. Build. Environ. 158, 151–164.

Bryan, H., Autif, S., 2002. Lighting/daylighting analysis: a comparison, in: Proceedings of the 27th National Passive Solar Conference. Nevada, United States, pp. 521–526.

Bryan, H.J., Clear, R.D., 1981. Calculating interior daylight illumination with a programmable hand calculator. J. Illum. Eng. Soc. 10, 219–227. https://doi.org/10.1080/00994480.1980.10748615.

Carroll, W.L., 1999. Daylighting Simulation: Methods, Algorithms, and Resources.

Chaitanya, C.R.A., Belcour, L., Hachisuka, T., Premoze, S., Pantaleoni, J., Nowrouzezahrai, D., 2018. Matrix bidirectional path tracing, in: Proceedings of the Eurographics Symposium on Rendering. Eurographics Association, pp. 23–32.

Chen, J., Wang, B., Yong, J.H., 2011. Improved stochastic progressive photon mapping with Metropolis sampling. Comput. Graph. Forum (Proceedings Eurographics 2011) 30, 1205–1213.

Christensen, P.H., 1999. Faster photon map global illumination. J. Graph. Tools 4, 1–10.

CIE, 2006. Test cases to assess the accuracy of lighting computer programs. Vienna, Austria.

Cline, D., Talbot, J., Egbert, P., 2005. Energy redistribution path tracing. ACM Trans. Graph. 24, 1186–1195.

Cohen, M.F., Chen, S.E., Wallace, J.R., Greenberg, D.P., 1988. A progressive refinement approach to fast radiosity image generation. Commun. ACM 22, 75–84.

Cohen, M.F., Wallace, J.R., 1993. Radiosity and Realistic Image Synthesis. Academic Press Professional, Massachusetts, United States.

Collins, S., 1995. Adaptive splatting for specular to diffuse light transport, in. In: Photorealistic Rendering Techniques. Springer, Berlin, Heidelberg, pp. 121–135.

Cook, R.L., Porter, T., Carpenter, L., 1984. Distributed ray tracing. Comput. Graph. (ACM) 18, 137–145.

Crane, K., 2006. Bias in rendering [WWW Document]. URL https://www.cs.cmu.edu/~kmcrane/Projects/Other/BiasInRendering.pdf (accessed 10.17.19).

Crawley, D.B., 2007. Creating Weather Files for Climate Change and Urbanization Impacts Analysis, in: Proceedings of the Building Simulation 2007. Beijing, China, pp. 1075–1082.

Dachsbacher, C., Křivánek, J., Hašan, M., Arbree, A., Walter, B., Novák, J., 2014. Scalable realistic rendering with many-light methods. Comput. Graph. Forum 33, 88–104.

Danks, R., Good, J., Sinclair, R., 2016. Assessing reflected sunlight from building facades: a literature review and proposed criteria. Build. Environ. 103, 193–202.

Davidovič, T., Georgiev, I., Slusallek, P., 2012. Progressive lightcuts for GPU, in: ACM SIGGRAPH 2012 Talks. p. 1.

Davoodi, A., Johansson, P., Enger, J., 2014. Comparison of Lighting Simulation Tools with Focus on Lighting Quality. In: Proceedings of the 8th International Conference on Improving Energy Efficiency in Commercial Building (IEECB). pp. 15–29.

de Boer, J., Erhorn, H., 1998. Survey of Simple Design Tools.

Dietrich, A., Wald, I., Benthin, C., Slusallek, P., 2003. The OpenRT application programming interface–towards a common API for interactive ray tracing. In: Proceedings of the 2003 OpenSG Symposium. pp. 23–31.

Donn, M., Xu, D., Harrison, D., Maamari, F., 2007. Using simulation software calibration tests as a consumer guide-A geasibility study using lighting simulation software. In: Proceedings of the 10th Conference of International Building Performance Simulation Association (IBPSA). China, pp. 1999–2006.

Dresler, A., 1954. The "reflected component" in daylighting design. Trans. Illum. Eng. Soc. 19, 50–60. https://doi.org/10.1177/147715355401900203.

Dutre, P., Bekaert, P., Bala, K., 2006. Advanced global illumination. AK Peters Ltd.

Dutre, P., Lafortune, E.P., Willems, Y.D., 1993. Monte Carlo Light Tracing with Direct Computation of Pixel Intensities. In: Alvor, P. (Ed.), Proceedings of the 3rd International Conference on Computational Graphics and Visualisation Techniques. pp. 128–137.

Frühling, H.G., 1928. Tagesbeleuchtung von Innenräumen: ihre Messung und ihre Berechnung nach der Wirkungsgradmethode [Daylighting of Indoors, Their Measurement and Calculation by the Efficiency Method]. Technical University of Berlin, Berlin, Germany.

Galasiu, A.D., Atif, M.R., 2002. Applicability of daylighting computer modeling in real case studies: comparison between measured and simulated daylight availability and lighting consumption. Build. Environ. 37, 363–377.

Garland, M., Willmott, A., Heckbert, P.S., 2001. Hierarchical face clustering on polygonal surfaces. In: Proceedings of the 2001 Symposium on Interactive 3D Graphics. pp. 49–58.

Geisler-Moroder, D., Dür, A., 2008. Validation of Radiance against CIE171: 2006 and improved adaptive subdivision of circular light sources. Fribourg, Switzerland.

Geisler-Moroder, D., Lee, E.S., Ward, G.J., 2017. Validation of the Five-Phase Method for Simulating Complex Fenestration Systems with Radiance against Field Measurements. In: Proceedings of the 15th International Building Performance Simulation Association (IBPSA 2017). San Francisco, United States, pp. 1523–1531.

Georgiev, I., Ize, T., Farnsworth, M., Montoya-Vozmediano, R., King, A., Lommel, B.V., Jimenez, A., Anson, O., Ogaki, S., Johnston, E., Herubel, A., 2018. Arnold: a brute-force production path tracer. ACM Trans. Graph. 37, 32.

Georgiev, I., Krivánek, J., Davidovič, T., Slusallek, P., 2012. Light transport simulation with vertex connection and merging. ACM Trans. Graph. 31, 192.

Glassner, A.S., 1995. Principles of Digital Image Synthesis. Morgan Kaufmann Publishers, California, United States.

Goldstein, R.A., Nagel, R., 1971. 3-D Visual simulation. Simulation 16, 25–31. https://doi.org/10.1177/003754977101600104.
Goral, C.M., Torrance, K.E., Greenberg, D.P., Battaile, B., 1984. Modeling the interaction of light between diffuse surfaces. ACM Siggraph. Comput. Graph. 18, 213–222. https://doi.org/10.1145/964965.808601.
Gortler, S., Schröder, P., Cohen, M.F., Hanrahan, P., 1993. Wavelet Radiosity. In: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1993). ACM Press, Florida, United States, pp. 221–230.
Govindaraju, V., Djeu, P., Sankaralingham, K., Vernon, M., Mark, W.R., 2008. Toward a multicore architecture for real-time ray-tracing. In: Proceedings of the 41st Annual IEEE/ACM International Symposium on Microarchitecture. pp. 176–187.
Green, S.A., Paddon, D.J., 1989. Exploiting coherence for multiprocessor ray tracing. IEEE Comput. Graph. Appl. 9, 12–26.
Grobe, L.O., 2019a. Photon mapping to accelerate daylight simulation with high-resolution, data-driven fenestration models. In: Journal of Physics: Conference Series. IOP Publishing, p. 012154.
Grobe, L.O., 2019b. Photon mapping in image-based visual comfort assessments with BSDF models of high resolution. J. Build. Perform. Simul. 12, 745–758.
Grobe, L.O., Schregle, R., Scartezzini, J.-L., Wittkopf, S., Bauer, C., 2015. EvalDRC: a tool for annual characterisation of daylight redirecting components with photon mapping. In: Proceedings of International Conference CISBAT 2015 “Future Buildings and Districts – Sustainability from Nano to Urban Scale”. pp. 217–222. https://doi.org/10.5075/EPFL-CISBAT2015-217-222.
Guarnera, D., Guarnera, G.C., Ghosh, A., Denk, C., Glencross, M., 2016. BRDF representation and acquisition. Comput. Graph. Forum 35, 625–650.
Hachisuka, T., 2013. Five common misconceptions about bias in light transport simulation [WWW Document]. URL https://www.ci.i.u-tokyo.ac.jp/~hachisuka/misc.pdf (accessed 10.19.19).
Hachisuka, T., Jensen, H.W., 2011. Robust adaptive photon tracing using photon path visibility. ACM Trans. Graph. 30 114:1–114: 11.
Hachisuka, T., Jensen, H.W., 2009. Stochastic progressive photon mapping. ACM Trans. Graph. (Proceedings SIGGRAPH Asia 2009) 28, 141:1–141:8.
Hachisuka, T., Kaplanyan, A.S., Dachsbacher, C., 2014. Multiplexed metropolis light transport. ACM Trans. Graph. 33, 100.
Hachisuka, T., Ogaki, S., Jensen, H.W., 2008. Progressive photon mapping. ACM Trans. Graph. (Proceedings SIGGRAPH Asia 2008) 27, 130:1–130:8.
Hachisuka, T., Pantaleoni, J., Jensen, H.W., 2012. A path space extension for robust light transport simulation. ACM Trans. Graph. 31, 191.
Hanrahan, P., Salzman, D., Aupperle, L., 1991. A rapid hierarchical radiosity algorithm. Comput. Graph. (ACM) 25, 197–206.
Hattrup, M.P., 1990. Daylighting Practices of the Architectural Industry (Baseline Results of a National Survey).
Havran, V., Herzog, R., Seidel, H.P., 2005. Fast final gathering via reverse photon mapping. Comput. Graph. Forum (Proceedings Eurographics 2005) 24, 323–334.
Heckbert, P.S., 1991. Simulating global illumination using adaptive meshing (PhD Thesis). University of California, Berkeley.
Herzog, R., Havran, V., Kinuwaki, S., Myszkowski, K., Seidel, H.P., 2007. Global illumination using photon ray splatting. Comput. Graph. Forum (Proceedings Eurographics 2007) 26, 503–513.
Higbie, H.H., 1934. Lighting calculations. John Wiley & Sons, New York.
Hopkinson, R.G., Longmore, J., Petherbridge, P., 1954. An empirical formula for the computation of the indirect component of the daylight factor. Trans. Illum. Eng. Soc. 19, 201–219. https://doi.org/10.1177/147715355401900701.
Humphreys, G., Ananian, C.S., 1996. Tigershark: a hardware accelerated ray-tracing engine. New Jersey, United States.
Ibarra, D.I., Reinhart, C.F., 2009. Daylight factor simulations-How close do simulation beginners “really” get? 11th Int. IBPSA Conf. Build. Simul. 2009, 196–203.
IBPSA, 2019. Software Listing | Building Energy Software Tools (BEST) Directory [WWW Document]. URL https://www.buildingenergysoftwaretools.com/ (accessed 6.20.19).
IES, 2012. Approved Method: IES Spatial Daylight Autonomy (sDA) and Annual Sunlight Exposure (ASE). Illuminating Engineering Society, New York.
Igawa, N., Shimasaki, S., Nakamura, H., 1999. A study on estimation of illuminance by irradiance. J. Archit. Planning, Environ. Eng. (in Japanese) 526, 17–24.
Immel, D.S., Cohen, M.F., Greenberg, D.P., 1986. A radiosity method for non-diffuse environments. ACM SIGGRAPH Comput. Graph. 20, 133–142.
Intel, 2018. New Intel Architectures and Technologies Target Expanded Market Opportunities | Intel Newsroom [WWW Document]. URL https://newsroom.intel.com/news/new-intel-architectures-technologies-target-expanded-market-opportunities/#gs.25u0tc (accessed 6.12.19).
Iversen, A., Roy, N., Hvass, M., Jørgensen, M., Christoffersen, J., Osterhaus, W., Johnsen, K., 2013. Daylight Calculations in Practice: An Investigation of the Ability of Nine Daylight Simulation Programs to Calculate the Daylight Factor in Five Typical Rooms.
Jakica, N., 2018. State-of-the-art review of solar design tools and methods for assessing daylighting and solar potential for building-integrated photovoltaics. Renew. Sustain. Energy Rev. 81, 1296–1328. https://doi.org/10.1016/j.rser.2017.05.080.
Jakob, W., Marschner, S.R., 2012. Manifold Exploration: a Markov Chain Monte Carlo technique for rendering scenes with difficult specular transport. ACM Trans. Graph. 31, 58.
Jarvis, D., Donn, M., 1997. Comparison of Computer and Model Simulations of a Daylit Interior with Reality. In: Proceedings of the 5th Conference of International Building Performance Simulation Association (IBPSA). Prague, Czech Republic.
Jensen, H.W., 2001. Realistic Image Synthesis Using Photon Mapping. A. K. Peters, Massachusetts, United States.
Jensen, H.W., 1997. Rendering caustics on non-Lambertian surfaces. Comput. Graph. (ACM) 16, 57–64.
Jensen, H.W., 1996. Global illumination using photon maps. In: Eurographics Rendering Workshop. pp. 21–30.
Jensen, H.W., 1995. Importance driven path tracing using the photon map. In: Eurographics Rendering Workshop. pp. 326–335.
Jensen, H.W., Christensen, N.J., 2000. A practical guide to global illumination using photon maps.
Jensen, H.W., Marschner, S.R., Levoy, M., Hanrahan, P., 2001. A Practical Model for Subsurface Light Transport. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH). California, United States, pp. 511–518.
Jones, N.L., Reinhart, C.F., 2016. Real-time visual comfort feedback for architectural design. In: the 32nd International Conference on Passive and Low Energy Architecture (PLEA 2016). California, United States, pp. 659–664.
Jones, N.L., Reinhart, C.F., 2015. Fast daylight coefficient calculation using graphics hardware. In: 14th Conference of International Building Performance Simulation Association, Hyderabad, India, December 7–9, 2015. pp. 1237–1244.
Kaempf, J., Paule, B., Basurto, C., Bodart, M., de Boer, J., Bueno, B., Dubois, M.C., Geisler-Moroder, D., Fusco, M., Hegi, M., Jorgensen, M., Roy, N., Wienold, J., 2016. Methods and Tools for Lighting Retrofits - State of the Art Review.
Kajiya, J.T., 1986. The Rendering Equation. ACM SIGGRAPH Comput. Graph. 20, 143–150.
Kaplanyan, A.S., Dachsbacher, C., 2013. Path space regularization for holistic and robust light transport. In: Computer Graphics Forum (Vol. 32, No. 2). pp. 63–72.
Kato, T., 2002. The Kilauea massively parallel ray tracer. In: Chalmers, A., Davis, T., Reinhard, E. (Eds.), Practical Parallel Rendering. A K Peters.
Kay, T.L., Kajiya, J.T., 1986. Ray tracing complex scenes. Comput. Graph. (ACM) 20, 269–278.
Kazanasmaz, T., Günaydin, M., Binol, S., 2009. Artificial neural networks to predict daylight illuminance in office buildings. Build. Environ. 44, 1751–1757.
Kelemen, C., Szirmay-Kalos, L., Antal, G., Csonka, F., 2002. A simple and robust mutation strategy for the metropolis light transport algorithm, in: In Computer Graphics Forum (Vol. 21, No. 3). pp. 531–540.
Keller, A., 2013. Quasi-Monte Carlo image synthesis in a nutshell, in: Dick, J., Kuo, F.Y., Peters, G.W., Sloan, I.H. (Eds.), Monte Carlo and Quasi-Monte Carlo Methods. Springer, Berlin, Heidelberg, pp. 213–249.
Keller, A., 1997. Instant Radiosity. In: Computer Graphics (Proceedings of SIGGRAPH 1997). pp. 49–56.
Keller, A., Karras, T., Wald, I., Aila, T., Laine, S., Bikker, J., Gribble, C., Lee, W.J., McCombe, J., 2013. Ray tracing is the future and ever will be.... In: ACM SIGGRAPH 2013 Courses. p. 9.
Keller, A., Wächter, C., Raab, M., Seibert, D., van Antwerpen, D., Korndörfer, J., Kettner, L., 2017. The Iray Light Transport Simulation and Rendering System. In: ACM SIGGRAPH 2017 Talks. ACM. p. 34.
Kenny, P., Lewis, J.O., 1995. Tools and Techniques for the Design and Evaluation of Energy Efficient Buildings. EC DG XVII Thermie Action No B 184, Energy Research Group, University College Dublin.
Kensek, K., Suk, J.Y., 2011. Daylight Factor (overcast sky) versus Daylight Availability (clear sky) in computer-based daylighting simulations. J. Creat. Sustain. Archit. Built Environ. 1, 3–14.
Kim, C.S., Chung, S.J., 2011. Daylighting simulation as an architectural design process in museums installed with toplights. Build. Environ. 46, 210–222.
Kirimtat, A., Koyunbaba, B.K., Chatzikonstantinou, I., Sariyildiz, S., 2016. Review of simulation modeling for shading devices in buildings. Renew. Sustain. Energy Rev. 53, 23–49.
Knaus, C., Zwicker, M., 2011. Progressive photon mapping: a probabilistic approach. ACM Trans. Graph. 30, 25.
Kollig, T., Keller, A., 2000. Efficient bidirectional path tracing by randomized quasi-Monte Carlo integration, in: Monte Carlo and Quasi-Monte Carlo Methods. Springer-Verlag, pp. 290–305.
Koomey, J.G., Berard, S., Sanchez, M., Wong, H., 2011. Implications of historical trends in the electrical efficiency of computing. IEEE Ann. Hist. Comput. 33, 46–54. https://doi.org/10.1109/MAHC.2010.28.
Kota, S., Haberl, J.S., 2009. Historical Survey of Daylighting Calculations Methods and their Use in Energy Performance Simulations. In: Proceedings of the 9th International Conference for Enhanced Building Pperations. Texas, United States, pp. 9–18.
Kuchkuda, R., 1988. An Introduction to Ray Tracing. Theor. Found. Comput. Graph. CAD. https://doi.org/10.1007/978-3-642-83539-1_44.
Labayrade, R., Jensen, H.W., Jensen, C.W., 2010. An iterative workflow to assess the physical accuracy of lighting simulation programs. Light Eng. 18, 60–70.
Lafortune, E.P., Foo, S.C., Torrance, K.E., Greenberg, D.P., 1997. Non-linear approximation of reflectance functions. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques. ACM Press/Addison-Wesley Publishing Co., pp. 117–126.
Lafortune, E.P., Willems, Y.D., 1993. Bi-directional path tracing. In: Computer Graphics. Alvor, Portugal, pp. 145–153.
Laine, S., Saransaari, H., Kontkanen, J., Lehtinen, J., Aila, T., 2007. Incremental instant radiosity for real-time indirect illumination. In: Proceedings of Eurographics Symposium on Rendering. Eurographics Association, pp. 277–286.
Lee, E.S., Geisler-Moroder, D., Ward, G.J., 2018. Modeling the direct sun component in buildings using matrix algebraic approaches: Methods and validation. Sol. Energy 160, 380–395.
Lehtinen, J., Karras, T., Laine, S., Aittala, M., Durand, F., Aila, T., 2013. Gradient-domain metropolis light transport. ACM Trans. Graph. 32, 95.
Lesev, H., 2010. Classification of global illumination algorithms. In: Proceedings of the Anniversary International Conference on Research and Education in Mathematics,

Informatics and Their Applications (REMIA2010). University Press “Paisii Hilendarski”, Plovdiv, Bulgaria, pp. 271–278.

Levermore, G.J., Parkinson, J.B., 2006. Analyses and Algorithms for new test reference years and design summer years for the UK. Build. Serv. Eng. Res. Technol. 27, 311–325.

Li, D.H., Lau, C.S., Lam, J.C., 2004. Predicting daylight illuminance by computer simulation techniques. Light. Res. Technol. 36, 113–129. https://doi.org/10.1191/1365782804li108oa.

Lischinski, D., Tampieri, F., Greenberg, D.P., 1993. Combining hierarchical radiosity and discontinuity meshing. In: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1993). pp. 199–208.

Lischinski, D., Tampieri, F., Greenberg, D.P., 1992. Discontinuity meshing for accurate radiosity. Comput. Graph. (ACM) 12, 25–39.

Littlefair, P.J., 1989. Predicting Hourly Internal Daylight Illuminances for Dynamic Building Energy Modelling. Garston, Building Environmental Performance Analysis Club (BEPAC), United Kingdom.

Maamari, F., 2004. International Survey on Lighting Simulation Tools. 2004.

Maamari, F., Fontoynont, M., 2003. Analytical tests for investigating the accuracy of lighting programs. Light. Res. Technol. 35, 225–239.

Maamari, F., Fontoynont, M., Adra, N., 2006. Application of the CIE test cases to assess the accuracy of lighting computer programs. Energy Build. 38, 869–877. https://doi.org/10.1016/j.enbuild.2006.03.016.

Mangkuto, R.A., 2016. Validation of DIALux 4.12 and DIALux evo 4.1 against the Analytical Test Cases of CIE 171: 2006. Leukos 12, 139–150.

Mardaljevic, J., 2004. Verification of program accuracy for illuminance modelling: assumptions, methodology and an examination of conflicting findings. Light. Res. Technol. 36, 217–239.

Mardaljevic, J., 2001. The BRE-IDMP dataset: a new benchmark for the validation of illuminance prediction techniques. Trans. Illum. Eng. Soc. 33, 117–134.

Mardaljevic, J., 2000. Simulation of annual daylighting profiles for internal illuminance. Light. Res. Technol. 32, 111–118. https://doi.org/10.1177/096032710003200302.

Mardaljevic, J., 1997. Validation of a lighting simulation program: a study using measured sky brightness distributions. In: Proceedings of the 8th European Lighting Conference. Amsterdam, pp. 555–569.

Mardaljevic, J., 1995. Validation of a lighting simulation program under real sky conditions. Light. Res. Technol. 27, 181–188. https://doi.org/10.1177/14771535950270040701.

McNeil, A., Jonsson, J., Appelfeld, D., Ward, G.J., Lee, E.S., 2013. A validation of a ray-tracing tool used to generate bi-directional scattering distribution functions for complex fenestration systems. Sol. Energy 98, 404–414.

McNeil, A., Lee, E.S., 2012. A validation of the Radiance three-phase simulation method for modelling annual daylight performance of optically complex fenestration systems. J. Build. Perform. Simul. 6, 24–37. https://doi.org/10.1080/19401493.2012.671852.

McNeil, A., Lee, E.S., Jonsson, J.C., 2017. Daylight performance of a microstructured prismatic window film in deep open plan offices. Build. Environ. 113, 280–297.

McNicholl, A., Lewis, J.O., 1994. Daylighting in Buildings. EC DG XVII Thermie, Energy Research Group, University College, Dublin.

Microsoft, 2018. Announcing Microsoft DirectX Raytracing | DirectX Developer Blog [WWW Document]. URL https://devblogs.microsoft.com/directx/announcing-microsoft-directx-raytracing/ (accessed 6.14.19).

Mitchell, R., Kohler, C., Klems, J., Rubin, M., Arasteh, D., Huizenga, C., Yu, T., Curcija, D., 2006. WINDOW 6.1/THERM 6.1 Research Version User Manual.

Mitsuba, 2010. Mitsuba – physically based renderer [WWW Document]. URL https://www.mitsuba-renderer.org (accessed 6.10.19).

Moazami, A., Nik, V.M., Carlucci, S., Geving, S., 2019. Impacts of future weather data typology on building energy performance–Investigating long-term patterns of climate change and extreme weather conditions. Appl. Energy 238, 696–720.

Modest, M.F., 1982. A general model for the calculation of daylighting in interior spaces. Energy Build. 5, 69–79. https://doi.org/10.1016/0378-7788(82)90030-5.

Moon, P., Spencer, D.E., 1948. Lighting Design. Addison-Wesley Press, Massachusetts.

Moore, G.E., 1965. Cramming more components onto integrated circuits. Electronics 38, 114–117.

Muneer, T., Angus, R.C., 1994. Daylight illuminance factors based on real sky distributions. In: Proceedings of the CIBSE National Lighting Conference. London, United Kingdom, pp. 260–266.

Muneer, T., Kinghorn, D., 1998. Luminous efficacy models – evaluation against UK data. J. Illum. Eng. Soc. 27, 163–170.

Murdoch, J.P., 1985. Illumination Engineering: From Edison’s Lamp to the Laser. Collier Macmillan Publishers, New York, United States.

Muuss, M.J., 1987. RT and REMRT-shared memory parallel and network distributed ray-tracing programs. In: Proceedings of the Fourth Computer Graphics Workshop. p. 42:16.

Nah, J.H., Kwon, H.J., Kim, D.S., Jeong, C.H., Park, W.C., Han, T.D., Manocha, D., Park, W.C., 2014. Raycore: A ray-tracing hardware architecture for mobile devices. ACM Trans. Graph. 33, 162:1–162:15.

Nasrollahi, N., Shokri, E., 2016. Daylight illuminance in urban environments for visual comfort and energy performance. Renew. Sustain. Energy Rev. 66, 861–874.

Navarro, C.A., Hitschfeld-Kahler, N., Mateu, L., 2014. A survey on parallel computing and its applications in data-parallel problems using GPU architectures. Commun. Comput. Phys. 15, 285–329.

Ng, E., 2001. A Study of the Accuracy of Daylighting Simulation of Heavily obstructed Buildings in Hong Kong. In: Proceedings of the 7th Conference of the International Building Performance Simulation Association (IBPSA). Rio de Janeiro, Brazil, pp. 1215–1222.

Ng, E.Y., Poh, L.K., Wei, W., Nagakura, T., 2001. Advanced lighting simulation in architectural design in the tropics. Autom. Constr. 10, 365–379.

Nicodemus, F.E., 1965. Directional reflectance and emissivity of an opaque surface. Appl. Opt. 4, 767–775.

Noback, A., Grobe, L.O., Wittkopf, S., 2016. Accordance of light scattering from design and de-facto variants of a daylight redirecting component. Buildings 6, 30.

Nvidia, 2019. GeForce RTX 20 Series Graphics Cards | NVIDIA GeForce [WWW Document]. URL https://www.nvidia.com/en-us/geforce/20-series/ (accessed 6.18.19).

Nvidia, 2018. NVIDIA RTX platform | NVIDIA Developer [WWW Document]. URL https://developer.nvidia.com/rtx (accessed 6.19.19).

O’Brien, P.F., 1955. Interreflections in rooms by a network method. J. Opt. Soc. Am. 45, 419–424. https://doi.org/10.1364/JOSA.45.000419.

Ochoa, C.E., Aries, M.B.C., Hensen, J.L.M., 2012. State of the art in lighting simulation for building science: a literature review. J. Build. Perform. Simul. 5, 209–233. https://doi.org/10.1080/19401493.2011.558211.

Ochoa, C.E., Aries, M.B.C., Hensen, J.L.M., 2010. Current Perspectives on Lighting Simulation for Building Science. In: Eindhoven, N. (Ed.), Proceedings of the International Building Performance Simulation Association – Netherlands + Flanders (IBPSA-NVL). Eindhoven, Netherlands, pp. 9–17. https://doi.org/http://dx.doi.org/10.1016/bs.adcom.2016.04.001.

Oh, S., Haberl, J.S., 2016a. Origins of analysis methods used to design high-performance commercial buildings: whole-building energy simulation. Sci. Technol. Built Environ. 22, 118–137. https://doi.org/10.1080/23744731.2015.1063958.

Oh, S., Haberl, J.S., 2016b. Origins of analysis methods used to design high-performance commercial buildings: Solar energy analysis. Sci. Technol. Built Environ. 22, 87–106.

Oh, S., Haberl, J.S., 2016c. Origins of analysis methods used to design high-performance commercial buildings: daylighting simulation. Sci. Technol. Built Environ. 22, 107–117.

Osborne, J., 2012. Building a comprehensive dataset for the validation of daylight simulation software, using complex real architecture. Victoria University of Wellington, Wellington, New Zealand.

Parker, S., Bigler, J., Dietrich, A., Friedrich, H., Hoberock, J., Luebke, D., McAllister, D., McGuire, M., Morley, K., Robison, A., Stich, M., 2010. OptiX: a general purpose ray tracing engine. ACM Trans. Graph. 29, 66:1–66, 13.

Parker, S., Martin, W., Sloan, P.J., Shirley, P., Smits, B., Hansen, C., 1999. Interactive ray tracing, In: ACM Symposium on Interactive 3D Graphics. pp. 119–126.

Parkhurst, J., Darringer, J., Grundmann, B., 2006. From single core to multi-core: Preparing for a new exponential. In: IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD. pp. 67–72. https://doi.org/10.1109/ICCAD.2006.320067.

Peddie, J., 2019. Ray tracing: A tool for all. Springer Nature, Cham, Switzerland.

Perez, R., Ineichen, P., Seals, R., Michalsky, J., Stewart, R., 1990. Modeling daylight availability and irradiance components from direct and global irradiance. Sol. Energy 44, 271–289.

Phong, B.T., 1975. Illumination for computer generated pictures. Commun. ACM 18, 311–317.

Reinhart, C., Breton, P.F., 2009. Experimental validation of 3ds Max design 2009 and Daysim 3.0, in: IBPSA 2009 – International Building Performance Simulation Association 2009. pp. 1514–1521.

Reinhart, C.F., 2018. Daylighting Handbook II. Building Technology Press.

Reinhart, C.F., 2011. Daylight Performance Predictions. In: Hensen, J.L., Lamberts, R. (Eds.), Building Performance Simulation for Design and Operation. Spon Press New York, United States, pp. 235–276.

Reinhart, C.F., Andersen, M., 2006. Development and validation of a Radiance model for a translucent panel. Energy Build. 38, 890–904. https://doi.org/10.1016/j.enbuild.2006.03.006.

Reinhart, C.F., Fitz, A., 2006. Findings from a survey on the current use of daylight simulations in building design. Energy Build. 38, 824–835.

Reinhart, C.F., Herkel, S., 2000. The Simulation of annual daylight illuminance distributions-a state-of-the-art comparison of six RADIANCE-based methods. Energy Build. 32, 167–187. https://doi.org/10.1016/S0378-7788(00)00042-6.

Reinhart, C.F., Mardaljevic, J., Rogers, Z., 2006. Dynamic daylight performance metrics for sustainable building design. Leukos 3, 7–31.

Reinhart, C.F., Pierre-Felix, B., 2009. Experimental validation of autodesk® 3ds max® design 2009 and daysim 3.0. LEUKOS. J. Illum. Eng. Soc. North Am. 6, 7–35. https://doi.org/10.1582/LEUKOS.2009.06.01001.

Reinhart, C.F., Walkenhorst, O., 2001. Validation of dynamic RADIANCE-based daylight simulations for a test office with external blinds. Energy Build. 33, 683–697. https://doi.org/10.1016/S0378-7788(01)00058-5.

Ritschel, T., Dachsbacher, C., Grosch, T., Kautz, J., 2012. The state of the art in interactive global illumination. In: Oxford, U.K. (Ed.), Computer Graphics Forum. Blackwell Publishing Ltd, pp. 160–188.

Roy, G.G., 2000. A comparative study of lighting simulation packages suitable for use in architectural design. Sch. Eng. Murdoch Univ.

Rushmeier, H.E., Torrance, K.E., 1990. Extending the radiosity method to include specularly reflecting and translucent materials. ACM Trans. Graph. 9, 1–27. https://doi.org/10.1145/77635.77636.

Saxena, M., Ward, G.J., Perry, T., Heschong, L., Higa, R., 2010. Dynamic Radiance–Predicting annual daylighting with variable fenestration optics using BSDFs. In: Proceedings of the Fourth National Conference of IBPSA-USA. New York, pp. 402–409.

Schardl, T.B., 2016. Performance engineering of multicore software : developing a science of fast code for the post-Moore era (PhD Thesis). Massachusetts Institute of Technology.

Schmittler, J., Wald, I., Slusallek, P., 2002. SaarCOR: a hardware architecture for ray tracing. In: Proceedings of the ACM SIGGRAPH EUROGRAPHICS Conference on

Graphics Hardware. Eurographics Association, pp. 27–36.
Schregle, R., 2003. Bias compensation for photon maps, in: In Computer Graphics Forum. Oxford, UK and Boston, USA: Blackwell Publishing, Inc, pp. 729–742. https://doi.org/10.1111/j.1467-8659.2003.00720.x.
Schregle, R., Grobe, L.O., Wittkopf, S., 2016. An out-of-core photon mapping approach to daylight coefficients. J. Build. Perform. Simul. 9, 620–632. https://doi.org/10.1080/19401493.2016.1177116.
Schregle, R., Grobe, L.O., Wittkopf, S., 2015. Progressive photon mapping for daylight redirecting components. Sol. Energy 114, 327–336. https://doi.org/10.1016/j.solener.2015.01.041.
Schregle, R., Wienold, J., 2004. Physical validation of global illumination methods: measurement and error analysis. Comput. Graph. (ACM) 23, 761–781.
Segovia, B., Iehl, J.C., Mitanchey, R., Péroche, B., 2006. Bidirectional instant radiosity. Rendering Techniques 389–397.
Segovia, B., Iehl, J.C., Péroche, B., 2007. Metropolis instant radiosity. In: Computer Graphics Forum (Vol. 26, No. 3). Oxford, UK: Blackwell Publishing Ltd, pp. 425–434.
Seo, D., 2018. Comparative analysis of all-sky luminous efficacy models based on calculated and measured solar radiation data of four worldwide cities. Int. J. Photoenergy 2018, Article ID 8180526, 9 pages.
Sillion, F., Puech, C., 1989. A general two-pass method integrating specular and diffuse reflection. ACM SIGGRAPH Comput. Graph. 23, 335–344. https://doi.org/10.1145/74334.74368.
Smits, B., Arvo, J., Greenberg, D.P., 1994. A clustering algorithm for radiosity in complex environments. In: Proceedings of the 21th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1994). pp. 435–442.
Spanier, J., Gelbard, E.M., 1969. Monte Carlo Principles and Neutron Transport Problems. Addison-Wesley Press, Massachusetts, United States.
Spjut, J., Kensler, A., Kopta, D., Brunvand, E., 2009. TRaX: A multicore hardware architecture for real-time ray tracing. IEEE Trans Comput. Des. Integr. Circuits Syst. 28, 1802–1815.
Subramaniam, S., 2018. Parametric Modeling Strategies for Efficient Annual Analysis of Daylight in Buildings (PhD Thesis). The Pennsylvania State University.
Tregenza, P., Mardaljevic, J., 2018. Daylighting buildings: Standards and the needs of the designer. Light. Res. Technol. 50, 63–79.
Tregenza, P.R., 2017. Uncertainty in daylight calculations. Light. Res. Technol. 49, 829–844. https://doi.org/10.1177/1477153516653786.
Tregenza, P.R., 1994. Daylighting computation: Radiosity method using triangular patches. Light. Res. Technol. 26, 1–7. https://doi.org/10.1177/096032719402600101.
Tregenza, P.R., 1989. Modification of the split-flux formulae for mean daylight factor and internal reflected component with large external obstructions. Light. Res. Technol. 21, 125–128. https://doi.org/10.1177/096032718902100305.
Tregenza, P.R., Waters, I.M., 1983. Daylight coefficients. Light. Res. Technol. 15, 65–71. https://doi.org/10.1177/096032718301500201.
Trotter, A., 1911. Illumination; its distribution and measurement. Nature 88, 72–73. https://doi.org/10.1038/088072a0.
Tsangrassoulis, A., Bourdakis, V., 2003. Comparison of radiosity and ray-tracing techniques with a practical design procedure for the prediction of daylight levels in atria. Renew. Energy 28, 2157–2162. https://doi.org/10.1016/S0960-1481(03)00078-8.
Tsangrassoulis, A., Santamouris, M., Asimakopoulos, D., 1996. Theoretical and experimental analysis of daylight performance for various shading systems. Energy Build. 24, 223–230. https://doi.org/10.1016/S0378-7788(96)00981-4.
Ulbricht, C., Wilkie, A., Purgathofer, W., 2006. Verification of physically based rendering algorithms. In: Oxford, U.K. (Ed.), Computer Graphics Forum. Blackwell Publishing Ltd, pp. 237–255.
Van Den Wymelenberg, K., Inanici, M., 2016. Evaluating a new suite of luminance-based design metrics for predicting human visual comfort in offices with daylight. Leukos 13, 113–138.
Van Den Wymelenberg, K., Inanici, M., 2014. A critical investigation of common lighting design metrics for predicting human visual comfort in offices with daylight. Leukos 10, 145–164.
Vardis, K., 2016. Efficient Illumination Algorithms for Global Illumination in Interactive and Real-Time Rendering. Athens University of Economics & Business, Greece.
Veach, E., Guibas, L.J., 1997. Metropolis light transport. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '97. pp. 65–76.
Veach, E., Guibas, L.J., 1995. Optimally combining sampling techniques for Monte Carlo rendering, in: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques. ACM, pp. 419–428.
Veach, E., Guibas, L.J., 1994. Bidirectional estimators for light transport, in: Eurographics Rendering Workshop. pp. 147–162.
Versage, R., Melo, A.P., Lamberts, R., 2010. Impact of Different Daylighting Simulation Results on the Prediction of Total Energy Consumption, in: Proceedings of the 4th National Conference of the International Building Performance Simulation Association (IBPSA-USA). New York, United States, pp. 1–7.
Vorba, J., 2011. Bidirectional photon mapping, in: Proceedings of the 15th Central European Seminar on Computer Graphics.
Vorba, J., Karlík, O., Šik, M., Ritschel, T., Křivánek, J., 2014. On-line learning of parametric mixture models for light transport simulation. ACM Trans. Graph. 33, 101.
Wald, I., Benthin, C., Slusallek, P., 2003. Interactive global illumination in complex and highly occluded environments, in: In Eurographics Symposium on Rendering: 14th Eurographics Workshop on Rendering. pp. 74–81.
Wald, I., Johnson, G., Amstutz, J., Brownlee, C., Knoll, A., Jeffers, J., Günther, J., Navrátil, P., 2017. Ospray-A CPU ray tracing framework for scientific visualization. IEEE Trans. Vis. Comput. Graph. 23, 931–940.
Wald, I., Slusallek, P., Benthin, C., Wagner, M., 2001. Interactive rendering with coherent ray tracing. Comput. Graph. Forum 20, 153–164.
Wald, I., Woop, S., Benthin, C., Johnson, G., Ernst, M., 2014. Embree: A kernel framework for efficient CPU ray tracing. ACM Trans. Graph. 33, 143:1–143:8.
Waldrop, M.M., 2016. The chips are down for Moore's law. Nat. News 530, 144–147.
Wallace, J.R., Cohen, M.F., Greenberg, D.P., 1987. A two-pass solution to the rendering equation: a synthesis of ray tracing and radiosity methods. ACM SIGGRAPH Comput. Graph. 21, 311–320. https://doi.org/10.1145/37402.37438.
Walter, B., Arbree, A., Bala, K., Greenberg, D.P., 2006. Multidimensional lightcuts. ACM Trans. Graph. 25, 1081–1088.
Walter, B., Fernandez, S., Arbree, A., Bala, K., Donikian, M., Greenberg, D.P., 2005. Lightcuts: a scalable approach to illumination. ACM Trans. Graph. 24, 1098–1107.
Walter, B., Khungurn, P., Bala, K., 2012. Bidirectional lightcuts. ACM Trans. Graph. 31, 59:1–59:11.
Wang, R., Zhou, K., Pan, M., Bao, H., 2009. An efficient GPU-based approach for interactive global illumination. ACM Trans. Graph. (Proceedings SIGGRAPH Asia 2009) 28, 91.
Wang, T., Ward, G.J., Lee, E.S., 2018. Efficient Modeling of Optically-Complex, Non-Coplanar Exterior Shading: Validation of Matrix Algebraic Methods.
Wang, T., Ward, G.J., Lee, E.S., 2017. Validating Radiance methods for parametric analysis of non-coplanar shading system - an update, in: 16th International Radiance Workshop. Portland, Oregon, USA.
Wang, T., Ward, G.J., Lee, E.S., 2016. Validation of F-matrix and six-phase method, in: 15th International Radiance Workshop. Padova, Italy.
Ward, G., Shakespeare, R., 1998. Rendering with Radiance: The Art and Science of Lighting Visualization. Morgan Kaufman. https://doi.org/citeulike-article-id:9944262.
Ward, G.J., 2015. Annual Simulation for Out-of-Plane Shading Systems. In: 14th International Radiance Workshop. Philadelphia, PA, USA.
Ward, G.J., 1994. The RADIANCE lighting simulation and rendering system. In: Proceedings of the 21st annual conference on Computer graphics and interactive techniques - SIGGRAPH '94. ACM Press, Florida, United States. https://doi.org/10.1145/192161.192286.
Ward, G.J., 1992. Measuring and modeling anisotropic reflection. Comput. Graph. (ACM) 26, 265–272.
Ward, G.J., Rubinstein, F.M., 1988. A new technique for computer simulation of illuminated spaces. J. Illum. Eng. Soc. 17, 80–91. https://doi.org/10.1080/00994480.1988.10748710.
Ward, G.J., Rubinstein, F.M., Clear, R.D., 1988. A ray tracing solution for diffuse interreflection. In: Proc. 15th Annu. Conf. Comput. Graph. Interact. Tech. - SIGGRAPH '88 22, 85–92. https://doi.org/10.1145/54852.378490.
Watt, A., 2000. 3D Computer Graphics, 3rd ed. Pearson Education Limited, Harlow, England.
Weghorst, H., Greenberg, D.P., 1984. Improved computational methods for ray tracing. ACM Trans. Graph. 3, 52–69. https://doi.org/10.1145/357332.357335.
Whitted, T., 1980. An improved illumination model for shaded display. Commun. ACM 23, 343–349. https://doi.org/10.1145/358876.358882.
Whittle, J., Jones, M.W., Mantiuk, R., 2017. Analysis of reported error in Monte Carlo rendered images. Vis. Comput. 33, 705–713.
Wong, L., Wong, I.L., 2017. A review of daylighting design and implementation in buildings. Renew. Sustain. Energy Rev. 74, 959–968. https://doi.org/10.1016/j.rser.2017.03.061.
Woop, S., Schmittler, J., Slusallek, P., 2005. RPU: a programmable ray processing unit for realtime ray tracing. ACM Trans. Graph. 24, 434–444.
Wu, Y., Kämpf, J.H., Scartezzini, J.L., 2019. Automated 'Eye-sight' Venetian blinds based on an embedded photometric device with real-time daylighting computing. Appl. Energy 252, 113317.
Yamauti, Z., 1926. The light flux distribution of a system of interreflecting surfaces. J. Opt. Soc. Am. 13, 561–571.
Yoon, Y.B., Jeong, W.R., Lee, K.H., 2014a. Window material daylighting performance assessment algorithm: comparing radiosity and split-flux methods. Energies 7, 2362–2376. https://doi.org/10.3390/en7042362.
Yoon, Y.B., Manandhar, R., Lee, K.H., 2014b. Comparative study of two daylighting analysis methods with regard to window orientation and interior wall reflectance. Energies 7, 5825–5846. https://doi.org/10.3390/en7095825.
Yserentant, H., 1986. On the multi-level splitting of finite element spaces. Numer. Math. 49, 379–412.