5th National Conference of the International Building Performance Simulation Association-USA
Madison, Wisconsin, August 1-3, 2012

HARDWARE ACCELERATED COMPUTATION OF DIRECT SOLAR RADIATION THROUGH TRANSPARENT SHADES AND SCREENS

Nathaniel L. Jones and Donald P. Greenberg
Program of Computer Graphics, Cornell University, Ithaca, New York, USA

ABSTRACT

Perforated and transparent shading devices can greatly reduce the amount of incident solar radiation on a building, but they present many challenges for building energy modeling. This paper describes two new methods for calculating direct solar gains on surfaces in geometrically complex environments. One handles shading surfaces containing irregular openings and intricate perforated patterns, while the other handles transparent shading surfaces. Both use fast shader programs written for graphics hardware to account for the effect of the shading surface's solar incidence angle on light transmission. The two methods can be used alone, together, or in conjunction with other shading algorithms. The effectiveness of these methods is graphically demonstrated on schematic design level architectural CAD models of real buildings with intricate shading devices. These fast algorithms allow the user to rapidly simulate thermal loads in EnergyPlus and compare the effects of many design alterations.

INTRODUCTION

Architects use perforated and transparent surfaces to shade buildings because they limit solar transmittance and preserve views (Figure 1). These shading devices can significantly alter the effect of solar radiation on a building, but they are challenging to model for building energy simulation (BES), particularly when they take on complex geometric arrangements. It is often necessary to create simplified models of shading devices for BES analysis, but simplification is itself time consuming and difficult for two reasons. First, overlapping shadows have additive effects, and the amount of overlap varies with the time of day. Second, the amount of solar radiation transmitted through the surface depends on the incidence angle at which light hits the surface. Many BES tools, including EnergyPlus (LBNL 2011), do not consider the latter fact in shading calculations, even when geometry is modeled without simplification.

This paper describes a novel approach for calculating the effects of geometrically complex transparent or perforated shading devices at high speeds without the need for simplification. The key innovation is the use of programmable shaders to precisely compute the effect of each shading surface as it is rendered to an off-screen image buffer. Shaders (not to be confused with shading devices) are small programs that run in a highly parallel environment on the computer's graphics processing unit (GPU). The process begins with a three-dimensional building model created in architectural computer-aided design (CAD) software. The complete geometry of the model is rendered as viewed from a variety of solar positions, each time using shader programs to reproduce the effect of transmissive shading devices. The sunlit portion of each building surface is calculated and fed as input to a modified version of EnergyPlus. For complex shading geometries, this method is faster than EnergyPlus's

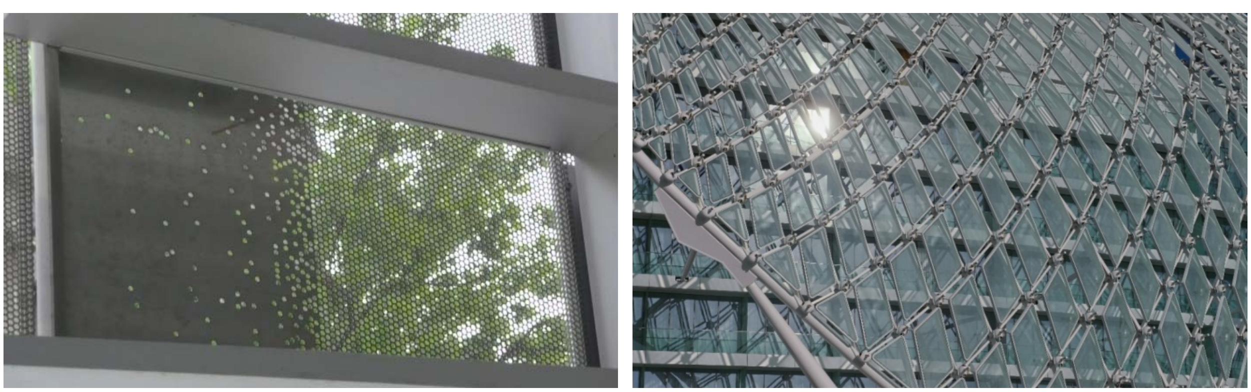


Figure 1 Shading devices with perforated screens (left) and transparent panels (right).

sunlit area calculations by several orders of magnitude, and it considers angle dependent transmittance properties of materials that are ignored by EnergyPlus.

Two shader programs are demonstrated through analysis of proposed and real buildings. The first simulates the perforated metal screen designed to surround Bill & Melinda Gates Hall in Ithaca, New York, by Morphosis Architects. The second reproduces the effect of the glass and steel grid shell over the Yas Viceroy Abu Dhabi hotel, designed by Asymptote Architecture and completed in 2009. In each study, several variations of the shading devices are produced using CAD software. Although the models feature highly complex geometry, EnergyPlus analysis can be performed quickly using the results of the shader programs. This allows fast comparison of the simulation results.

PREVIOUS WORK

The quantity of direct solar radiation incident on a surface is described by the expression

$$I_b \frac{A_s}{A_t} \cos\theta \qquad (1)$$

where I_b is the intensity of direct radiation from weather data, A_s and A_t are the surface's sunlit and total areas, respectively, and θ is the incidence angle of the sun's rays (EnergyPlus Development Team 2011). To compute the area of the sunlit portion of the surface, an algorithm must account for an arbitrarily large number of surfaces that may shade it, some of which may not be completely opaque. This calculation must be repeated for many different solar positions. EnergyPlus performs this calculation using projection and polygon Boolean operations (Groth and Lokmanhekim 1969, Walton 1979). With version 7.0, EnergyPlus introduces a more robust polygon Boolean algorithm by Sutherland and Hodgeman (1974). Both approaches suffer the same drawbacks; they are unreasonably slow for large collections of shading surfaces, and they require that all shading surfaces be convex.

Jones et al. (2011) resolve these issues using graphics hardware to quickly compute the number of pixels visible to the sun in an image of a surface. They define the projected sunlit surface fraction (PSSF) found in Equation 1 and observe that

$$PSSF = \frac{A_s}{A_t} \cos\theta \approx N \times A_p \times \frac{1}{A_t} \qquad (2)$$

where N is the number of visible pixels obtained from a hardware occlusion query, and A_p is the area represented by a single pixel in model space. The rendering steps that produce the pixel count are performed in the GPU's fixed function pipeline (FFP), a set of hardware-accelerated operations that enable basic rendering (Figure 2). This method outperforms other algorithms in computation speed by several orders of magnitude with negligible loss of accuracy. However, its applicability to non-opaque shading surface is limited by the hardcoded steps of the FFP.

Greater control over the graphics pipeline can be achieved using programmable shaders. Shaders are simple programs that bypass portions of the FFP to allow customized processing of image data (Akenine-Möller et al. 2008). Originally, shaders were developed to reproduce visual characteristics of materials in animations (Hanrahan and Lawson 1990). Today, three types of shaders can be loaded onto the GPU at application runtime. Vertex shaders process the position, color, and other properties of geometric vertices. Geometry shaders allow customized creation of geometric primitives such as points, lines, and triangles based on those vertices. Fragment shaders control the appearance of individual pixels belonging to geometric primitives.

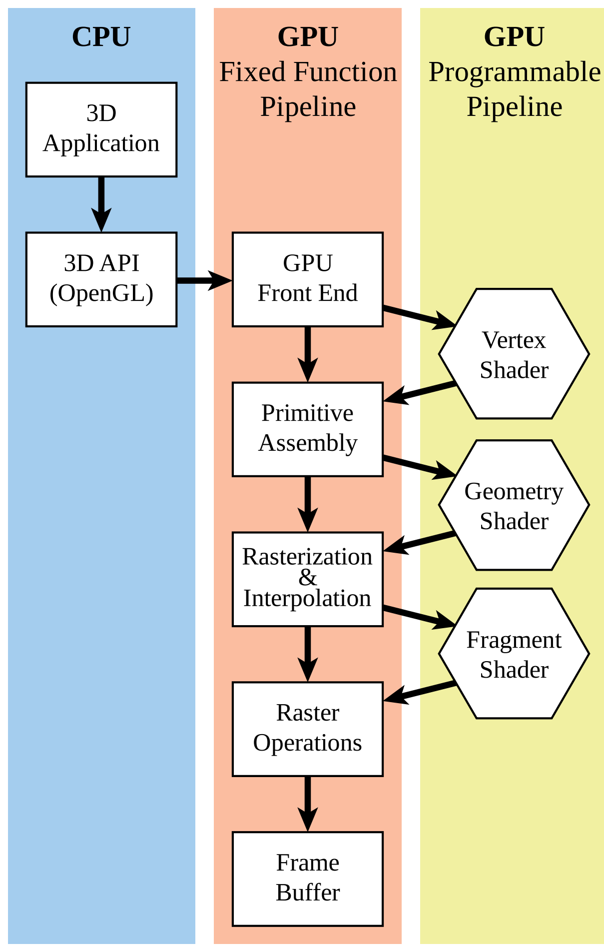


Figure 2 The GPU's fixed function pipeline offers a default set of algorithms for transforming geometric input from the CPU into an image. These algorithms may be replaced with custom shader programs.

METHODS

The two algorithms described in this paper are modifications of pixel counting. Both use shader programs to calculate sunlit surface areas in situations where current methods fail or are too slow to be effective. The first method uses a geometry shader to simulate the effect of perforated screens. The second uses a fragment shader to simulate shading by transparent surfaces, including the effect of incidence angle on transmittance.

Perforated Shading Surfaces

Intricate geometry such as that of a metal mesh or screen is too complicated to be modeled geometrically, especially if the material is used over a large surface area. Instead, the appearance of a complex surface can be created by applying a texture image to the surface (Catmull 1974). The texture image source may be any digital image and may be created with a variety of image generation software packages. Architectural modeling programs such as SketchUp (Google 2012) allow a texture image to be associated with a material and applied to surfaces within the model. In order to be understood as a perforated shading device, the texture image is required to include opacity information. The FFP renders pixels of a textured shading surface only where its opacity is above a threshold. This allows pixels of the receiving surface to be counted by an occlusion query even if they are behind transparent portions of the textured shading surface.

The texture image generally shows the material as seen orthogonally and thus provides no depth information. When viewed obliquely, the texture image does not reproduce changes in transmittance due to material thickness (Figure 3a). To account for this, a geometry shader is used to create both a front and back surface given the single surface submitted to the graphics hardware (Figure 3b). Additional surfaces placed between the front and back surfaces by the geometry shader prevent moiré patterns in which holes unintentionally line up from certain viewpoints (Figure 3c). This method is preferable to modeling the geometry of perforations because it reduces the simulation's memory requirements, involves no additional preprocessing, and fits naturally into the architect's modeling process.

Transparent Shading Surfaces

When transparent shading surfaces such as glass are present, color information must be supplied to the graphics hardware as well. Color can be recorded as a single luminance value rather than the red, green, and blue values typical in computer graphics. The receiving surface is rendered with a luminance value of one and all other surfaces with a luminance value of zero. Transparent surfaces that occlude the receiving surface create pixels with fractional luminance values between zero and one. The pixel count is now replaced with the sum of pixel luminances L_i as follows

$$PSSF \approx \frac{A_p}{A_t} \sum_{i=1}^{w \times h} L_i \qquad (3)$$

where w and h are the width and height of the image buffer in pixels.

Memory transfer from graphics hardware to the CPU is expensive, even if performed asynchronously. Optimal performance is achieved by summing local neighborhoods of the image buffer on the GPU and returning a compressed image of 32×32 pixels to the CPU for a final summation. Sums can be computed in hardware using a vertex texture (Scheuermann and Hensley 2007) or using general purpose GPU programming techniques (Harris et al. 2007).

A faster approach is to average neighboring luminance values using mip mapping (Williams 1983). Mip maps are trees of compressed images made by progressively reducing the resolution of an original image. At each mip map level, regions of four pixels from the previous mip map level are averaged to a single pixel value, so that the resulting image is half the size of its predecessor in each dimension. The original image is considered mip map level zero. Because mip maps are generated in graphics hardware, they can be made very quickly without specialized programming. The sum of pixel luminances from the full image is replaced to give

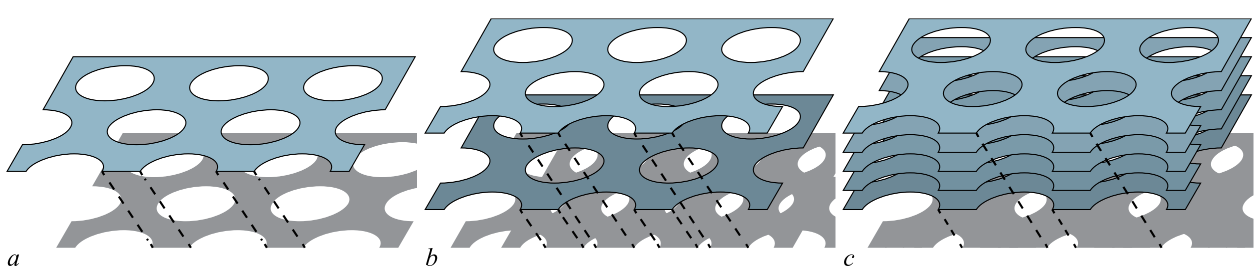


Figure 3 A two-dimensional texture allows a uniform amount of radiation to pass through from any angle (a). Duplicating the texture for the front and back surface planes still allows radiation to pass through incorrectly at certain angles (b), which is corrected by adding interstitial planes (c).

$$PSSF \approx 4^m \frac{A_p}{A_t} \sum_{i=1}^{w_m \times h_m} L_i \qquad (4)$$

where w_m and h_m are the width and height of the mth mip map level.

Typical rendering of transparent surfaces by the FFP requires that the opacity be set in advance. However, transmittance depends on the incidence angle of radiation on the surface (Rubin et al. 1998). This variation becomes even more pronounced if the transparent surface is curved. Instead, the transmittance of each material can be modeled as a polynomial function and calculated by a fragment shader. During rendering, the shader program calculates the opacity of each pixel belonging to the surface in parallel.

VALIDATION

The accuracy of these methods was tested using simple models with predictable behavior. Each model consisted of a horizontal square receiving surface measuring 1 m on a side with a similar shading surface suspended 0.1 m above it. PSSF was measured for 43,805 solar positions corresponding to 6-minute time intervals at 42° north latitude. This process was carried out for both the new and accepted methods for both transparent and perforated shading materials.

The transparent shading surface was modeled as a double-pane window consisting of two 5.7 mm glass panes with an overall solar transmittance of 48% at normal incidence (Figure 4a). Results of pixel luminance counting were compared to analytical calculations based on the method from EnergyPlus (Groth and Lokmanhekim 1969) with a robust polygon Boolean algorithm (Vatti 1992) and angle dependency. The difference in PSSF was negligible (Table 1).

The accuracy of pixel counting with textures is more difficult to assess because it is highly dependent on the size and arrangement of perforations. The shading material modeled for testing was a 6.35 mm (0.25 in.) thick panel perforated with 6.35 mm diameter holes arranged in a hexagonal pattern so as to give it a solar transmittance of 50% at normal incidence. The shading device was modeled both as a textured surface using a geometry shader to mimic its thickness (Figure 4b) and as a geometric object composed of 391,950 polygons (Figure 4c). As this amount was too large to perform analytical calculations on in a reasonable amount of time, both versions were analyzed by pixel counting. The accuracy achieved with the texture increased with the number of layers created by the geometry shader, as did the time required for simulation. A reasonably small error was achieved using ten layers (Table 1). The appropriate number of layers for other perforated materials will be a function of hole geometry, thickness, and texture resolution.

Table 1 PSSF deviation compared to accepted methods

SHADING DEVICE	DEVIATION IN PSSF	
	MEAN	MAXIMUM
Transparent	1.24×10^{-5}	1.22×10^{-3}
Perforated	1.15×10^{-3}	1.07×10^{-2}

Both new methods perform somewhat slower than the original pixel counting algorithm described by Jones et al. This is to be expected, as they necessarily expose portions of the FFP that can be conveniently turned off for pixel counting of opaque surfaces. However, they are sufficiently fast to perform analysis on complex models that would have been impractical to study using analytical methods, as will be demonstrated through case studies.

CASE STUDIES

The following two case studies demonstrate the utility of the fast algorithms described above for calculating direct incident radiation through perforated screens and transparent shades. Bill & Melinda Gates Hall serves

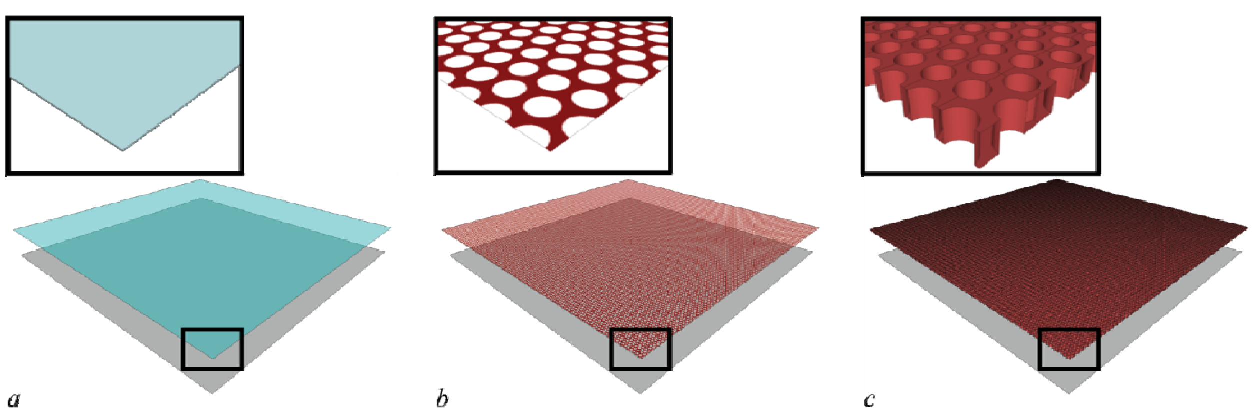


Figure 4 Test models of a gray receiving surface below a transparent shading surface (a), a perforated texture (b), and a shading device with geometrically-modeled perforations (c).

as an example for analysis of perforated screens, and the Yas Viceroy Abu Dhabi demonstrates analysis of transparent shading devices. Previously, the amount of time required to generate simplified models for buildings of their complexity, and then to perform BES analysis on those models, prohibited the analysis of more than a handful of design options. However, these fast algorithms allow BES analysis to be performed on each design options in roughly the same amount of time required to create that option.

The goal in modeling the two buildings was to achieve a level of geometric detail expected in schematic design phase architectural models. Spaces within the building envelopes were partitioned into thermal zones with geometry that remained static through all of the design options; only the exterior shading devices were altered. Information was not readily available on the buildings' materials and mechanical systems, nor are such decisions typically made during schematic design. Instead, both buildings were modeled with a small palette of materials and a simple purchased air system.

For each design alternative considered, sunlit surface fractions were calculated in a preprocessing module prior to full BES analysis. The preprocessing module, implemented in Java with shader programs written in the OpenGL Shading Language, calculated the sunlit fraction of each building envelope surface at regular intervals over the course of the day and year. B-spline interpolation was then used to fill in additional values in order to provide EnergyPlus with data at fifteen-minute time steps for every day of the year. Jones et al. showed that satisfactory accuracy can be achieved using B-spline interpolation when the original data is generated only every fourteenth day of the simulated year at 15° intervals of change in solar azimuth, which translates to roughly 400 data samples per building envelope surface. Simulation times are reported both for the preprocessor and for the modified version of EnergyPlus, which used the preprocessor's output in place of running its own sunlit surface fraction algorithm. All simulations ran on a 3.4 GHz Intel® Core™ i7-2600 workstation with an Nvidia® GeForce® GTX 570 graphics card.

Bill & Melinda Gates Hall

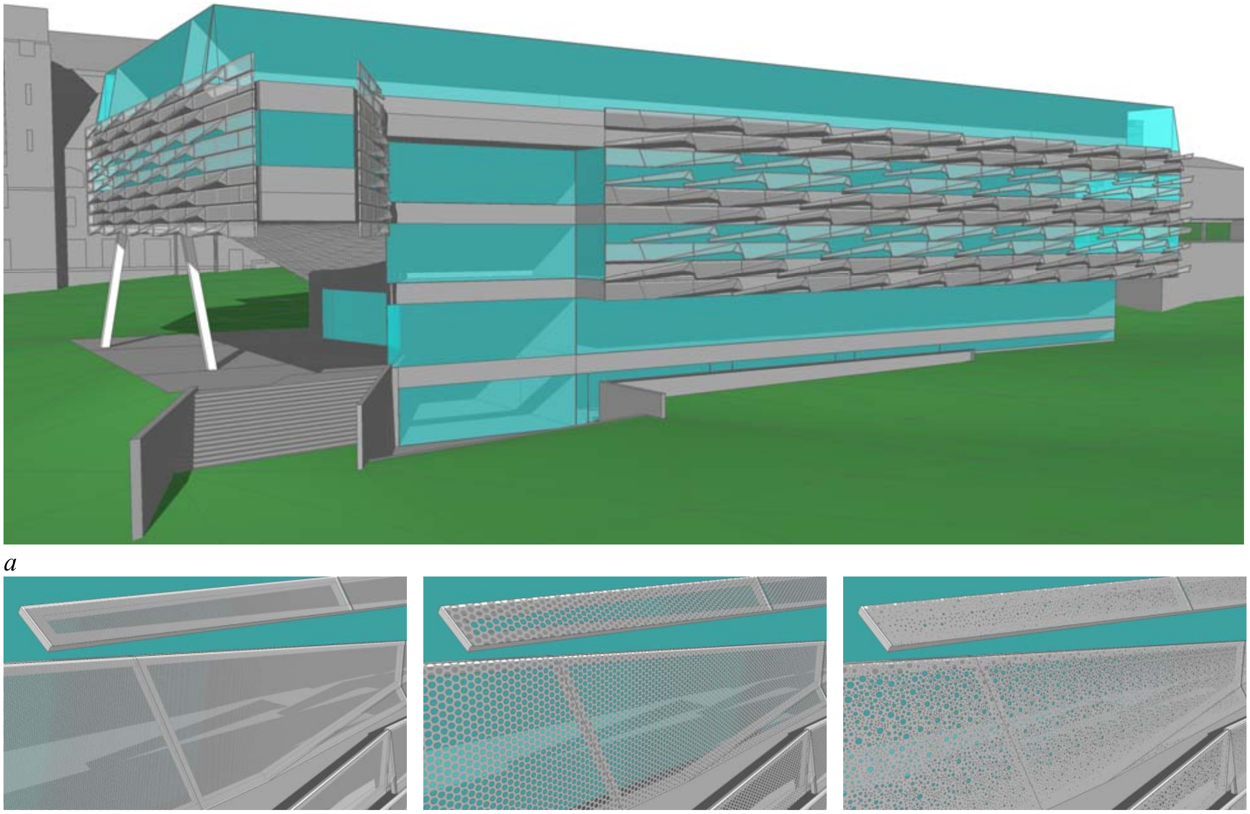
Bill & Melinda Gates Hall is a four-story computer science building under construction in Ithaca, New York. It features a perforated metal screen suspended around its second and third floors that both regulates the entry of light and air and protects its windows from a neighboring baseball diamond (Figure 5). The screen's geometry includes many folds and irregular shapes, allowing complex overlaps that make approximation of its transmittance properties difficult. Shade is also provided by hilly terrain and eleven nearby buildings included in the model.

Figure 5 Bill & Melinda Gates Hall concept rendering.

To test the speed and predictive abilities of the analysis method for perforated screens, three design alternatives for the screens were considered. All options used 6.35 mm (0.25 in.) thick perforated panels, but the size, number, and arrangement of openings varied. The first alternative had 6.35 mm diameter holes arranged in a hexagonal pattern so as to give it a solar transmittance of 50% at normal incidence (Figure 6b). The second had 25.4 mm (1 in.) holes arranged in the same pattern with increased spacing between holes so as to maintain 50% overall normal transmittance (Figure 6c). The final alternative had a random arrangement of holes of varying sizes that produced a transmittance gradient from top to bottom (Figure 6d). The purpose of this last alternative was to provide an option in which the panel's transmittance could not be described with a single value.

Timing results for tests on Bill & Melinda Gates Hall are shown in Table 2. In general, preprocessing could be performed in less than six seconds, and the modified version of EnergyPlus took under three minutes for a full-year simulation. In contrast, a simulation with an unmodified version of EnergyPlus at the same level of detail took twenty-four days and generated incorrect results.

Two observations are particularly noteworthy. First, while the first two design alternatives have the same nominal transmittance value, the larger holes of the second option admit substantially more direct radiation at oblique angles. Over the course of the year, south-facing windows received 35% more direct radiation annually in the second design alternative than in the first, based on local weather data. East- and west-facing windows received twice as much direct solar radiation in the second design alternative. Second, although the nominal transmittance of the third shading alternative is unknown and different for every receiving surface, it is no more difficult or time consuming to perform analysis on this option than the other two. In fact, no expert knowledge is necessary to analyze this complex shading device.



a *b* *c* *d*

Figure 6 BES model of Bill & Melinda Gates Hall (a) with details showing screens with 6.35 mm perforations (b), 25.4 mm perforations (c), and random perforations (d).

Yas Viceroy Abu Dhabi

The Yas Viceroy Abu Dhabi hotel consists of two elliptical towers of varying heights connected by a two-story bridge that spans a Formula One race track. Shade is provided by cantilevered balconies with glass fencing and by a curving steel frame called the grid shell, which supports 5800 glass panels that can be illuminated at night (Figure 7). The double curvature and varied width of the grid shell make a determination of its shading characteristics impractical without simulation.

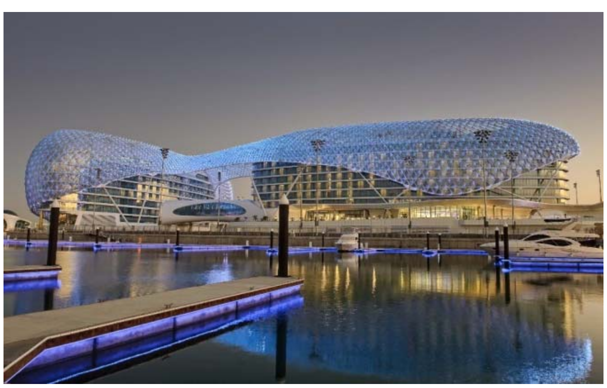


Figure 7 The Yas Viceroy Abu Dhabi.

To demonstrate the effectiveness of the analysis method for transparent shading surfaces, three design alternatives were developed for the grid shell. A parametric model of the grid shell allowed changes to be made quickly to its geometry and materials. In the first design option, each glazed facet was rotated 20° out of plane, approximately matching the design as built (Figure 8b). In the second, the glazed facets were rotated 90° out of plane (Figure 8c). The final variation maintained the original glazing angle, but replaced the single panes of glass with a triple-glazed construction (Figure 8d). The shading provided by the balconies remained constant through all three design alternatives.

Timing results of tests on the Yas Viceroy Abu Dhabi are shown in Table 2. Generally, shading preprocessing could be performed within three minutes, while the modified version of EnergyPlus took about thirty-eight minutes to run for each design option. Running the same model in a standard EnergyPlus release would have been impractical due to time constraints and would have generated inaccurate results.

The second design option is of particular interest because the effect of pivoting the glazed panels is essentially impossible to predict without simulation. Rotating the panels reduces their projected area onto the

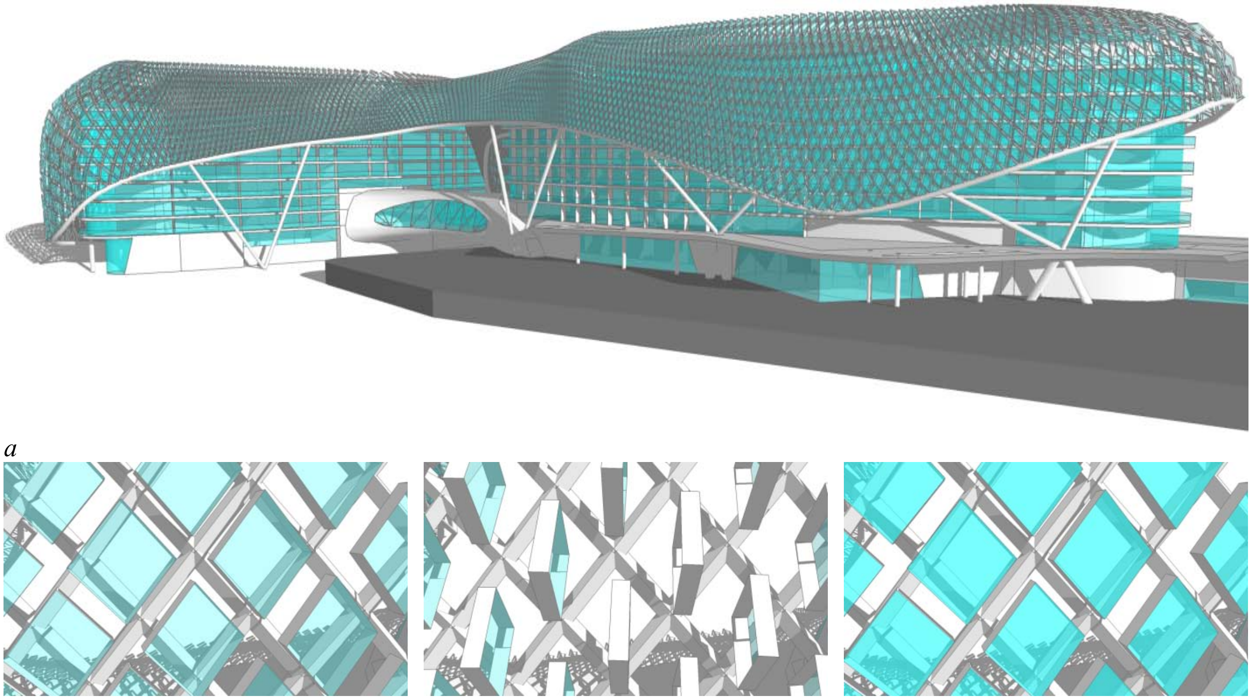


Figure 8 BES model of the Yas Viceroy Abu Dhabi (a) with details showing panels rotated at 20° (b), 90° (c), and 20° with triple glazing (d).

building's envelope, but also decreases their transmittance to radiation in the grid shell's normal direction. At the same time, the rotated glass panes allow more light to penetrate at certain oblique angles, and different shadows are cast by the pivoted frames. The unpredictable effect of the change was measured for two representative hotel rooms, one facing east and one south. Insolation on the two rooms increased by 4.8% and 5.3%, respectively.

The third design option produced a more pronounced effect. The triple-glazed panels had a nominal solar transmittance of 34%, compared to the single-glazed panels' 69% solar transmittance at direct incidence. The two representative hotel rooms received 22% and 24% less direct radiation as a result of the change.

CONCLUSION

Programmable shaders can compute sunlit surface fractions quickly and accurately for shading conditions that could not previously be analyzed with BES software. This paper presented two algorithms that can be implemented in shader programs; one handles perforated shading devices, and the other handles transparent shading devices. By incorporating these algorithms into a preprocessor and feeding the results directly to a modified version of EnergyPlus, direct

Table 2 Model complexity and simulation times at various preprocessing sampling frequencies

OPTION	THERMAL ZONES	RECEIVING SURFACES	SHADING SURFACES	PREPROCESSING (SECONDS)			ENERGYPLUS (SECONDS)
				Fine[a]	Medium[b]	Coarse[c]	
Bill & Melinda Gates Hall							
6.35 mm holes	34	149	32,709	7.78	5.27	4.12	154.67
25.4 mm holes				7.79	5.28	4.13	154.35
Random holes				8.77	5.91	4.64	154.49
Yas Viceroy Abu Dhabi							
20° single pane	130	2385	94,997	209.04	145.96	114.60	2286.93
90° single pane				205.77	143.82	113.02	2277.44
20° triple pane				209.29	146.20	114.77	2284.70

[a] Sampled at 10° intervals of solar azimuth every 14th day.
[b] Sampled at 15° intervals of solar azimuth every 14th day.
[c] Sampled at 20° intervals of solar azimuth every 14th day.

solar radiation on buildings with complex shading devices can be simulated without geometric simplifications or material approximations that may cause loss of accuracy.

This is significant because modern CAD software allows architects to experiment with increasingly complex geometric forms early in the design process. These complex forms change how the building interacts with its environment and affect its thermal performance. Fast BES analysis gives building designers the information they need to make more intelligent decisions regarding form, orientation, and materials. The ability to perform shading calculations directly on the geometric model created by the architect removes a number of steps from typical simulation procedures. This is in keeping with the goal to promote accurate thermal analysis of buildings early in the design process when there are more opportunities to improve building performance.

ACKNOWLEDGEMENTS

This material is based upon work supported by the Department of Energy under Award Number DE-EE0003921. Tests were run on prototype software created with the assistance of Lars Schumann and David Bosworth. Andrew Heumann and Colin McCrone modeled the Yas Viceroy's grid shell and bridge. The authors thank Kevin Pratt for his input and Morphosis Architects for providing plans for Bill & Melinda Gates Hall.

DISCLAIMER

REFERENCES

Akenine-Möller, T., Haines, E. and Hoffman, N. 2008. *Real-Time Rendering, Third Ed*. Natick, MA: A. K. Peters, Ltd.

Catmull, E.E. 1974. *A Subdivision Algorithm for Computer Display of Curved Surfaces*. Thesis (PhD). University of Utah, Salt Lake City, UT.

EnergyPlus Development Team 2011. EnergyPlus engineering reference: The reference to EnergyPlus calculations. *EnergyPlus Version 7.0*. US Department of Energy.

Google 2012. Google SketchUp, Version 8. Google Inc., Mountain View, CA.

Groth, C.C. and Lokmanhekim, M. 1969. Shadow — A new technique for the calculation of shadow shapes and areas by digital computer. In: *Second Hawaii International Conference on System Sciences*, 22-24 January 1969 Honolulu, HI. 471-474.

Hanrahan, P. and Lawson, J. 1990. A language for shading and lighting calculations. *Computer Graphics*, 24 (4), 289-298.

Harris, M., Shubhabrata, S. and Owens, J.D. 2007. Parallel prefix sum (scan) with CUDA. *GPU Gems 3*. H. Nguyen, ed. Upper Saddle River, NJ: Addison-Wesley.

Jones, N.L., Greenberg, D.P. and Pratt, K.B. 2011. Fast computer graphics techniques for calculating direct solar radiation on complex building surfaces. *Journal of Building Performance Simulation*. DOI: 10.1080/19401493.2011.582154.

LBNL 2011. EnergyPlus, Version 7.0. Lawrence Berkeley National Laboratory, Berkely, CA.

Rubin, M., Von Rottkay, K. and Powles, R., 1998. Window optics. *Solar Energy*, 62 (3), 149-161.

Scheuermann, T. and Hensley, J. 2007. Efficient histogram generation using scattering on GPUs. In: *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games*, 30 April-2 May 2007 Seattle, WA. 33-37.

Sutherland, I.E. and Hodgman, G.W. 1974. Reentrant polygon clipping. *Communications of the ACM*, 17 (1), 32-42.

Vatti, B.R. 1992. A generic solution to polygon clipping. *Communications of the ACM*, 35 (7), 56-63.

Walton, G.N. 1979. The application of homogeneous coordinates to shadowing calculations. *ASHRAE Transactions*, 84 (1), 174-180.

Williams, L. 1983. Pyramidal parametrics. *Computer Graphics*, 17 (3), 1-11.