INTEGRATING SHADOW CASTING METHODOLOGY AND THERMAL SIMULATION

Ali Malkawi, Ph.D.
Wassim Jabi, M.Arch., A.B.D.
College of Architecture and Urban Planning
University of Michigan
2000 Bonisteel Blvd.
Ann Arbor, MI 48109-2069

ABSTRACT

This paper describes an experiment that integrates shadow casting methodology and thermal simulation algorithms developed by the authors. The 3D shadow procedures use a polyhedral representation of solids within a Cartesian space that allows for accurate casting of shadows. The algorithm is also capable of calculating surface areas of polygonal shadows of any arbitrary shape and size. The thermal simulation algorithms – using the Transfer Function Method (TFM) – incorporate the shaded area calculations to better predict solar heat gain from glazing based on transmitted, absorbed, and conducted cooling loads. The paper describes the use of a 3D computer model to illustrate the impact of the pattern and area of shading on the visual and thermal properties of building apertures. The paper discusses the objectives of this experiment, the algorithms used, and their integration. Conclusions and findings are drawn.

1. INTRODUCTION

Many commercial CAD packages available today can display photo-realistic three-dimensional models that incorporate lighting, shadows, reflections, and refractions. However, generally, either the algorithms that calculate these effects have little correspondence to the physical artifact being modeled or the data they generate is not accessible to the user. In many cases, only the geometric data of the model itself can be exported to other simulation packages that need to re-generate the required information to carry out their calculations.

Thermal simulation tools can accurately and extensively predict the behavior of the model being analyzed. However, they usually lack the graphical interface to input complex three-dimensional data. As a result, they often have to rely on a limited and significantly abstracted representation.

The experiment described in this paper aims to overcome these limitations by: (1) Using an accurate shadow-casting algorithm that allows access to the internal data structures needed to calculate the insolated and shaded areas, and (2) to supply this data to a rigorous thermal simulation tool to better evaluate the building elements affected by shadows both quantitatively and qualitatively.

The interface allows the user to construct a 3D model of any complexity and specify the incident angle of sun rays. The visual result is then displayed on the screen while the resulting geometries are saved in a database to be read by the thermal simulation program.

The experiment strives to achieve several objectives:

- To use a fast and computationally inexpensive algorithm that yields the necessary data for visualization as well as thermal simulation.
- To enable the user to quickly build 3D models, cast shadows, view the results graphically, simulate thermal behavior and modify the design solution accordingly – all within an integrated environment.
- To better predict thermal behavior due to a more accurate output of insolated and shaded areas.
- To offer the end-user more flexibility in simulating windows and walls of any arbitrary shape, size, and orientation. In particular, to offer the ability to model shading devices that cast shadows with holes and disjoint pieces.

2. THE SHADOW CASTING ALGORITHM

The shadow casting algorithm (Jabi, 1989) was developed as part of the GEDIT solids modeling environment (Turner, 1993). In GEDIT, solids are represented using a true polyhedral boundary representation that enables them to have holes and disjoint pieces grouped as one polyhedral set (Borkin et al., 1978). The shadow casting algorithm uses this polyhedral representation as a starting point. The angle of the sun is represented as a directional vector within a pre-determined Cartesian space (Figure 1). Furthermore, North coincides with the positive Y direction, East with a positive X direction and so forth.

After the sun direction is encoded as a vector (Figure 2-A), a transformation matrix (Foley and Van Dam, 1982) is derived that, when applied to the model, projects it as seen from the point of view of the sun (Figure 2-B). Using the incident angle of the sun and the normal vector of the surfaces, the polygons of the model are classified into two categories: (1) Those facing away from the sun and (2) those that face the sun. Surfaces that face away from the sun are discarded as they do not receive any shadows. Surfaces that face the sun have two data structures associated with them: the insolated polygons and the shaded polygons. At the start of the algorithm, the data structures for the insolated polygons contain all the polygons of the parent surface while those of the shaded polygons are empty. Each surface is then tested for intersection with all other surfaces to determine if shadows need to be cast. In larger models, many surfaces fail this test quickly with a simple two-dimensional bounding rectangle intersection test. This is due to the large distances that separate surfaces indicating the absence of shadow interference. Surfaces that do intersect are examined further to determine if the receiving surface is farther from the sun than the casting surface – otherwise shadow casting is not carried out. While determining nearness to the sun, one must test only the sub-areas that form the intersection of the two surfaces and not the surfaces as a whole. It is also important to note that the location of the sun can be any arbitrary point along the sun vector that lies beyond the boundaries of the overall model.

Once the surfaces pass the intersection test, the shadows are ready to be projected from one surface to the other. At that stage, the problem is a two-dimensional Boolean operation where the intersection of the surfaces forms the shadow polygon which is then added to or, more accurately, unioned with, the existing shaded polygons. At any instance during the cycle of the algorithm, the areas of the insolated and shaded portions always add up to the original area of the parent surface. As an example, 82% of the surface shown in figure 2-C is insolated while 18% is shaded. The subtraction of the shadow polygon from the existing insolated portion forms the new resulting insolated portion. All polygons are then projected back to the location of the original surface (Figure 2-D).

As surfaces receive shadows, the insolated portion decreases and could disappear altogether. Should that happen, the surface in question is ignored for subsequent shadow casting which helps the algorithm proceed at a faster pace.

Once the algorithm has iterated through all the surfaces, the shadow and insolated portions are associated with the appropriate colors and displayed graphically to the user.

Some shortcuts are used to save processing time. For example, if the objects repeat as in a multi-storey building and the different levels do not cast shadows on each other, then one can calculate the shadows for one level, save the result, and then instantiate that level as many times as needed and at different locations (Figure 3).

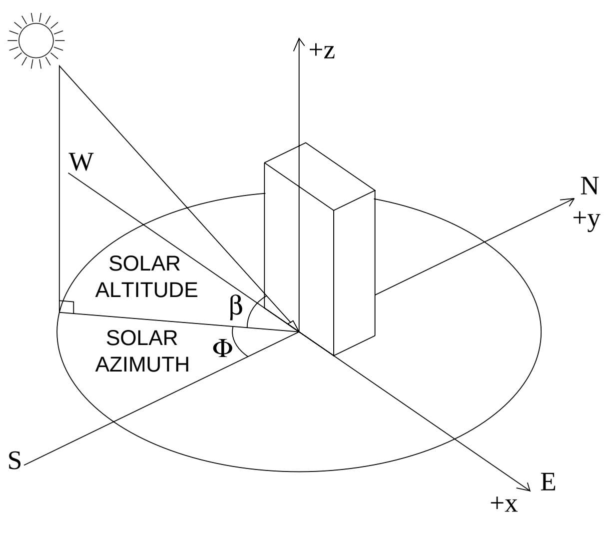

Fig. 1 The correspondence between the sun angles and the Cartesian coordinate system.

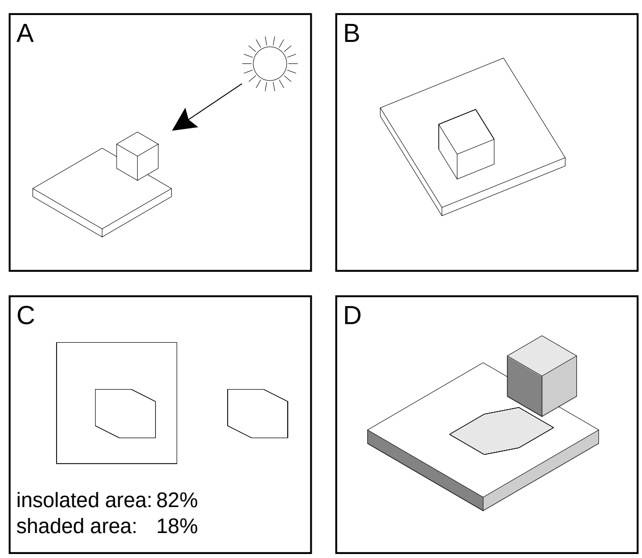

Fig. 2 A graphical depiction of the shadow casting algorithm.

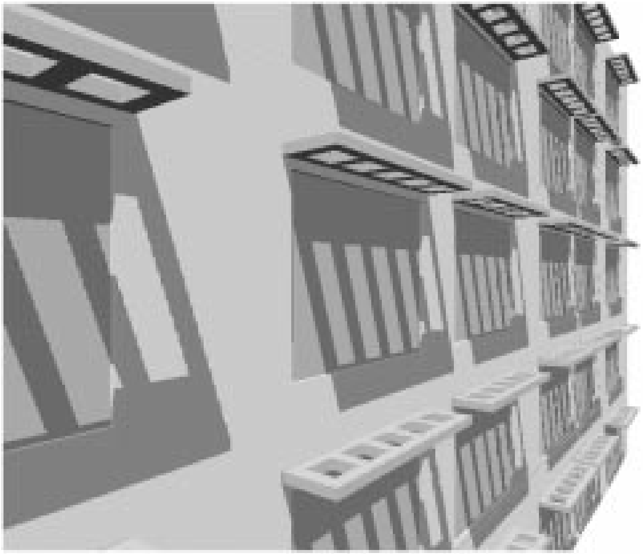

Fig. 3 A 3D model with shadows showing multiple Instantiations of one prototypical group of elements.

Using only Lambert's cosine shading (Foley and Van Dam, 1982), this algorithm does not strive for photo-accurate images as those produced by Radiance (Ward, 1994) or photo-realistic ones as those produced by the Renderman™ Shading Language (Upstill, 1989). Rather, its main advantage is its ability to geometrically determine the exact shape and location of shaded areas in three-dimensional space. This allows one to extract that information and perform other analyses that would not otherwise be readily feasible. In addition, this shadow casting algorithm is view-independent. In other words, one need only cast shadows once and then one can view the model, with shadows, multiple times and from varying points of view without the need for a re-calculation of the shadows. This can be achieved because the insolated and shadow polygons are saved in databases that can be re-accessed and redrawn multiple times.

3. THERMAL SIMULATION ALGORITHMS

The thermal simulation program used in this experiment was developed as part of an intelligent thermal optimization system (Malkawi, 1994), (Malkawi and Akridge, 1994). It uses the Transfer Function Method (TFM) as its engine for simulation. TFM was first introduced by Stephenson and Mitalas (Stephenson, 1967), (Mitalas, 1973). This procedure is based on response factors and the interplay of heat exchange between various surfaces and sources of heat gain (Romine, 1992). Transfer functions are based on two concepts: the conduction transfer factors (CTF) and the weighting factors (WF). The CTF are used to describe the heat flux at the inside wall, roof, partition, ceiling or floor as a function of previous values of the heat flux and previous values of inside and outside temperatures (McQuiston, 1992). The WF are used to translate the zone heat gain into cooling loads (Sowell, 1984). These functions are derived mainly from response factors. These response factors are defined as an "infinite series that relates a current variable to past values of other variables at discrete time intervals. A transfer function converts the theoretically infinite set of response factors into a finite number of terms that multiply both past values of the variable of interest and past values of other variables" (McQuiston, 1992).

The thermal simulation program contains two levels of analysis in its calculations. First the determination of the heat gain or loss the building produced based on the origin of the elements that produce these loads. Second, the conversion of this heat gain or loss into cooling or heating loads.

To compute the heat gain, the elements that produced it are considered. These elements are exterior surface opaque building material or any thermally massive interior surface that separates different temperature spaces, the exterior glazing area, lighting, occupants and equipment. In calculating the exterior opaque building materials, the TFM uses a series of conduction transfer function coefficients and weighting factors. It applies them to these surfaces taking into account the difference between outside and inside temperatures to determine heat gain or loss with appropriate reflection of thermal inertia of such surfaces. To take into consideration the changing conditions in the temperature of the outside air, the TFM uses the concept of sol-air temperature when calculating the difference between the outside and inside temperatures. These calculations are dependent upon the hourly calculations of solar intensity for each exterior surface.

The solar heat gain for glazing is calculated based on the transmitted solar heat gain, absorbed solar heat gain and conductive heat gain or loss. The transmitted and absorbed solar heat gain for each hour for each window is calculated based on the transmitted and absorbed solar heat gain factors. These calculations depend on the hourly calculation of solar intensity for each exterior surface. Conductive heat gain, on the other hand, depends on the flow of heat due to the difference between exterior and interior temperatures. Lighting, occupants and equipment heat gain are computed hourly. At the end of these calculations the total heating gain or loss for the building is calculated.

The next level of analysis involves converting these instantaneous hourly gains into instantaneous loads. At this stage, the TFM applies a second series of weighting factors, or room transfer functions to heat gain from all load elements having radiant components. This accounts for the thermal storage effect in converting heat gain into cooling loads. The room transfer functions are related to the geometric characteristics of the space and its configuration as well as mass properties. This allows the TF method to account for the interrelationship between elements that are responsible for the loads on a time basis. The infiltration

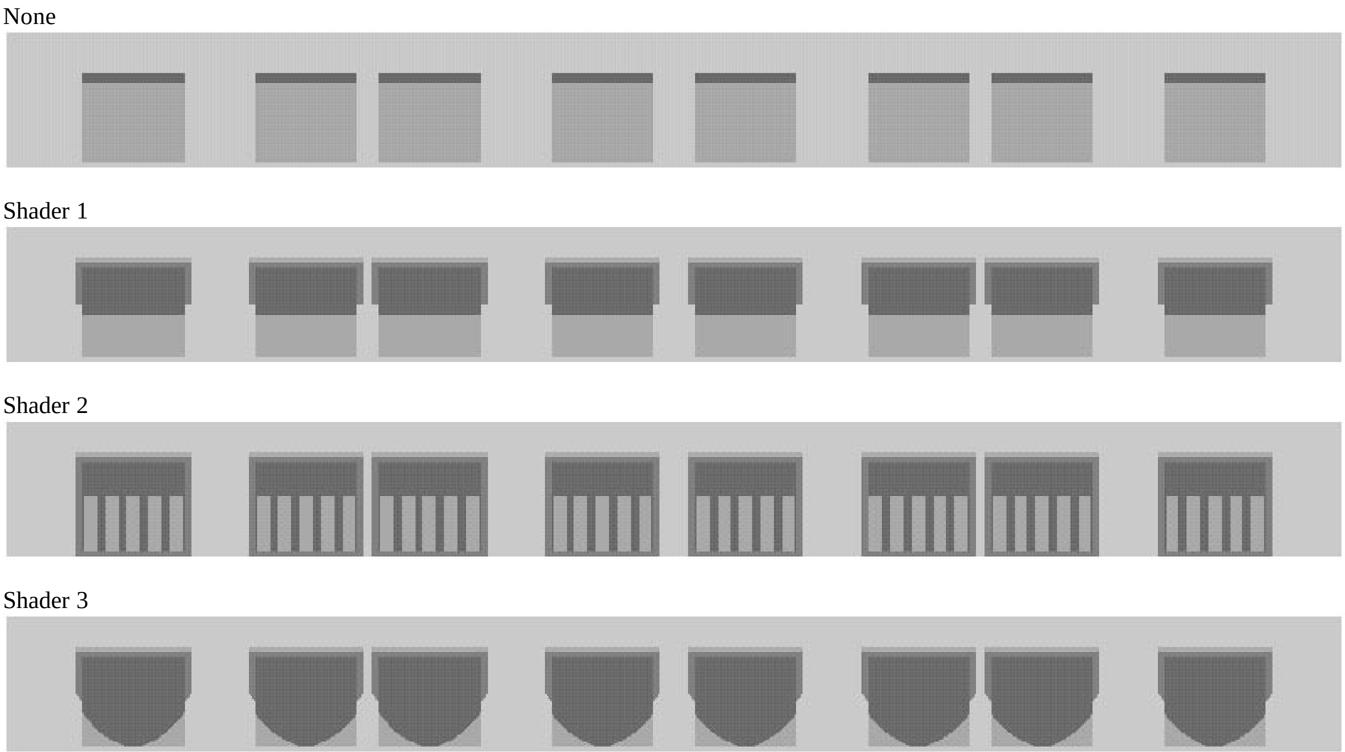

Fig. 4 Facades of the computer model with no shading device and three types of shading devices.

and ventilation hourly cooling loads are added to yield the overall building cooling load.

4. TEST CASE

Since the focus of this experiment was on the exterior glazing of an existing building, a computer model was built with a series of south-facing windows that have configurable shading devices (Figure 4). Once the location, date, time, position, area and orientation of the model were determined, the graphical result was displayed on the screen and the data was channeled to the simulation algorithm that calculated the cooling loads (Table 1). The graphical result readily indicated that the geometry of the shading devices used greatly impacts the total visual perception of the facade. These impacts are important for the designer to achieve a balance between visual characteristics and thermal behavior. For example, while the visual characteristics of the shadow created by devices 1 and 2 significantly differ, their thermal behavior is almost equal. This is due to the fact that their shaded areas are almost equal (Table 1).

5. INTEGRATION

To integrate the shadow casting method and thermal simulation, the thermal simulation algorithms were modified to accept input from 3D shadow procedures. TFM calculations require the determination of all loads entering the space, the radiative and convective interactions between them and the time lag involving storage and release of energy in thermal mass. Given the focus of the experiment, only the glazing modules and their algorithms were modified to accept the input from the 3D shadow procedures.

TABLE 1. MODEL ATTRIBUTES AND SIMULATION RESULTS FOR A SAMPLE HOUR

Location	Atlanta, Georgia
Date	June 21
Solar Time	12:00 noon
Solar Azimuth	0.00^{o}
Solar Altitude	76.90^{o}
Sun Vector	(0.000, -1.000, 4.297)
Surface Orientation	South
Surface Area	1822.38 ft^2
Wall Area	1066.38 ft^2
Window Area	756.00 ft^2

Shadow and Thermal Simulation Results				
	None	Shader 1	Shader 2	Shader 3
Insolated Area (ft^2)	669.43	350.49	336.29	91.66
Shaded Area (ft^2)	86.57	405.51	419.71	664.34
Cooling Load (Btu)	23751.90	20457.00	20341.70	18354.00

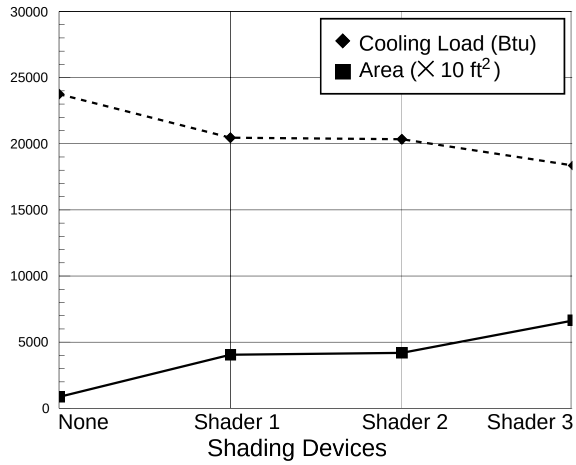

Fig. 5 Chart showing correspondence of shaded glazing areas and glazing cooling loads.

The solar heat gain for glazing is calculated based on the transmitted solar heat gain, absorbed solar heat gain and conductive heat gain or loss. The transmitted and absorbed solar heat gain depend on the hourly calculation of solar intensity for each exterior surface which takes into account the shading of the surface. Conductive heat gain, on the other hand, depends on the flow of heat due to the change of temperature between outside and inside. The 3D shadow procedures output precise shaded and insolated window areas and channel them to the thermal simulation to be used in the calculation of both transmitted and absorbed solar heat gain for each hour for each window. The conduction heat gain is not affected by the shadows. To convert these heating loads to cooling loads, a series of different WF are applied. Using the transmitted solar heat gain and the *solar* WF, the program calculates the hourly cooling loads due to transmitted solar heat gain. The absorbed solar heat gain and the window conduction heat gain for each hour for each window uses the *conduction* WF coefficients to calculate the hourly cooling loads due to both conduction heat gain and absorbed solar heat gain (Figure 5).

6. CONCLUSION

Heat gain through the building envelope is complicated by the wide cyclic variation in the outside air temperature and sun intensity, therefore the heat flow through the building envelope is not constant. Transparent materials, such as glass, have low mass and therefore the amount of heat transfer is more a function of thermal radiation characteristics than thermal storage. As a result, the accurate prediction of the impact of exterior shading devices on such surfaces influences the accuracy of predicting their thermal performance. The experiment showed the feasibility and advantage of integrating advanced algorithms in both thermal simulation and shadow casting methodology to enhance the simulation accuracy and provide visual feedback to designers. The experiment demonstrated the advantages of using a computer-aided graphical representation of shadow casting for both establishing more output accuracy and allowing complex shapes to be simulated. In addition, the experiment revealed the importance of integrating different algorithms that use rigorous computer representations. Such an integrated tool can help the designer better assess the visual and thermal characteristics of the design solution and provide a robust computerized design aid for energy conservation.

7. ACKNOWLEDGMENTS

The authors wish to thank Professor Harold Borkin, Professor James Turner, and Dr. Theodore Hall for developing the GEDIT solids modeling system and for their advice in developing the shadow casting algorithm.

8. REFERENCES

(1) Borkin, H., McIntosh, J., Turner, J., The Development of Three-Dimensional Spatial Modeling Techniques for the Construction of Nuclear Power Plants, Computer Graphics: A Quarterly Report of SIGGRAPH-ACM, The Association for Computing Machinery, August 1978

(2) Foley, J. and Van Dam, A., Fundamentals of Interactive Computer Graphics, Addison-Wesley, 1983

(3) Jabi, W., SHADOW (Master of Architecture Thesis), College of Architecture and Urban Planning, The University of Michigan, 1989

(4) Malkawi, A., Akridge, M., Intelligent Computer-Aided Building Thermal Design, Proceedings of the 19th National Passive Solar Conference, American Solar Energy Society, 1994

(5) Malkawi, A., Building Energy Design and Optimization: Intelligent Computer-Aided Theoretical Design, (PhD. Dissertation), Georgia Institute of Technology, 1994

(6) McQuiston, F.C. and Spitler, J.D., Cooling and Heating Load Calculation Manual, 2nd ed., American Society of Heating, Refrigerating and Air Conditioning Engineers, Inc., 1992

(7) Mitalas, G.P., Calculating Cooling Load Caused By Lights, ASHRAE Journal, June 1973

(8) Romine, T.B., Jr., Cooling Load Calculation: Art or Science?, ASHRAE Journal, January 1992

(9) Sowell, E.F. and Chiles, D.C., Zone Descriptions and Response Characterization for CLF/CLTD Calculations, ASHRAE Transactions, 1984, Vol.91, Pt.2A

(10) Stephenson, D.G. and Mitalas, G.P., Cooling Load Calculations by Thermal Response Factor Method, ASHRAE Transactions Vol. 73, Pt.1, 1967

(11) Turner, J., GEDIT User's Manual, College of Architecture and Urban Planning, The University of Michigan, 1993

(12) Upstill, S., The Renderman™ Companion, Addison-Wesley, 1989

(13) Ward, G., The RADIANCE Lighting Simulation and Rendering System, Computer Graphics: A Quarterly Report of SIGGRAPH-ACM, The Association for Computing Machinery, August 1994