INTERACTIVE RENDERING OF COMPLEX FENESTRATION MATERIALS FOR ARCHITECTURAL DAYLIGHTING DESIGN

By

Yu Sheng

A Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Major Subject: COMPUTER SCIENCE

Approved:

_____

Barbara Cutler, Thesis Adviser

Rensselaer Polytechnic Institute
Troy, New York

July 2009
(For Graduation August 2009)

CONTENTS

LIST OF TABLES iii

LIST OF FIGURES iv

ACKNOWLEDGMENT v

ABSTRACT vii

1. INTRODUCTION 1
   - 1.1 Motivation 1
   - 1.2 Complex Fenestration Systems 2

2. RELATED WORK 6
   - 2.1 Architectural Lighting Design Software 6
   - 2.2 Computer Graphics Research on Global Illumination Simulations . . 7
     - 2.2.1 Global Illumination Algorithms 7
     - 2.2.2 Radiosity 8
       - 2.2.2.1 Rendering Equation 8
       - 2.2.2.2 Form Factor 8
       - 2.2.2.3 Hierarchical Radiosity 9
       - 2.2.2.4 Graphics Hardware 11
     - 2.2.3 Other Algorithms 11

3. INTERACTIVE DAYLIGHTING SIMULATION SYSTEM 13
   - 3.1 Sampling the Direct Sun & Sky Illumination 13
   - 3.2 Radiosity for Diffuse Reflections 14
   - 3.3 Factoring Direct and Indirect Illumination 14
   - 3.4 Modeling the Specular Lobes of BTDF data 17
   - 3.5 Rendering Complex Fenestration Systems 18
   - 3.6 Interactive Relighting 19

4. RESULTS 23
   - 4.1 System 23
     - 4.1.1 GUI 23
   - 4.2 Case Study Example Scenes 24

5. System Validation 27
5.1 Rendering Speed 27
5.2 Qualitative Comparison of Visual Results 27
5.3 Quantitative Evaluation of Rendering Accuracy 28
5.4 Quantitative Comparison with Sensors 31
6. CONCLUSIONS 33
7. FUTURE WORK 34
LITERATURE CITED 34

LIST OF TABLES

5.1 Average pixel luminance for Radiance and our algorithm 28

5.2 Example comparison of irradiance sensor data collected from LSV and Radiance: Values of sensors with lowest and highest relative difference at three times of day for March 21. 31

LIST OF FIGURES

1.1 Heliodon 2

1.2 Architectural Studio Project 3

1.3 Prismatic Panel Diagram 4

1.4 Prismatic Panel Simulations 5

3.1 Hybrid Radiosity/Shadow Volumes Method 15

3.2 Laser Cut Panel BTDF Visualization 18

3.3 CFS Rendering of a Small Test Room 20

3.4 CFS Rendering of a Medium-Sized Office Configuration 21

4.1 Prototype Graphical User Interface (GUI) 24

4.2 Curved Residential Design 26

5.1 Qualitative Rendering Comparison 29

5.2 Quantitative Rendering Comparison 30

5.3 Quantitative Rendering Comparison 32

ACKNOWLEDGMENT

First of all, I would like to express my sincere gratitude to my advisor, Professor Barbara Cutler, for her guidance throughout my master study. Her helpful suggestions, important advice and constant encouragement are of the most important factors that I can successfully get my degree. I will never forget those busy days and nights rushing for deadlines together and the generous dinners and rides she provided.

Special thanks to my wife, Jishen Zhao. Her endless love, care, and encouragement have been and will always be my motivation for improvement. Resigning from her Ph.D. study without any hesitation, she has sacrificed her own study to join me from the other part of the Earth. Sharing happiness with her is now part of my everyday life.

I would like to thank my family: my father, my step-mother, my younger sister. Their persistent support and belief in me have been invaluable wealth during my master study at RPI. I also want to thank my uncle, aunt for their great support. I will never forget my crying with my uncle before I set off for USA for the first time. I would like to dedicate this thesis to my mother, who passed away when I was 13 years old. Her kind love, her effort to support the family, and her integrity have been a precious gift to me.

I am grateful to appreciate Professor Marilyne Andersen, Sian Kleindienst, and Jaime Lee from Department of Architecture, Massachusetts Institute of Technology, for their cooperation and continuous providing expert suggestions in daylighting design.

I also wish to express my appreciation to my teammates, Steve Martin, Theodore C. Yapo, Josh Nasman. Their smart ideas and useful suggestions are often the source of my inspiration in research. I would also like to appreciate my labmates, Gehua Yang, Chao Chen, Jacob Becker, Brad King, Avi Kelman, Chris Stuetzle, Eric Smith, Alper Ayvaci, Zhongxian Chen, Matt Turek. Their help in my research and study will benefit my entire life. I would like to thank all my friends during my

stay at Troy. Their company has made my life more exciting and colorful.

Finally, I would like to thank RPI, IBM, and NSF for their generous funding during my master's study.

ABSTRACT

Complex fenestration systems, such as prismatic and laser cut panels, can be used to redirect daylighting and more evenly illuminate interior spaces in a building, reducing the need for electric lighting. However, it is challenging to incorporate these materials into architectural design due to the counter-intuitive behavior of light refraction through these panels.

In this thesis, I present a method for rendering and visualization of complex fenestration materials. A hybrid method of shadow volumes for per-pixel hard shadows for direct illumination and patch-based radiosity for indirect illumination is first proposed for interactive rendering. This rendering technique can relight a mesh with about 1500 surfaces in an interactive speed, which makes it possible for architects to use in the schematic architectural design process. This algorithm enables simulation of the direct and indirect illumination from the sun and sky throughout each day for different months of the year. The rendering result is validated with Radiance both qualitatively and quantitatively.

A simple method to model 4D Bidirectional Transmission Distribution Function measurement data of complex fenestration materials is then used to simulate complex fenestration systems by leveraging the hybrid rendering method. The user can interactively explore the rendering results for different times and days to select appropriate materials early in the design process.

1. INTRODUCTION

1.1 Motivation

Designing with daylight shows great promise for reducing energy demands of buildings, increasing exposure to natural light, increasing awareness of the outside environment and creating interesting shaped spaces [34, 5, 29, 33, 79]. There has been a resurgence of interest in using environmentally sustainable methods to make use of daylight and take into account time, season, location, and climate. The emergence of complex fenestration system (CFS) has provided an important technology for people to more efficiently design for daylighting by leveraging these factors [72, 39, 48, 40].

But only if daylighting is accounted for early in the design process, when scale, appearance, and adjacencies of an evolving design are explored and critiqued, can it have a significant positive impact on the sustainability of the building. To assist the designer in accounting for the many factors that influence daylight during this crucial schematic design phase, computer simulations and/or studies with physical scale models can be very effective tools and provide useful and quick feedback. An architect may choose to study the interrelations among light, materials, and space in high-quality renderings or photographs whereas an engineer may analyze point-by-point light measurements. The investigation of many scenarios and conditions will be necessary in either case before conclusions can be reached regarding the most successful design choices.

Available tools cover a wide range in the level of complexity and accuracy of information that can be obtained. Traditional sunlight penetration analysis tools such as the *heliodon* are still taught and used during schematic design [25, 56, 7]. A physical mock-up of the design is placed on the heliodon table to provide instantaneous and intuitive qualitative feedback on direct sun penetration as the model is rotated and manipulated (Figure 1.1).

More advanced approaches include the diffuse sky component (daylight) in their analysis as well, either relying on artificial skies (mirror boxes, domes, scanning

methods) for physical models [45, 54, 7] or on virtual sky representations [10, 59].

The heliodon's equivalents in computer simulation are the daily and annual sun-path shadow analysis tools available in software like Sketchup [26] or Ecotect [70]. An illustrative example of a design that did not benefit from a careful sunlight penetration analysis is given in Figure 1.2, whose elegant assembly of curved elements unfortunately allows serious glare issues that could have been prevented with an appropriate preliminary study.

Physically-accurate computer simulations of global illumination, including radiosity and Monte Carlo ray tracing are available in some CAD programs; however, these lighting tools have not made significant inroads into the architecture community for early stage schematic design [21]. Furthermore, most architects are untrained or unaware of how to prepare their models or tune parameters for speedy yet accurate renderings. Thus, daylighting simulation software is *very seldom used in academic or professional practice to inform design.*

1.2 Complex Fenestration Systems

Recently, advanced fenestration products have drawn interest from both architects and manufacturers because they can be used to illuminate the room more evenly. However, their complex interactions with light make it tremendously difficult

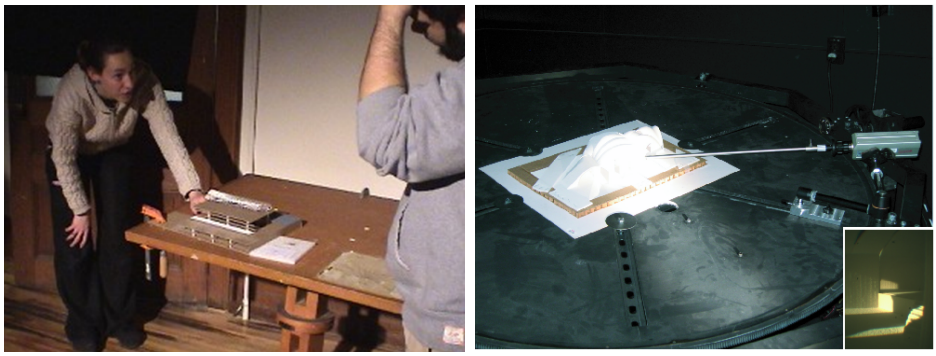

Figure 1.1: The depth of direct beam light penetration in a model is easily ascertained with the *heliodon*. Interior views (right inset) may require endoscopic lenses depending on the model size. The heliodon table rotates along many dimensions relative to a fixed light source (the sun), facilitating exploration.

Figure 1.2: Sunlight penetrates the louvres of this architectural studio project in an unanticipated way, causing discomfort. An interactive daylight renderer could have predicted this problem, allowing the designer to optimize the orientation of the louvres.

to predict what illumination effects will be generated depending on the sun position and sky conditions. Prismatic panels are an example of one such CFS and can be used to redirect illumination deep into the room if the facet orientations are chosen appropriately (Figure 1.3). These panels were popular in the 1890's; however, they fell out of use as electric lighting became available in the early 20th century. The somewhat counter-intuitive behavior of light refraction through these panels can cause confusion and architects are "not certain how to incorporate the prismatic glass into the aesthetic of a building" [67]. Furthermore, studies of occupants in existing buildings with advanced fenestration products reveal additional challenges for adoption: "This system produces not only uneven lighting on wall and working surfaces but also (unexpected, to the client) sharp shadows on the ceiling" [73].

Goniophotometers can be used to measure the Bidirectional Transmission Distribution Function (BTDF) of each fenestration product [1], making it possible to accurately predict (both qualitatively and quantitatively) the behavior of each material in a given building scenario. However, linking material characteristics to space illumination conditions is a difficult task because of the many variables involved and appropriate visualization tools are needed.

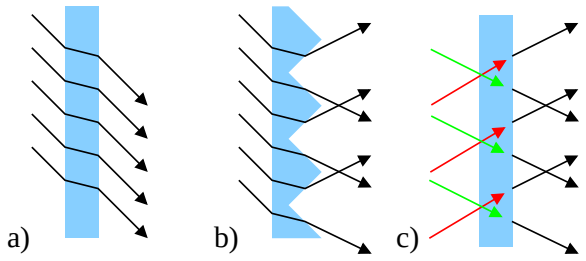

Figure 1.3: a) In a flat pane of glass, light rays are refracted, but exit parallel to the input rays. b) Light rays that pass through a prismatic panel are refracted differently and exit in two different directions due to the microfaceting. c) We can reverse engineer the directions of two "fake suns" allowing us to render the specular refraction in real time (Section 3.4).

The method described in this thesis focuses on efficiently predicting daylighting illumination to evaluate the current design and material selection early in the design process. It is illustrated in Figure 1.4 and includes:

- An *interactive* global illumination system for natural daylighting that is appropriate to the models used in schematic design.
  - Combines computation of direct illumination (shadow volumes) with indirect illumination (radiosity) in a hybrid system.
  - Qualitative and Quantitative validation of our rendering algorithm with Radiance. The results show that our algorithm is accurate enough to be used in schematic daylighting design.
  - Handles non-diffuse BTDF for a variety of specular fenestration materials.
- Demonstration of our system on several case study designs from our architectural collaborators.

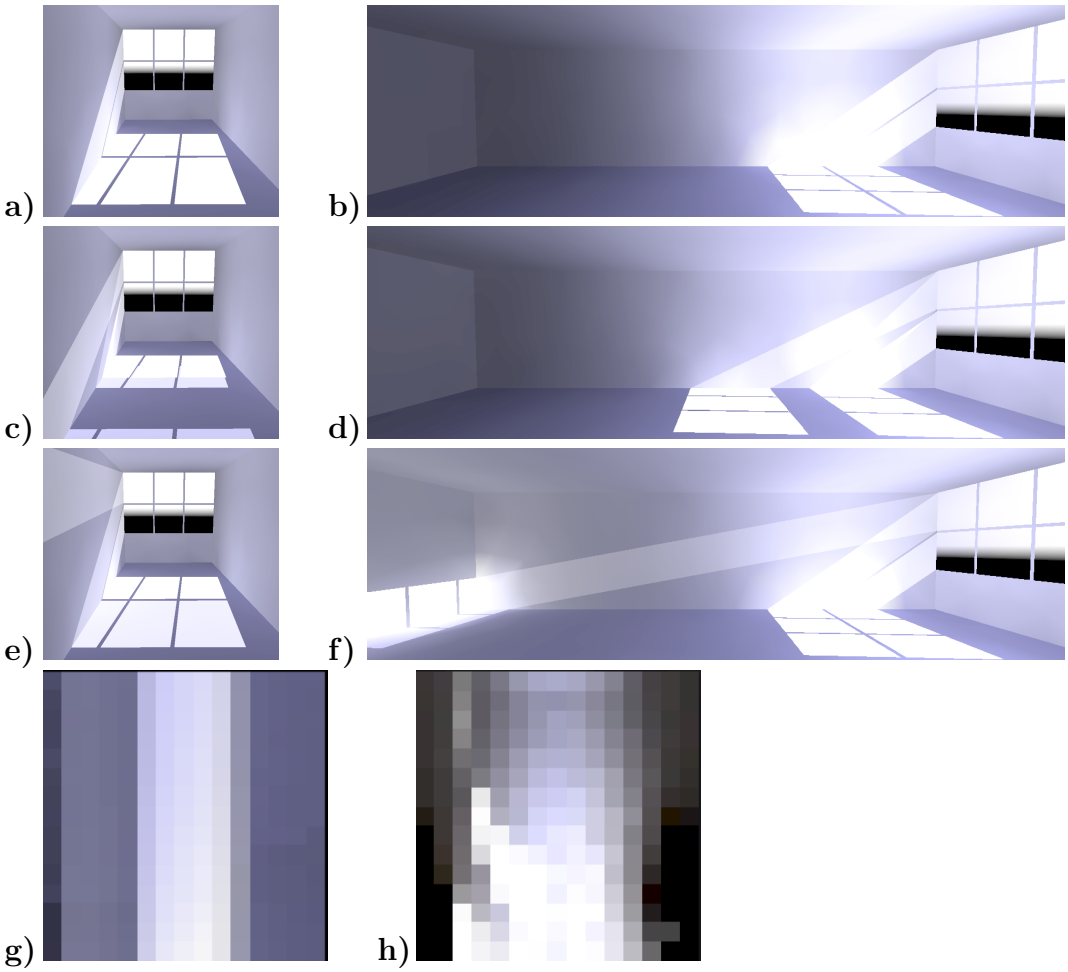

Figure 1.4: Initially all six windows in this simple room scene are filled with a&b) simple planar glass. Note how the far back corner of the room is dark. In c&d) we replace the top 3 panes with a prismatic panel having shallow angles and in e&f) we select a more extreme angle for the upper microfacets in these panels, which sends direct light to the far wall to more uniformly illuminate the room. To optimize the material selection for the center point of the floor at a given time and day of the year, we g) vary the two angular parameters of the prismatic material. Fixing the material, we h) vary the time of day and day of year and calculate the illumination on the center point of the floor.

2. RELATED WORK

2.1 Architectural Lighting Design Software

Architects have many different lighting design software programs to choose from, some are commercially available and others can be downloaded for free. A comprehensive survey [21] of these programs reported that more than 50% of the cited tools were based on the Radiance simulation engine [41, 77, 75]. Unfortunately, most architects do not use this tool frequently because model preparation is non-trivial and appropriate renderings can take from minutes to hours to create. Only expert users seem to have the necessary knowledge about the underlying algorithms to correctly adjust the numerous parameters to produce quick yet accurate and useful images and data. Front-end interfaces such as the "Radiance Control Panel" from Ecotect [70] have partly solved this problem but they generally only offer limited capabilities compared to Radiance.

Ecotect is a recent tool gaining popularity within the building technology community and is making its way into architectural design programs at the university level and in architectural practice. Within Ecotect one can perform lighting calculations with diffuse skies, optimize the shape of exterior shading devices, and perform other environmental evaluations such as thermal and energy analyses. This tool requires some training time and designs must be precisely annotated before analysis and simulation can begin.

A large range of interactive, easy-to-learn tools with a specific focus on early design stages are also found, such as the MIT Design Advisor [46], DIAL-Europe [58], and Daylight1-2-3 [61]. The simulations are limited to simple geometries and, except for Daylight1-2-3, rely on strong simplifying assumptions about the sky model. One of their major drawbacks is that they only provide quantitative information and have very limited visualization capabilities. Not being able to interactively view the illuminated space and quickly explore alternatives with both visual and performance criteria in mind seem to be one of the main reasons why computer simulations of global illumination are rarely used in the early stages of design.

2.2 Computer Graphics Research on Global Illumination Simulations

For the last several decades, global illumination (**GI**) has been one of the most active research areas in computer graphics, and tremendous advances have been made. Considering not only the light coming directly from light sources, but the subsequent light interactions caused by reflections and refraction, **GI** can generate images that are much more photorealistic than direct illumination, and thus are widely used by applications in various fields. Different fields have different requirement for the rendering results. In movie and animation industry, images need to be pretty and realistic, but not necessarily physically accurate. However, in architectural daylighting design, quantitatively accurate lighting results are more important.

2.2.1 Global Illumination Algorithms

Ray tracing [80] was introduced by Turner Whitted in 1980. The capability of simulating specular and refractive effects made ray tracing a very popular rendering method from when it was proposed. Monte Carlo method was also applied to ray tracing to simulate interreflection of light on diffuse, glossy, and specular surfaces to generate a physically accurate global illumination solution. However, it is too slow for practical use. Therefore, most attention was on how to accelerate the rendering speed, such as hierarchical bounding volumes [63] [37], and irradiance caching [76]. Then more research was done to enhance the lighting effects and rendering speed, such as distributed ray tracing [14], photon mapping [35], etc. Leveraging the parallel computing capability of modern graphics hardware, it is now possible to interactively render scenes of moderate complexity [44] [85].

Another popular **GI** method is radiosity. Radiosity has its root in the simulation of thermal radiation in mechanical engineering [69], and was extended to computer graphics by researchers from Cornell University [27] in 1984. In contrast to ray tracing, radiosity is a scene-based and view independent algorithm. In radiosity, the scene is subdivided into patches and radiance functions across patches are computed. During the rendering process, radiosity can make use of graphics hard-

ware. In addition, view is not used to limit the scope of solution. A recomputation of the solution is not needed when the view is changed.

2.2.2 Radiosity

2.2.2.1 Rendering Equation

The rendering equation, one of the most important concepts of **GI**, was first introduced to the research field of computer graphics by Kajiya in 1986 [36]. It shows that the radiance at a point in certain direction $\vec{\omega}'$ consists of two parts, one is the emitted illumination and the other is indirect illumination caused by reflecting and/or refracting lights from other points from direction $\vec{\omega}$ in the scene.

The rendering equation describes how light transports in the physical world. Despite its completeness, it is difficult and expensive to solve. Radiosity simplifies this equation by assuming that all the surfaces in the scene are perfectly Lambertian. A Lambertian surface reflects incoming light evenly to every possible direction in the surrounding hemisphere. Each surface in the scene is partitioned into rectangular or triangular patches. Then, the classical radiosity equation specifies the diffuse light transportation between patches:

$$B_i = E_i + \rho_i \sum_{j=1}^{n} B_j F_{ij} \tag{2.1}$$

Here B_i is the radiosity for each patch in the scene, E_i is the self-emissive light intensity of patch j, ρ_i is the reflectivity of patch i, and F_{ij} is the form factor which represents the fraction of energy leaving i and arriving directly at j, given by

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} G(\mathbf{x}, \mathbf{x}') V(\mathbf{x}, \mathbf{x}') dA_j dA_i \tag{2.2}$$

2.2.2.2 Form Factor

Computing the form factor for equation 2.2 is the most expensive part in solving equation 2.1, and normally covers 90% [66] of the entire computation time. Of the computation, visibility is the most time-consuming part. The complexity of naive visibility testing is $O(n^3)$. Implicit visibility evaluation becomes a promising direction to avoid the intensive visibility computation. One popular method to re-

solve visibility is the hemicube proposed by Cohen [13]. This method makes use of graphics hardware to accelerate the computation. Dachsbacher [18] proposed a **GI** algorithm which solves the implicit visibility problem by reformulating the rendering equation and propagating negative lights (called *anti-radiance*) to compensate for the light transport that incorrectly traverses occlusions. This method achieves interactive rendering speed for moderately complex scene with moving objects and lights. Dong [20] also developed an interactive **GI** algorithm to implicitly compute visibility. This method build the link structure between scene elements in the same way as traditional hierarchical radiosity (**HR**) [30] except only storing the link to the closest element, which solves the visibility implicitly. Elements are subdivided if the solid angle seen from one element to the other is too big. Energy is propagated with the same push-pull procedure as traditional **HR**. However, a lot of factors might cause artifacts to the rendering results, such as insufficient discretization of surrounding sphere, uneven distribution of solid angles across bins, etc.

2.2.2.3 Hierarchical Radiosity

The time and space complexity of classical radiosity is $O(n^2)$, where n is the number of surfaces in the scene. Therefore, it is impractical to apply the radiosity algorithm to complex scenes with a large number of surfaces. Another research direction for accelerating the radiosity algorithm is to reduce the complexity for solving equation 2.2. Progressive Radiosity [12] was introduced to improve the interactivity of the traditional radiosity algorithm by two changes. One is performing the traditional Gauss-Seidel iteration by *shooting* light from elements instead of *gathering*. The other change is choosing the element with the largest *unshot* power to distribute light instead of an arbitrary one. However, this algorithm does not change the complexity of radiosity. In order to reduce the computation time for conventional radiosity, **HR** solves this problem by reducing the number of interactions between elements in the mesh.

The first hierarchical algorithm is *substructuring*, developed by Cohen [11] in 1986. A two-level hierarchy of the mesh is built, one coarse with m patches, and one fine with n ($n >> m$) patches. The final image is rendered with the fine mesh.

The complexity of algorithm is $O(mn)$ with $m << n$, which is a great improvement from $O(n^2)$. Cohen's substructuring algorithm was improved by hierarchical radiosity [30], which was inspired by N-body problem [4] [6]. **HR** builds a multi-level hierarchy mesh by subdividing surfaces into sub-surfaces (called *patches*). In this hierarchical mesh, light can be transported from any patch to a patch in any level. The order notation of **HR** is $(O(s^2) + O(p))$, where s is the number of surfaces in initial mesh, and p is the number of elements in the fine mesh. **HR** has proved to be an effective way to adapt the solution mesh to the radiance value, which both refines rendering details and improves speed. However, the initial linking phase is quadratic in the number of initial surfaces, and thus **HR** only runs fast for coarse initial meshes. The essential reason is that **HR** can only subdivide patches into smaller ones, but cannot group smaller patches into larger ones.

Automatic construction of clustering hierarchies for **HR** were proposed by Smits [68] and Sillion [64] [65], which were called hierarchical radiosity algorithms with clustering (**HRC**). **HRC** builds a complete hierarchy of the scene with a *root* cluster that represents the whole scene itself. Light can be transported from clusters to clusters, clusters to surfaces, and vice versa. The complexity of **HRC** is $O(s \log s + p)$ to $O(s + p)$, where s is the number of initial patches, and p is the number of resulting patches. Substantial follow-up research was done to improve **HR** and **HRC**. Gibson [24] introduced a rapid error-driven refinement and clustering algorithm. Sillion's volume approximation was extended to anisotropic volumes, and the accuracy of energy exchanges was improved. Stamminger [71] proposed a shooting **HR** algorithm to use shooting instead of gathering to eliminate the storage of links in **HR**. It was also proved that the error bound of this algorithm would be larger than gathering **HR** algorithm with a small number of iterations, but would have the same error at convergence. Willmott [82] developed face clustering radiosity to prevent light from being pushed to leaf surfaces by using multi-resolution meshes, which yields sub-linear performance in the number of input surfaces. In addition, analysis and comparison of radiosity and **HR** algorithms are also available. Hasenfratz [31] analyzed all of the available clustering algorithms for **HRC**, and proposed a taxonomy of them. Willmott [81] [83] made an extensive comparison

(error, speed, and memory consumption) between classical radiosity, progressive radiosity, and wavelet radiosity. This paper concluded that wavelet radiosity using the Haar basis is often the fastest for moderately complex geometries. However, wavelet radiosity uses much more memory than other alternative methods.

2.2.2.4 Graphics Hardware

Modern graphics hardware has been used to accelerate global illumination to interactive rates, including: intersection of rays against scene geometry [8]; precomputed light paths for static geometry [43]; GPU-assisted implementation of the hemicube form factor method [15]; and GPU implementation of irradiance and radiance caching [23]. Keller [38] has described an algorithm called instant radiosity that uses hardware to compute radiosity. Instant radiosity performs a Quasi-Monte Carlo particle sampling on the mesh. Hundreds of virtual point light sources are put at those positions allowing direct illumination and shadow maps to approximate the full **GI** solution. Carr [9] developed an algorithm that performs Jacobi iterations on the GPU to solve radiosity with a precomputed form factor matrix on the CPU. Martin et al [52] computed a coarse level **HR** on the CPU, then refined it on the GPU with shadow mapping. Coombe [16] published the first paper that implemented the full progressive radiosity computation, including form factors, visibility, and shooting selection on the GPU. Real-time ($\geq$ 30 frames per second) global illumination rendering of dynamic environments have been achieved using a coarse volumetric sampling grid [55]; however, the potential artifacts from shadow approximations may not yield the accuracy required for architectural applications. Recently, real-time rendering speeds have been achieved by performing imperfect shadow maps on instant radiosity when rendering dynamic geometry and lights [62]. However, in order to use this method, one must tune parameters to generate realistic rendering results, which is difficult for most people other than computer graphics experts.

2.2.3 Other Algorithms

Alternative methods to radiosity exist for modeling indirect lighting such as irradiance caching, which progressively samples and interpolates scene irradiance values [78, 42, 28]. A more complete summary of recent advances for both interactive

and offline global illumination can be found in [19].

Radiosity and irradiance caching are primarily useful for low-frequency elements of scene illumination. While a solution combining radiosity with ray tracing to obtain high-frequency lighting effects such as hard shadows has been described [74] and implemented with shadow mapping [23, 18], we are not aware of other proposals that use our hybrid method of shadow volumes for per-pixel hard shadows of direct sun illumination and patch-based radiosity for sky and indirect illumination.

3. INTERACTIVE DAYLIGHTING SIMULATION SYSTEM

In the following sections we outline the features of our rendering algorithm for schematic architectural daylighting design. Together they efficiently and accurately model the illumination and allow a designer to analyze, evaluate, and optimize his/her design.

3.1 Sampling the Direct Sun & Sky Illumination

Natural illumination in the built environment is provided not only from direct parallel rays of sunlight, but also by omni-directional illumination from the non-uniform, seasonally-varying sky hemisphere. We model the hemispherical distribution and relative intensity of the sky using standard models [10, 59].

We use forward ray tracing to cast rays from the sun and sky through windows in the model to compute the direct illumination for each surface. For the sun, we cast approximately 5,000 total parallel rays through all windows whose normal faces the sun position. For the sky, we choose approximately 5,000 samples on the hemispherical sky and for each ray approximately 50 random samples on all the windows and trace all rays starting from each point on the sky through each point on the windows. The samples are appropriately normalized by area. The direct illumination rays from the sun and sky are traced into the scene and the light is stored with the surface. We make note of which rays came from parallel sunlight rays versus the omni-directional skylight (this information is used in Section 3.3). For a typical relighting event, we cast roughly 255,000 rays through the scene. We use a spatial data structure to ensure that reasonable performance is achieved despite the large number of rays. In order to further improve the computing speed, the rays traced from the sky are cached for each intersected patch. Thus, rays do not need to be traced when relighting for a different time or day for the same geometry.

3.2 Radiosity for Diffuse Reflections

Light from direct sun and sky illumination is distributed via diffuse reflection between surfaces using radiosity. The linear equation 2.1 can be solved using a Gauss-Seidel iterative method. The first two iterations are listed in the following equations:

$$B_i^{(0)} = E_i^{(0)} \tag{3.1}$$

$$B_i^{(1)} = E_i^{(1)} + \rho_i \sum_{j=1}^{N} B_j^{(0)} F_{ij} \tag{3.2}$$

The first equation shows that the initial value of each patch is the self-emissive intensity. The second equation computes the illumination each patch receives directly from the light sources in the scene. At each patch, three scalar quantities are maintained: the direct illumination received from the sun and sky, the indirect illumination received on the face in the previous iteration, and the cumulative illumination reflected from the face.

3.3 Factoring Direct and Indirect Illumination

In most architectural scenes involving daylighting, light transfers due to diffuse reflection from surfaces dominates the indirect lighting. Additionally, hard-edged shadows from the direct sun provide important visual cues that are necessary to understand the aesthetic of the space. Furthermore, the possibility of glare due to high contrast in the illumination values at the direct shadow boundaries must be considered. Per-pixel hard shadows greatly improve the perceived visual quality, but are usually not critical for computing accurate indirect illumination in diffuse-dominant scenes.

With a low-resolution mesh, traditional radiosity is likely to generate unacceptable visual artifacts from the direct illumination, as shown in Figure 3.1a. Traditionally, these artifacts are reduced by either significantly increasing the mesh resolution or employing discontinuity meshing [47]. However, this will also dramatically increase the computation time and is typically applied only for static lighting conditions.

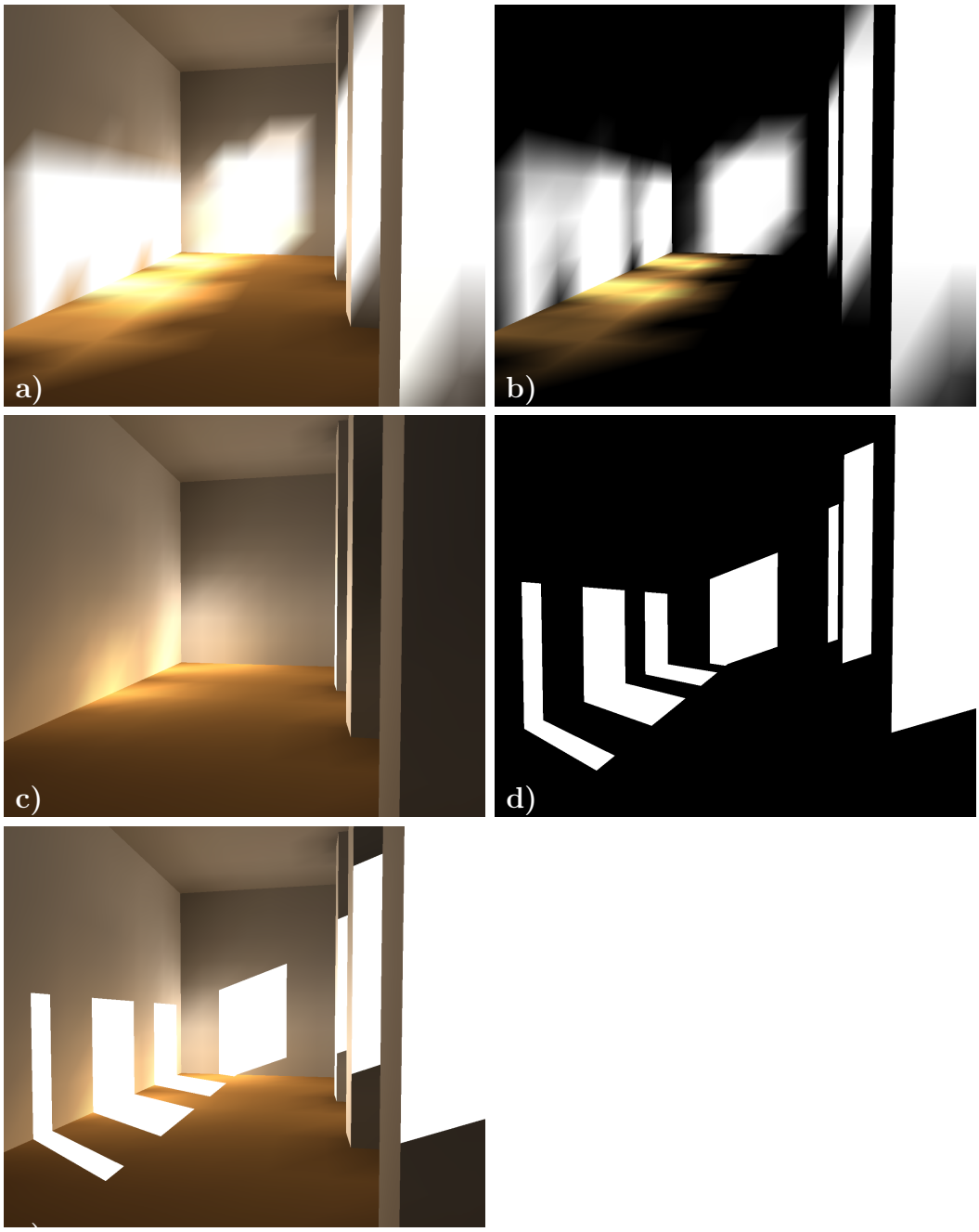

Figure 3.1: The a) classical radiosity solution does not capture hard-edged shadows. We factor the radiosity solution into b) the first bounce direct illumination and c) the indirect illumination by subtracting b) from a). d) Shadow volumes are used to generate per-pixel hard shadows. e) Our hybrid radiosity/shadow volumes rendering is generated by adding c) and d).

In our method, we reduce these artifacts by factoring the radiosity solution into direct illumination (Figure 3.1b) and indirect illumination (Figure 3.1c) and replace the direct illumination by a fast per-pixel rendering method, called *shadow volumes* [17]. In our hybrid radiosity/shadow volumes technique, the radiosity is computed on a coarse per-face basis, while the shadows from direct sunlight are computed at render time on a per-pixel basis (Figure 3.1d). Although the sky contributes significantly to direct illumination, because it produces only soft shadows it is well represented in the original radiosity solution.

The direct illumination from the sun is generated with per-raster-sample computations using multi-pass stencil shadow volumes [17, 32, 53]. This algorithm is chosen because it is supported by almost all graphics hardware, and can achieve real-time speed with a complexity of $O(E)$, where E is the number of window edges plus the number of silhouette edges for a given light direction [84]. We have implemented one variation of this algorithm called "depth-fail" to support rendering shadows when the camera is in the shadow. The pseudo code is listed in algorithm 1.

Algorithm 1 Shadow Volumes Rendering

1: Compute silhouette edges according to current light position
2: Disable writing to depth and color buffer, disable lighting
3: Set stencil operation to increment on depth pass
4: Set back-face culling
5: Extend silhouette edges away from light source, add front and back cap to form a closed mesh, then draw the shadow volume
6: Set front-face culling
7: Set stencil operation to decrement on depth pass
8: Extend silhouette edges away from light source, add front and back cap to form a closed mesh, then draw the shadow volume

To produce the final composite solution, the scene is first rendered using the direct sky illumination combined with total indirect illumination. Next, shadow volume polygons are generated as projections from the window and silhouette edges and rendered to the stencil buffer. Finally, the per-pixel direct illumination from the sun is additively rendered to the frame modulated by the contents of the stencil buffer.

3.4 Modeling the Specular Lobes of BTDF data

CFS, such as prismatic and laser cut panels are advanced window systems that can used to redirect sun and sky light more evenly into the interior part of a building so that people can make more use of daylighting and reduce energy consumption. In a simple panel of window glass, the light exits the window parallel to the incoming direction. However in prismatic and laser cut panels, the light is refracted and reflected within the micro-geometry of the material and can exit the panel in two or more directions, depending on the incoming light direction. However, they are hard to be incorporated into architectural design due to their counter-intuitive reflection and refraction features. We extend the method described in the previous section to model CFS. Currently, our algorithm can only support these materials that can be modeled as a set of one or more specular lobes. A specular lobe means that the reflection is a sharp peek highlight rather than a fuzzy or blurry one. When the exact microfacet geometry of the material is known – for example, the prismatic panel shown in Figure 1.3 – we compute the orientation and relative intensity of the two lobes by tracing a recursive refractive ray through each of the microfacet orientations.

For other materials (such as the laser cut panels shown in Figure 3.2), we use the 4D BTDF measurement data produced at EPFL, Switzerland [2]. For each measured incoming light direction (θ_i, ϕ_i) we greedily select k outgoing directional lobes subtending α degrees that minimize the un-represented transmissive illuminance. For the laser cut panel dataset, we found that k=3 lobes of width α=23 degrees were sufficient to represent 82-100% of the transmitted outgoing illuminance (Figure 3.2).

When rendering illumination from an arbitrary incoming direction, we locate the three nearest measured incoming directions (θ_{i0}, ϕ_{i0}), (θ_{i1}, ϕ_{i1}), and (θ_{i2}, ϕ_{i2}) using a Delauney triangulation. Then we rank and correspond the lobes between each measured direction by decreasing intensity. Finally, the lobe direction and brightness is linearly interpolated for the queried direction. We have found this simple correspondence method to be sufficient for interpolation of a variety of specular BTDF data. However, we anticipate that this simplistic method may result in in-

Algorithm 2 Algorithm for the greedy selection of lobes that minimizes unrepresented illumination

INPUT: Intensity of each outgoing direction (θ_o, ϕ_o) on the refraction hemisphere
OUTPUT: List *lobe* that contains outgoing direction vectors
for each i=0,. . . ,k-1 **do**
  $maxv = 0$
  for each outgoing direction (θ_o, ϕ_o) **do**
    Compute the illuminance v centered at (θ_o, ϕ_o) with area $\frac{\alpha\pi}{180}$
    if $v > maxv$ **then**
      lobe[i]=(θ_o, ϕ_o)
      $maxv = v$
    end if
  end for
  Zero out the intensities of the region used to compute for *lobe*[i]
end for
Return *lobe*

correct correspondences for some BTDF measurements and plan to implement a more robust correspondence technique in future work.

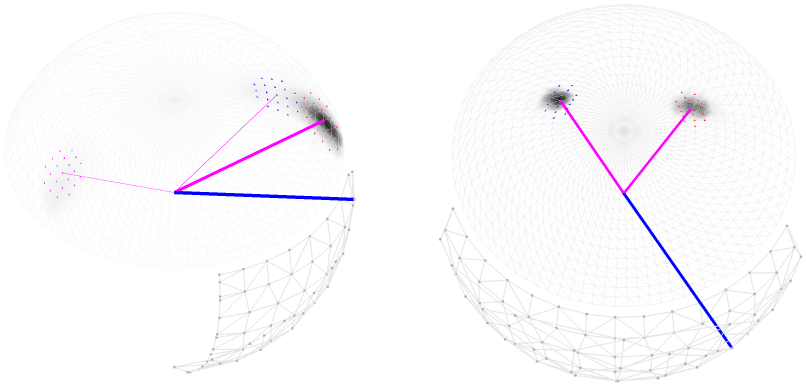

Figure 3.2: Plots of the BTDF data for the laser cut panel CFS for two sample incoming light directions (shown in blue). Up to three lobes are fit to the data to model the specular outgoing illuminance (shown with different widths).

3.5 Rendering Complex Fenestration Systems

Once the lobe directions and intensities are computed, we systematically render each specular reflection from the sun through the different fenestration materials.

As before, we first render the direct sky illumination and total indirect illumination from radiosity. Then we loop through the distinct BTDF materials in the model, and for each of the k specular lobes we position a *fake sun* such that non-refracting rays from the fake sun will be parallel to the outgoing specular lobe (Figure 1.3c) and scale the brightness of the fake sun by the transmittance value. Then we render the stencil buffer shadow volume for the window edges of that material. The number of rendering passes required for a scene with multiple complex fenestration materials is thus $1+2*k*d$ where d is the number of distinct fenestration materials and installation orientations. The detailed algorithm is listed as follows,

Algorithm 3 Multi-pass Shadow Volumes Rendering

```
1: Assign each material an index for rendering, negative values for Lambertian
   materials, non-negative values for for transparent window material (e.g., plain
   glass or CFS)
2: for each non negative material index do
3:    Generate directional virtual lights for current material
4:    for each directional light do
5:       Draw shadow volumes to set stencil buffer
6:       Draw scene again with stencil buffer settings
7:    end for
8: end for
```

Figures 3.3 and 3.4 demonstrate our rendering method for a variety of CFS. For CFS that include a significant diffuse component, the diffuse radiosity emittance for those window patches can be set appropriately.

3.6 Interactive Relighting

To facilitate schematic architectural design that incorporates, responds to, and embraces daylighting, our system must support efficient, interactive recomputation of illumination when the sun is moved and the sky illumination varies with the time of day, season, and climate. We use progressive radiosity [12] to smoothly interpolate the illumination values as the lighting solution for the new sun and sky direct illumination is computed. When the sun is moved, we first recompute the direct light, and then redistribute the light. On each iteration of the "light shooting" radiosity solution, the updated incremental illumination for the old position is replaced by that

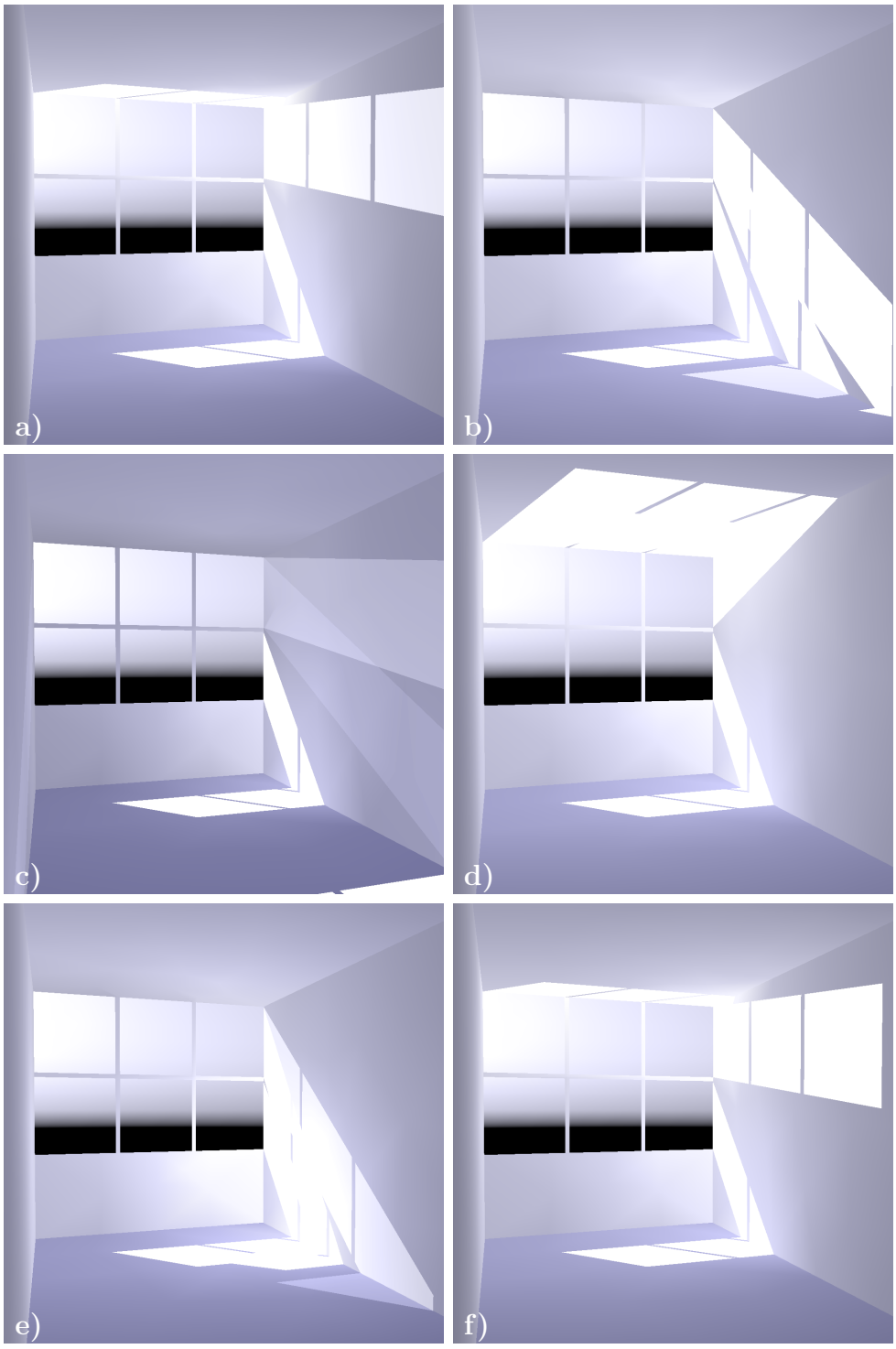

Figure 3.3: Sample CFS renderings in a small test room with south facing windows at 10am on March 21st. The bottom 3 window panes are plain glass in all images. The top 3 panes are: a) laser cut panel, b) optical film (interior), c) perforated blind (open), d) Lumitop^{TM}, e) mirrored Venetian blind, and f) Serraglaze^{TM}.

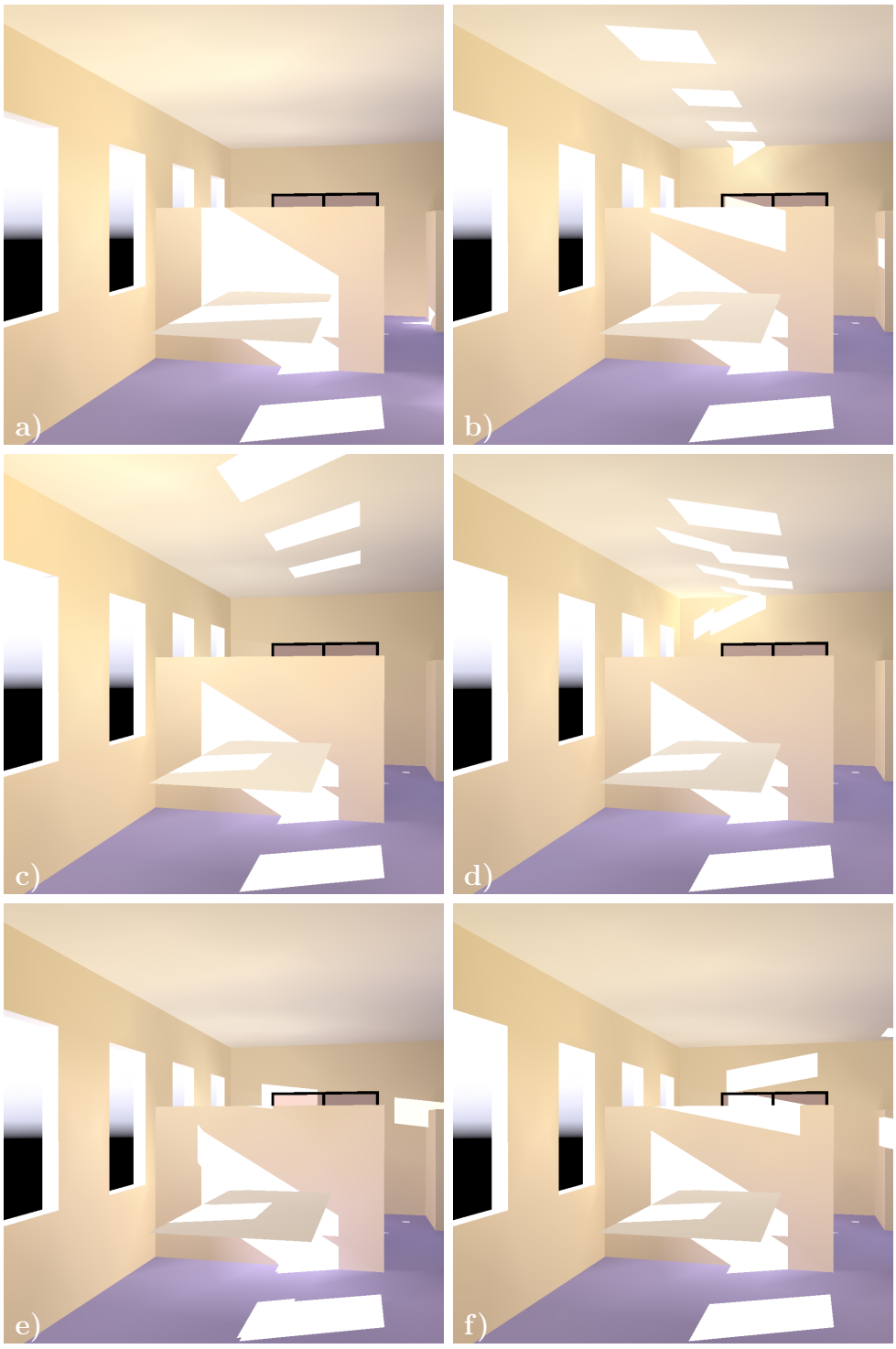

Figure 3.4: Sample CFS rendering in a medium-sized office scene with west facing windows at 3:30pm on March 21st. The lower portion of each window is plain glass in all images. The upper portion of each window is a) plain glass, b) laser cut panel, c) mirrored Venetian blind, d) Lumitop^{TM}, e) optical film (interior), and f) Serraglaze^{TM}.

of the current light position. By separating the direct sun illumination from direct sky illumination and indirect illumination, the system can provide real-time updates of the per-pixel hard shadows as the sun moves. Meanwhile, a separate thread of computation re-calculates at interactive rates the radiosity global illumination data for each face in the mesh. Figure 4.2 shows screen captures of interactive relighting within our system.

4. RESULTS

4.1 System

For our interdisciplinary LightSolve architectural daylighting project [3], we have built an interactive rendering software called *LSV* based on our hybrid radiosity/shadow volumes rendering method and the CFS modeling and rendering technique. This C++ program runs on Linux, FreeBSD, and Windows/Cygwin with the prototype user interface controls shown in Figure 4.1. Similar to standard CAD programs, the 3D model can be viewed from arbitrary camera positions with interactive rotation, translation, and zoom. The time of day, day of year, sky conditions (e.g., clear vs. overcast), and different kinds of CFS can be interactively changed. In addition, the renderings can be saved as images and the configuration data stored and used to create corresponding high-accuracy offline Radiance renderings for result validation.

4.1.1 GUI

The GUI of LSV consists of two windows, one for user interaction, the other for displaying the rendering results. The UI window is built on GLUI user interface library [57]. The window has several panels of options targeted towards both the end user and also the programmer to illustrate the different components of the rendering algorithm for use in validation and debugging. The user control part contains the follow options that users may choose,

Time/day. A scroll bar and a spinner are used to change the time or day. The scroll bar can be used for fast browsing and interactively view the rendering in different time. The spinner can be used to input a specific time.

Exposure. This option corresponds to the exposure setting on a traditional camera and can be used to adjust the overall brightness of the rendering.

Sky type. Current supported sky types are clear, turbid, overcast, uniform, and intermediate. The sky distribution is based CIE standard models.

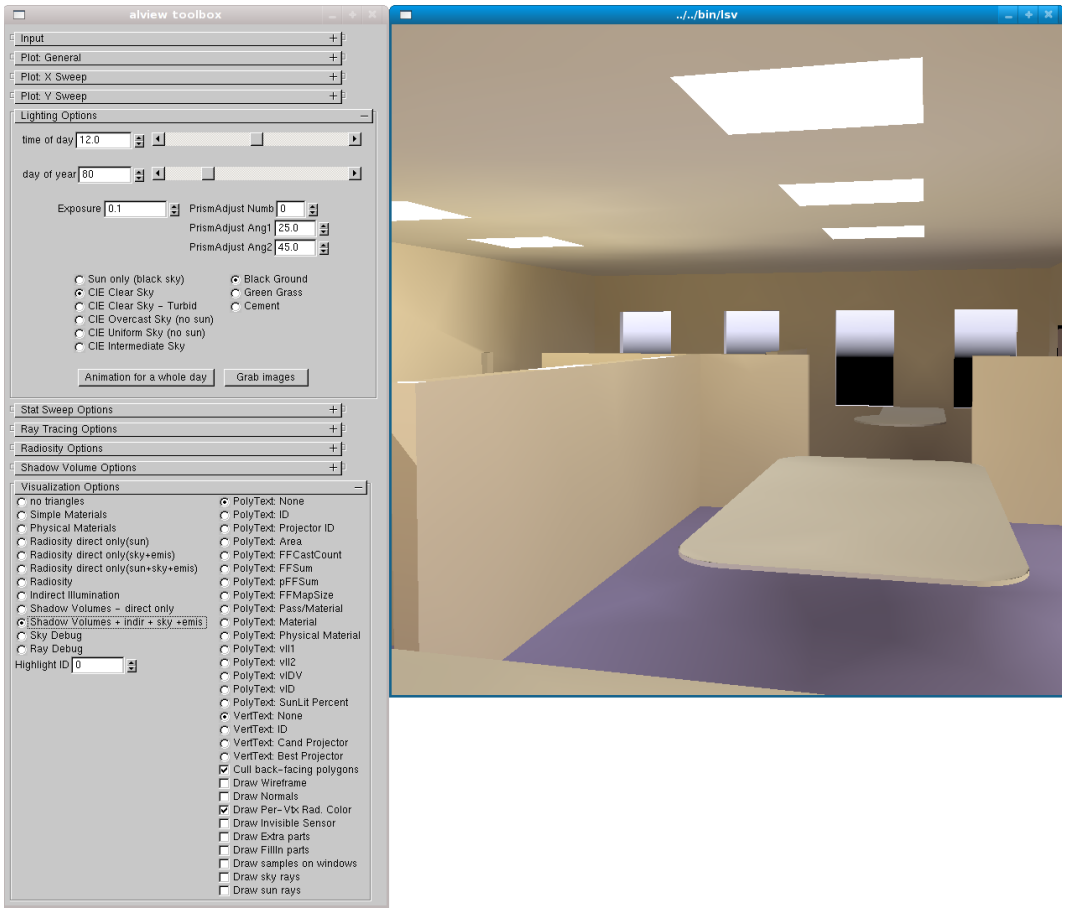

Figure 4.1: Our prototype Graphical User Interface (GUI) includes a control panel and visual display window for photorealistic renderings.

Window type. The user may select window materials from our BTDF database.

Visualization. The user may select to view the direct illumination from sun and sky, indirect illumination separately.

4.2 Case Study Example Scenes

Our results showcase both simple test scenes and more complex real world design situations. The architects we consulted with during the development of our system were quite interested in using daylighting analysis to inform their design processes. Simple test scenes are used in Figures 1.4, 3.1, 3.3.

Figures 3.4, 4.1, and 5.2 show a moderate-sized office environment with low

partition walls and an interesting variety and layout of desk and table furnishings. Few of the desks are close to exterior windows; thus, skylights were used to add illumination to the interior spaces.

The architect of the design shown in Figures 4.2 and 5.1 used curvilinear shapes to bring indirect light into a house and enhance the warm colors of late evening sun. The quantitative lux values received by each patch in the scene are not significantly impacted by the low-frequency soft shadow edges inherent in the radiosity method. However, the qualitative visual improvement that the hard shadows in the hybrid radiosity/shadow volume method bring to the results improves the effectiveness of simulation for daylighting design and prevent mistakes such as those seen in Figure 1.2 and observed by Sweitzer [73].

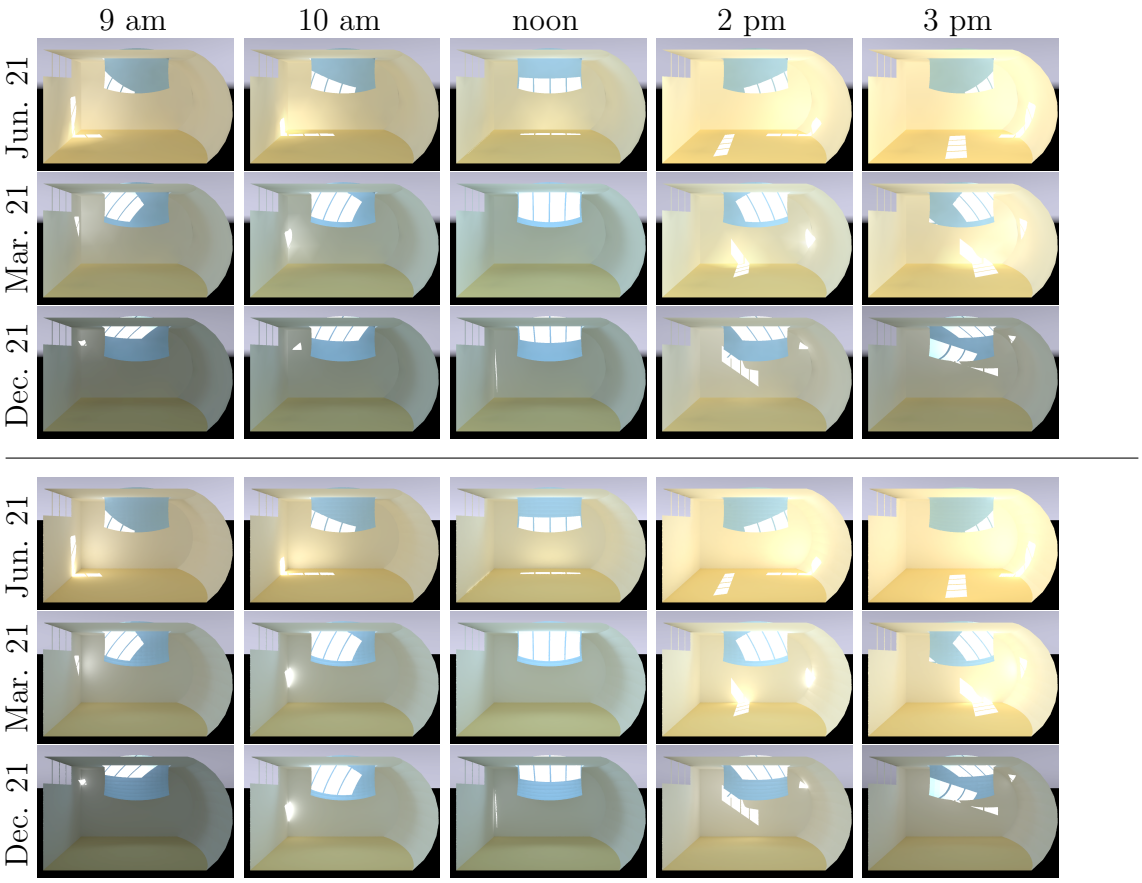

Figure 4.2: In designing the interesting curved geometry for the living space in this residential design, the architect redirects the strong overhead noon sunshine from a set of skylights with a curved diffuse blue deflector but allows the warmer late afternoon sun to penetrate deep into the room and wash over the far wall. Our interactive global illumination relighting system allows him to quickly evaluate this geometry for different sun positions and sky conditions. The top 3 rows of images are rendered with our system. The bottom 3 rows are produced by Radiance.

5. System Validation

For validation of our hybrid radiosity/shadow volumes technique, we compare both the speed and accuracy of our system to Radiance, the accepted industry standard for architectural lighting simulations. We produce two versions of Radiance images for comparison. The first is the *ground truth rendering*, a highly accurate image that is generated by increasing each of the Radiance parameters until the image is constant. The rendering time to create a single image was 45-90 minutes for our test scenes. Next we adjusted the Radiance parameters to produce a *fast rendering*, which sacrifices some accuracy in the rendering, but is much quicker (approximately 5 minutes) and thus more useful in schematic architectural design. The comparison was done on a standard PC, with a Intel Core 2 Duo E6400 CPU (2.13GHz), and 2G memory.

5.1 Rendering Speed

The performance of our prototype implementation is quite compatible with interactive rendering of CFS for daylighting during the schematic design phase of architecture. A model with 1000-3000 triangular patches is loaded into the system with approximately 10 seconds of initialization to compute the form factors for radiosity. Changing the rotation, translation, or zoom of the virtual "camera" can be done in realtime ($>$ 30 frames per second). When the time of day, day of year, or the parameters of the BTDF for the CFS is changed, the per-pixel direct illumination contribution from the sun is updated in real time using shadow volumes and the contribution from the sky and indirect illumination is progressively updated in approximately 1-3 seconds.

5.2 Qualitative Comparison of Visual Results

Figure 4.2 shows a qualitative comparison of images captured from our system and Radiance at different times and days of the year. Difference images for two of these times are shown in Figure 5.1 for visual (qualitative) comparison. The Radi-

ance fast rendering (Figure 5.1a) exhibits locally uneven surface brightness artifacts and the total brightness of the scene is incorrect. In contrast, our hybrid radiosity/shadow volumes technique (Figure 5.1b) matches the ground truth Radiance image (Figure 5.1c) in both smoothness and overall brightness. From the difference image (Figure 5.1d), we observe that the primary source of deviation between the renderings is the secondary bounce when bright direct illumination from the sun light falls in the corner of a room. An adaptive meshing method, such as hierarchical radiosity, which we plan implement in future work, will significantly reduce this source of error. Note that errors in the illumination of the curved skylight redirecting panel are due to a lack of surface normals in the Radiance mesh.

5.3 Quantitative Evaluation of Rendering Accuracy

In lighting design, supplementary fixture selection and placement decisions are made based on predicted numerical illuminance and luminance conditions. Thus we also compare our simulation quantitatively to Radiance, which has been extensively validated for precision use in architectural daylighting design [49, 51, 50, 22]. We calculate the luminance difference pixel by pixel between Radiance and our hybrid radiosity/shadow volumes rendering for both the entire image and an area of interest. The results of a quantitative comparison of the luminance values computed in our system and Radiance is shown in Table 5.1. The margin of error between the systems is less than 10% for a variety of different scenes, camera positions and daylighting conditions (Figure 5.2). Thus our system provides accurate qualitative and quantitative renderings appropriate for use in schematic architectural design.

Figure		Radiance average pixel luminance (cd/m^2)	LSV average pixel luminance (cd/m^2)	average error (%)
5.2a)	entire image	916.40	861.35	8.2
	desks only	1858.06	1748.75	7.4
5.2b)	entire image	1414.77	1320.76	8.6
	area of interest	946.51	828.62	8.8

Table 5.1: Average pixel luminance for Radiance and our algorithm

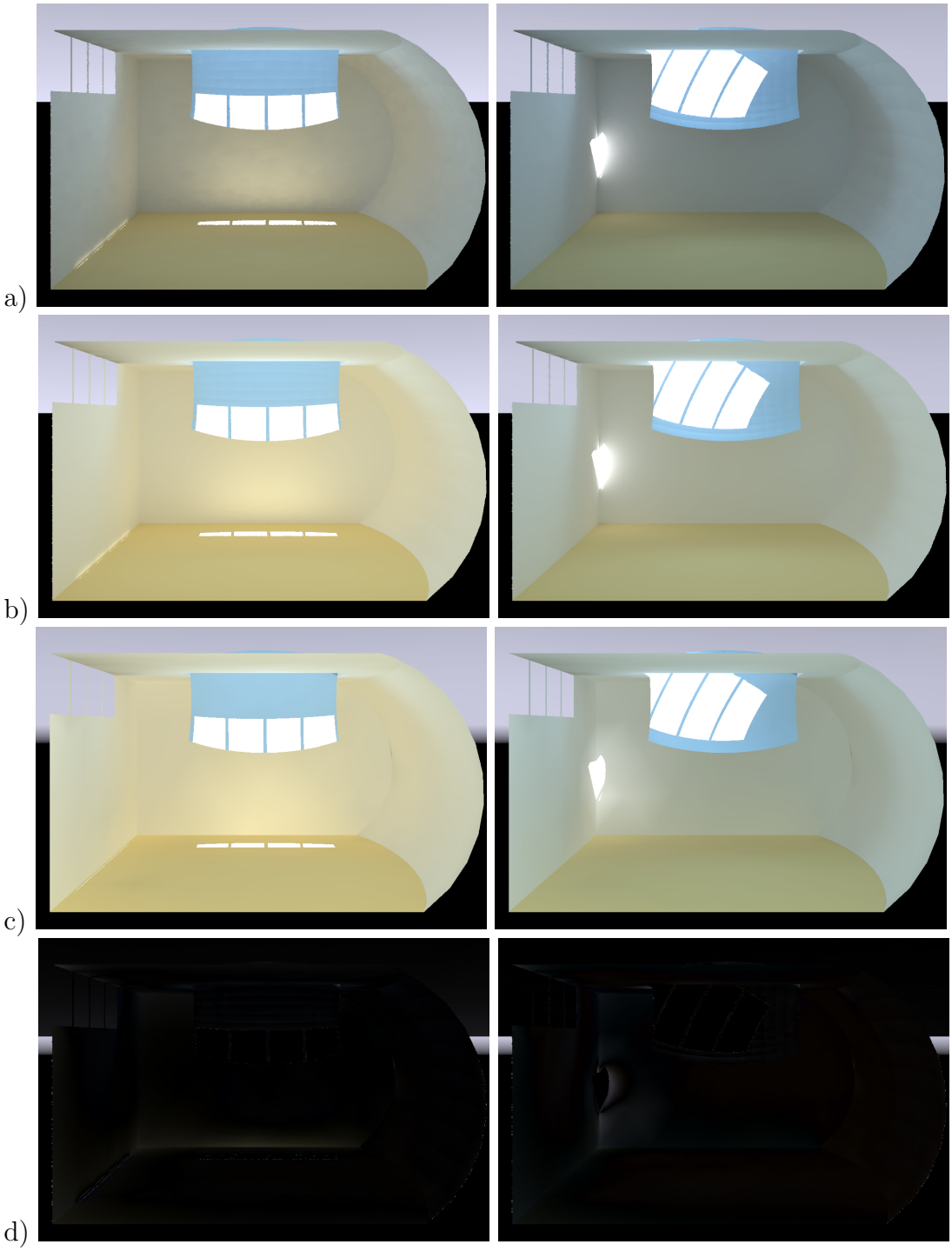

Figure 5.1: Qualitatively our interactive renderings are very similar to Radiance's offline renderings: a) Fast Radiance renderings, b) Ground truth Radiance renderings, c) Our hybrid radiosity/shadow volumes method, and d) Difference images between b) and c).

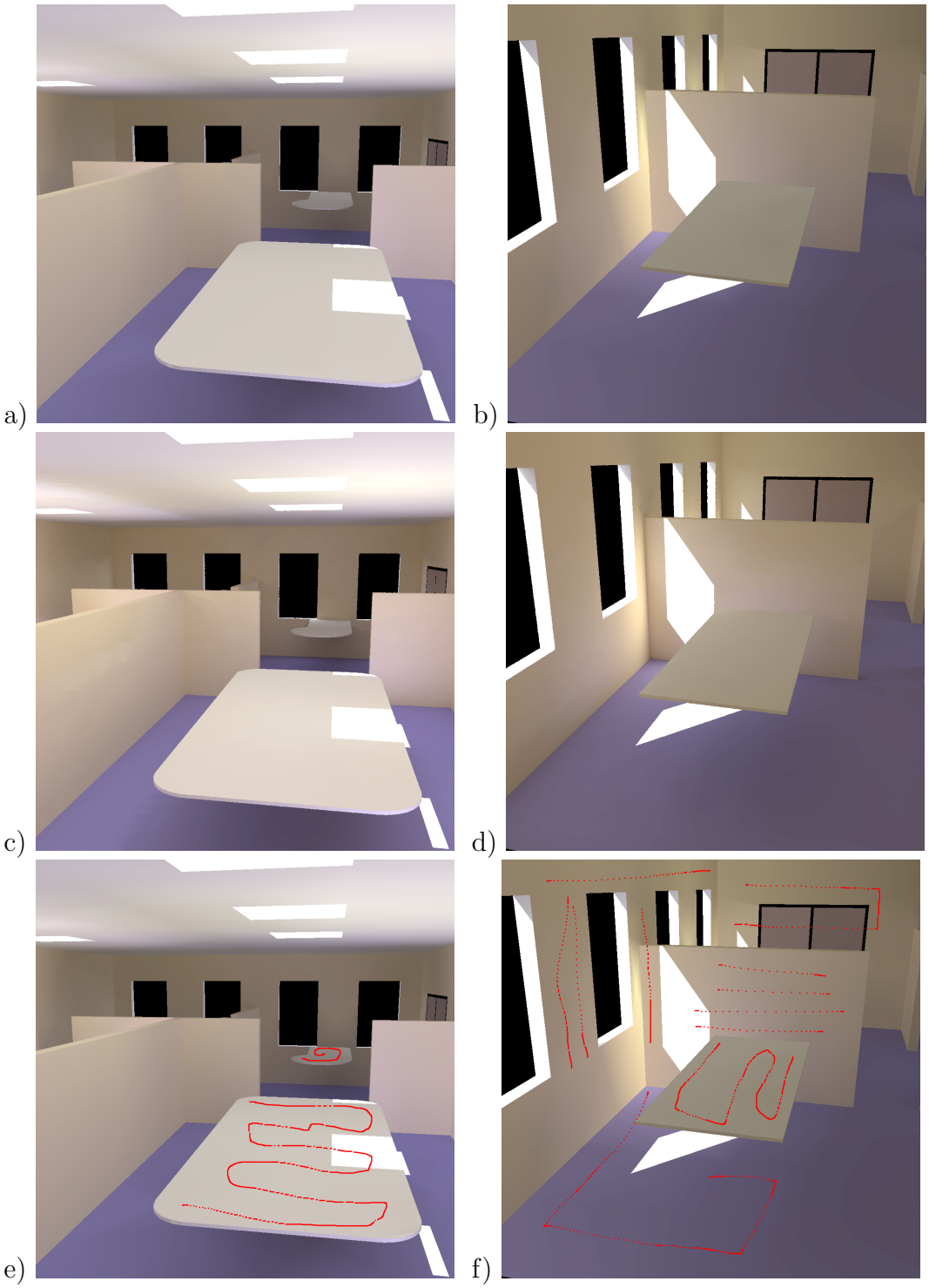

Figure 5.2: We performed a quantitative comparison between our system (a & b) and Radiance (c & d). The comparisons were done both over all pixels in the image and on a region of interest marked by the user, (shown in red in images e & f). The numerical results are presented in Table 5.1.

5.4 Quantitative Comparison with Sensors

We have also implemented an area based patch "sensor" in LSV, which corresponds to a physical sensor that one would use to measure lighting levels in the real world. We validated our patch sensor using Radiance. To do this, each patch sensor was sampled with a sufficient number of points that are fed to *rtrace*, the function in Radiance that computes irradiance. The irradiance values from the sampled points were averaged for each patch and compared with the results from LSV. Table 1 summarizes these comparisons for three different times of day on March 21 for an example model in Boston, MA. The values of the sensor patches with the lowest and highest relative difference from Radiance are indicated. Similar values were found for June 21 and December 21 (with an overall highest difference of 28%). Figure 1 shows renderings from both LSV and Radiance at the same time and day for visual comparison. This set of analyses brought confidence that our system provides reasonably accurate renderings, appropriate for use in daylighting design, although a further improvement of the resultsaccuracy is underway.

Time	Sensor	LSV	Radiance	Relative Difference
10 am	Best	46795.047	46885.998	0.19%
	Worst	2672.785	2478.62	7.83%
12 pm	Best	61244.923	61127.122	0.19%
	Worst	3952.056	3711.887	6.47%
2 pm	Best	4859.246	4851.126	0.17%
	Worst	4288.043	4155.265	3.20%

Table 5.2: Example comparison of irradiance sensor data collected from LSV and Radiance: Values of sensors with lowest and highest relative difference at three times of day for March 21.

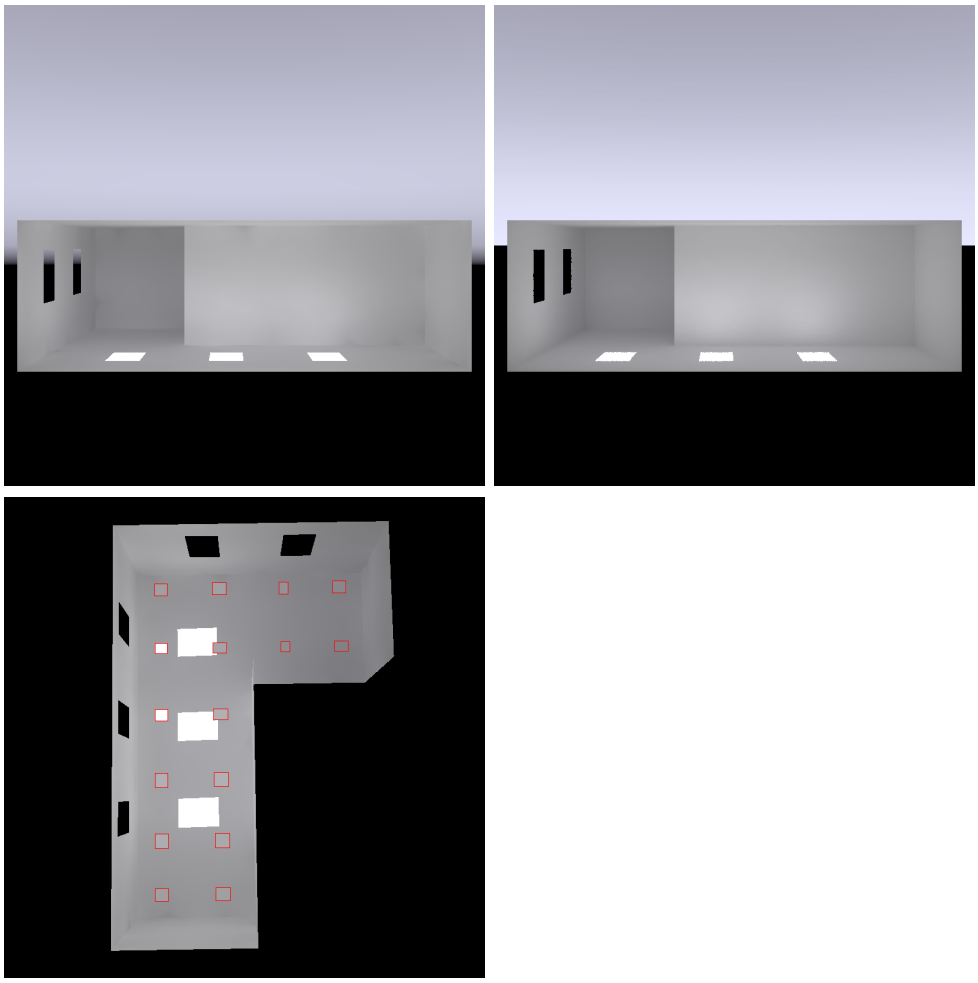

Figure 5.3: Sample renderings of a test scene using our rendering algorithm (left image) and Radiance (middle image) at noon on March 21. The right image shows the same scene with the 16 area sensors distributed across the working plane.

6. CONCLUSIONS

In this thesis, I have presented a hybrid rendering algorithm that combines stencil shadow volumes for per-pixel shadow rendering and per-patch radiosity for indirect illumination computation. The algorithm achieves a rendering speed at about 1 frame per second for a mesh with more than 1000 surfaces. I also propose a method to simulate 4D BTDF data of complex fenestration systems. This method concisely models the distribution of refraction light into several lobes, which can be used to substitute the direct illumination refracted from fenestration materials. A rendering software has been built based on the interactive rendering algorithm and simulation of fenestration systems.

Our rendering speed results compare favorably with the performance data from the study by Glaser et al. [25] of typical use times for the current practice physical heliodon daylighting analysis tool. The architectural models used in that study are of similar complexity to our test cases and our target audience is the same group of users. In that study it took an average of 86 seconds for a user to position the model on the heliodon table. This corresponds to our load and initialization time. It took an average of 29 seconds to adjust the heliodon table to capture a new sun position. Likewise, this corresponds to our relighting time. Thus we believe that our tool would integrate well into the typical schematic design phase of architectural design. Furthermore, heliodons model the sun only and can only provide shadow analyses and qualitative studies. Our tool is much more powerful than a heliodon because in addition to qualitative renderings, it includes the sky component, can be used to quantitatively measure light and can model CFS for which physical scale models are not available for use with a heliodon.

7. FUTURE WORK

Though our system can satify an architect's requirements for schematic architectural daylighting design by supporting rich set of window types and materials, there are a number of interesting opportunities for further enhancement.

The latest version of Radiance supports CFS whose transmissive properties are described by BTDF window data. Validation of our current simulation and rendering of CFS is important to quantify the accuracy of our techniques.

More window types will be supported, such as transluent panels. Currently, our simulation of window materials only supports its specular properties. However, a wide range of windows also reflect and refract part of incoming light diffusely. Incorporating the rendering and validation of these windows will be an interesting and useful improvement for the system [60].

We are currently working to extend our rendering system to support edits and computer-assisted optimization of the geometry. We have an adaptive form factor sampling framework that facilitates geometry addition, deletion, and modification. Since we are targeting the implementation to architectural students and practitioners, our current implementation makes limited use of advanced graphics hardware. However, we plan to incorporate recent GPU techniques to improve performance as our prototype is expanded. We will also investigate perceptually-appropriate tone mapping of the high dynamic range illumination values throughout the scene. Furthermore, rather than re-invent a CAD modeling user interface, we envision our algorithms as a plug-in to Sketchup [26], a tool that is favored by architects for schematic design. Providing alternate and augmented design tools to architects will improve the energy efficiency and occupant comfort in both new construction and renovation of existing architecture.

LITERATURE CITED

[1] M. Andersen and J. de Boer. Goniophotometry and assessment of bidirectional photometric properties of complex fenestration systems. *Energy and Buildings*, 38(7):836–848, July 2006.

[2] M. Andersen, C. Roecker, and J.-L. Scartezzini. Design of a time-efficient video-goniophotometer combining bidirectional functions assessment in transmission and reflection. *Solar Energy Materials and Solar Cells*, 88(1):97–118, June 2005.

[3] Marilyne Andersen, Sian Kleindienst, Lu Yi, Jaime Lee, Magali Bodart, and Barbara Cutler. An intuitive daylighting performance analysis and optimization approach. *Building Research & Information*, 36(6), November 2008.

[4] Andrew W. Appel. An efficient program for many-body simulation. *SIAM J. Sci. and Stat. Comput.*, 6(1):85–103, January 1985.

[5] D. Arasteh, J. Apte, and Y.J. Huang. Future Advanced Windows for Zero-Energy Homes. *ASHRAE Transactions*, 109(2), 2003.

[6] Josh Barnes and Piet Hut. A hierarchical o(nlogn) force-calculation algorithm. *Nature*, 324:446–449, 1986.

[7] M. Bodart, A. Deneyer, A. De Herde, and P. Wouters. Design of a single-patch sky and sun simulator. *Lighting Research and Technology*, 38(1):73–89, 2006.

[8] Nathan A. Carr, Jesse D. Hall, and John C. Hart. The ray engine. In *Graphics Hardware 2002*, pages 37–46, September 2002.

[9] Nathan A. Carr, Jesse D. Hall, and John C. Hart. GPU algorithms for radiosity and subsurface scattering. In *Graphics Hardware 2003*, pages 51–59, July 2003.

[10] CIE Commission Internationale de l'Eclairage. Spatial distribution of daylight-cie standard general sky. *CIE*, S 011/E., 2001.

[11] M. F. Cohen, D. P. Greenberg, D. S. Immel, and P. J. Brock. An efficient radiosity approach for realistic image synthesis. *IEEE Computer Graphics and Applications*, 6(3):26–35, March 1986.

[12] Michael F. Cohen, Shenchang Eric Chen, John R. Wallace, and Donald P. Greenberg. A progressive refinement approach to fast radiosity image

generation. In *Computer Graphics (Proceedings of SIGGRAPH 88)*, volume 22(4), pages 75–84, August 1988.

[13] Michael F. Cohen and Donald P. Greenberg. The hemi-cube: A radiosity solution for complex environments. In *Computer Graphics (Proceedings of SIGGRAPH 85)*, volume 19(3), pages 31–40, August 1985.

[14] Robert L. Cook, Thomas Porter, and Loren Carpenter. Distributed ray tracing. In *Computer Graphics (Proceedings of SIGGRAPH 84)*, pages 137–145, July 1984.

[15] Greg Coombe, Mark J. Harris, and Anselmo Lastra. Radiosity on graphics hardware. In *GI '04: Proceedings of the 2004 conference on Graphics interface*, pages 161–168, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004. Canadian Human-Computer Communications Society.

[16] Greg Coombe, Mark J. Harris, and Anselmo Lastra. Radiosity on graphics hardware. In *Graphics Interface 2004*, pages 161–168, May 2004.

[17] Franklin C. Crow. Shadow algorithms for computer graphics. In *Computer Graphics (Proceedings of SIGGRAPH 77)*, volume 11(2), pages 242–248, July 1977.

[18] Carsten Dachsbacher, Marc Stamminger, George Drettakis, and Frédo Durand. Implicit visibility and antiradiance for interactive global illumination. *ACM Transactions on Graphics*, 26(3):61:1–61:10, July 2007.

[19] Cyrille Damez, Kirill Dmitriev, and Karol Myszkowski. State of the art in global illumination for interactive applications and high-quality animations. *Computer Graphics Forum*, 22(1):55–77, March 2003.

[20] Zhao Dong, Jan Kautz, Christian Theobalt, and Hans-Peter Seidel. Interactive global illumination using implicit visibility. In *PG '07: Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*, pages 77–86, Washington, DC, USA, 2007. IEEE Computer Society.

[21] Reinhart C F and Fitz A. Findings from a survey on the current use of daylight simulations during building design. *Energy and Buildings*, pages 824–835, 2006.

[22] Anca D. Galasiu and Morad R. Atif. Applicability of daylighting computer modeling in real case studies: Comparison between measured and simulated daylight availability and lighting consumption. *Building and Environment*, 37:363–377, 2002.

[23] Pascal Gautron, Jaroslav Krivánek, Kadi Bouatouch, and Sumanta Pattanaik. Radiance Cache Splatting: A GPU-Friendly Global Illumination Algorithm.

In *Rendering Techniques 2005: 16th Eurographics Workshop on Rendering*, pages 55–64, June 2005.

[24] S. Gibson and R. J. Hubbold. Efficient hierarchical refinement and clustering for radiosity in complex environments. *Computer Graphics Forum*, 15(5):297–310, 1996.

[25] Dan C. Glaser, F. Whitney Smith, and Barbara Cutler. Using video for analyzing daylighting simulation tools. In *SimBuild 2006: Building Sustainability and Performance through Simulation*, August 2006.

[26] Google. SketchUp: 3D modeling software, 2008. `http://www.sketchup.com`.

[27] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modelling the interaction of light between diffuse surfaces. In *Computer Graphics (Proceedings of SIGGRAPH 84)*, volume 18(3), pages 213–222, July 1984.

[28] Gene Greger, Peter Shirley, Philip M. Hubbard, and Donald P. Greenberg. The irradiance volume. *IEEE Computer Graphics & Applications*, 18(2):32–43, March-April 1998.

[29] M. Guzowski. *Daylighting for Sustainable Design*. McGraw-Hill, 2000.

[30] Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. In *Computer Graphics (Proceedings of SIGGRAPH 91)*, pages 197–206, July 1991.

[31] Jean-Marc Hasenfratz, Cyrille Damez, François X. Sillion, and George Drettakis. A practical analysis of clustering strategies for hierarchical radiosity. *Computer Graphics Forum*, 18(3):221–232, September 1999.

[32] Tim Heidmann. Real shadows, real time. *Iris Universe*, 18:28–31, 1991. Silicon Graphics, Inc.

[33] Lisa Heschong. Daylighting and human performance. *ASHRAE*, June 2002.

[34] D. Jenkins and M. Newborough. An approach for estimating the carbon emissions associated with office lighting with a daylight contribution. *Applied Energy*, 84:608–622, 2007.

[35] Henrik Wann Jensen. Global illumination using photon maps. In *Eurographics Rendering Workshop 1996*, pages 21–30, June 1996.

[36] James T. Kajiya. The rendering equation. In *Computer Graphics (Proceedings of SIGGRAPH 86)*, pages 143–150, August 1986.

[37] Timothy L. Kay and James T. Kajiya. Ray tracing complex scenes. In *Computer Graphics (Proceedings of SIGGRAPH 86)*, pages 269–278, August 1986.

[38] Alexander Keller. Instant radiosity. In *Proceedings of SIGGRAPH 97*, Computer Graphics Proceedings, Annual Conference Series, pages 49–56, August 1997.

[39] M. Kischkoweit-Lopin. An overview of daylighting systems. *Solar Energy*, 73(2):7782, 2002.

[40] H. Köster. *Dynamic Daylighting Architecture - Basics, Systems, Projects.* Birkhäuser, 2004.

[41] G. Ward Larson and R. Shakespeare. *Rendering with Radiance - The Art and Science of Lighting Visualization.* Morgan Kaufmann Publishers, Inc., 1998.

[42] Gregory Ward Larson and Maryann Simmons. The holodeck ray cache: An interactive rendering system for global illumination in non-diffuse environments. *ACM Transactions on Graphics*, 18(4):361–368, October 1999.

[43] Szecsi Laszlo, Szirmay-Kalos Laszlo, and Mateu Sbert. Light animation with precomputed light paths on the GPU. In *GI '06: Proceedings of the 2006 conference on Graphics interface*, pages 187–194, Toronto, Ont., Canada, Canada, 2006. Canadian Information Processing Society.

[44] Christian Lauterbach, Michael Garland, Shubhabrata Sengupta, David Luebke, and Dinesh Manocha. Fast BVH Construction on GPUs. *Comput. Graph. Forum*, 28(2):375–384, 2009.

[45] N. Lechner. *Heating, Cooling, Lighting.* John Wiley & Sons Inc., 2001.

[46] M.A. Lehar and L.R. Glicksman. A simulation tool for predicting the energy implications of advanced facades. pages 513–518, 2003.

[47] Dani Lischinski, Filippo Tampieri, and Donald P. Greenberg. Combining hierarchical radiosity and discontinuity meshing. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 199–208, New York, NY, USA, 1993. ACM.

[48] Laar M. and Grimme F.W. German developments in guidance systems: an overview. *Building Research and Information*, 30(4):282–301, July 2002.

[49] J. Mardaljevic. Validation of a lighting simulation program under real sky conditions. *Lighting Research and Technology*, 27(4):181–188, 1995.

[50] J. Mardaljevic. Verification of program accuracy for illuminance modelling: assumptions, methodology and an examination of conflicting findings. *Lighting Research and Technology*, 36(3):217–242, 2004.

[51] John Mardaljevic. The bre-idmp dataset: A new benchmark for the validation of illuminance prediction techniques. *Lighting Research and Technology*, 33(2):117–136, 2001.

[52] I. Martin, X. Pueyo, and D. Tost. A two-pass hardware-based method for hierarchical radiosity. *Computer Graphics Forum*, 17(3):159–164, 1998.

[53] Morgan McGuire, John F. Hughes, Kevin Egan, Mark Kilgard, and Cass Everitt. Fast, practical and robust shadows. Technical report, NVIDIA Corporation, Austin, TX, Nov 2003. `http://developer.nvidia.com/object/fast\_shadow\_volumes.html`.

[54] L. Michel, C. Roecker, and J.-L. Scartezzini. Performance of a new sky scanning simulator. *Lighting Research and Technology*, 27(4):197–207, 1995.

[55] Mangesh Nijasure, Sumanta Pattanaik, and Vineet Goel. Interactive global illumination in dynamic environments using commodity graphics hardware. In *PG '03: Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, page 450, Washington, DC, USA, 2003. IEEE Computer Society.

[56] R. Osser, M. Andersen, and L. Norford. Development of Two Heliodon Systems and Recommendations for their Use. In *Proceedings of SOLAR 2007: Sustainable Energy Puts America to Work*, Cleveland, July 7-12, 2007.

[57] Paul Rademacher. GLUI User Interface Library, 2006. `http://glui.sourceforge.net`.

[58] B. Paule, N. Baker, S. Lawton, M. McEvoy, R. Yao, J. de Boer, H. Erhorn, S. Woesner, E. de Groot, L. Zooneveldt, and J.-L. Scartezzini. Dial-europe european daylighting design tool. In *Proceedings International Conference on Sustainable Development in Building and Environment*, 2003.

[59] R. Perez, R. Seals, and J Michalsky. All-weather model for sky luminance distribution — preliminary configuration and validation. *Solar Energy*, 50(3):235–245, 1993.

[60] C. Reinhart and M. Andersen. Development and validation of a Radiance model for a translucent panel. *Energy and Buildings*, 38(7):890–904, 2006.

[61] C.F. Reinhart, D. Bourgeois, F. Dubrous, A. Laouadi, P. Lopez, and O. Stelescu. Daylight 1-2-3 - a state-of-the-art daylighting/energy analysis software for initial design investigations. In *Submitted to Building Simulation 2007*, 2007.

[62] Tobias Ritschel, Thorsten Grosch, Min H. Kim, Hans-Peter Seidel, Carsten Dachsbacher, and Jan Kautz. Imperfect shadow maps for efficient computation of indirect illumination. *ACM Transactions on Graphics (Proc. SIGGRAPH ASIA 2008)*, 27(5):1–8, 2008.

[63] Steven M. Rubin and J. Turner Whitted. A 3-dimensional representation for fast rendering of complex scenes. In *Computer Graphics (Proceedings of SIGGRAPH 80)*, pages 110–116, July 1980.

[64] François X. Sillion. Clustering and volume scattering for hierarchical radiosity calculations. In *Fifth Eurographics Workshop on Rendering*, pages 105–117, June 1994.

[65] François X. Sillion. A unified hierarchical algorithm for global illumination with scattering volumes and object clusters. *IEEE Transactions on Visualization and Computer Graphics*, 1(3):240–254, September 1995.

[66] G.B. Singh, S.G. Abraham, and F.H. Westervelt. Parallel radiosity computation on a shared memory multiprocessor. In *The 36th Midwest Symposium on Circuits and Systems*, pages 165–168, 1993.

[67] Joseph M. Siry. *Carson Pirie Scott: Louis Sullivan and the Chicago Department Store*. The University of Chicago Press, 1956.

[68] Brian Smits, James Arvo, and Donald Greenberg. A clustering algorithm for radiosity in complex environments. In *Proceedings of SIGGRAPH 94*, Computer Graphics Proceedings, Annual Conference Series, pages 435–442, July 1994.

[69] Ephraim M. Sparrow. On the calculation of radiant interchange between surfaces. *Modern Developements in Heat Transfer*, 1963.

[70] Square One Research. ECOTECT, 2000-2008. `http://ecotect.com`.

[71] M. Stamminger, H. Schirmacher, P. Slusallek, and Hans-Peter Seidel. Getting rid of links in hierarchical radiosity. *Computer Graphics Forum*, 17(3):165–174, 1998.

[72] R. Sullivan, L. Beltran, E.S. Lee, M. Rubin, and S.E. Selkowitz. Energy and Daylight Performance of Angular Selective Glazings. Lawrence Berkeley National Laboratory, Supported by the U.S. Department of Energy, 1998.

[73] G. Sweitzer. Prismatic panel sidelighting systems: Daylighting distribution and electric lighting user patterns in perimeter office workplaces. In *Right Light Bright Light*, 1991.

[74] John R. Wallace, Michael F. Cohen, and Donald P. Greenberg. A two-pass solution to the rendering equation: A synthesis of ray tracing and radiosity methods. In *Computer Graphics (Proceedings of SIGGRAPH 87)*, volume 21(4), pages 311–320, July 1987.

[75] Greg Ward. Radiance: Synthetic imaging system, 1985-1997. `http://radsite.lbl.gov/radiance/HOME.html`.

[76] Greg Ward and Paul Heckbert. Irradiance gradients. In *Eurographics Rendering Workshop*, pages 85–98, May 1992.

[77] Gregory J. Ward. The radiance lighting simulation and rendering system. In *Proceedings of SIGGRAPH 94*, Computer Graphics Proceedings, Annual Conference Series, pages 459–472, July 1994.

[78] Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear. A ray tracing solution for diffuse interreflection. In *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 85–92, New York, NY, USA, 1988. ACM Press.

[79] A.R. Webb. Considerations for lighting in the built environment: Non-visual effects of light. *Energy and Buildings*, 38(7):721–727, 2006.

[80] Turner Whitted. An improved illumination model for shaded display. *Communications of the ACM*, 6(23):343–349, 1980.

[81] Andrew Willmott and Paul Heckbert. An empirical comparison of radiosity algorithms. Technical Report CMU-CS-97-115, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, May 1997.

[82] Andrew Willmott, Paul S. Heckbert, and Michael Garland. Face cluster radiosity. In *Eurographics Rendering Workshop 1999*, June 1999.

[83] Andrew J. Willmott and Paul S. Heckbert. An empirical comparison of progressive and wavelet radiosity. In *Eurographics Rendering Workshop 1997*, pages 175–186, June 1997.

[84] Andrew Woo, Pierre Poulin, and Alain Fournier. A survey of shadow algorithms. *IEEE Comput. Graph. Appl.*, 10(6):13–32, 1990.

[85] Kun Zhou, Qiming Hou, Rui Wang, and Baining Guo. Real-time kd-tree construction on graphics hardware. *ACM Transactions on Graphics*, 27(5):126:1–126:11, December 2008.