Experimental System for Visualisation of the Light Load

Martin Čadík
Czech Technical University
Faculty of Electrical Engineering
Karlovo náměstí 13
121 35 Prague, Czech Republic
cadikm@cs.felk.cvut.cz

Pavel Slavík
Czech Technical University
Faculty of Electrical Engineering
Karlovo náměstí 13
121 35 Prague, Czech Republic
slavik@cslab.felk.cvut.cz

Jan Přikryl
Czech Technical University
Faculty of Transport Sciences
Na Florenci 25
11000 Prague, Czech Republic
prikryl@fd.cvut.cz

ABSTRACT

This paper presents our work on an experimental system for visualisation of the light load. The light load is defined as the total amount of light radiation received by all areas of a 3D scene. The emphasis of the presented system lays on outdoor architectural scenes, but indoor scenes are handled as well. The aim of the system is to visualise the light load either in a single moment or integrated during a longer time period. We have selected a hierarchical Monte Carlo radiosity method to solve the specified problem. This method was extended to handle parallel light sources and specular reflections. New "lighting" iteration was added to computation phase to consider natural light sources such as Sun and sky. The system was implemented in Java programming language using the Java3D API. Thanks to this implementation environment, our system is flexible, easy to modify and extend and it is suitable for experimental and educational purposes.

Keywords

Computer graphics, global illumination, radiosity, ray tracing, Monte Carlo techniques.

1. INTRODUCTION

The aim of our work was to develop an experimental system for physically accurate global light propagation simulation and visualisation of the light load mainly of outdoor architectural scenes. Light load is defined as the amount of light radiation received by the areas of the solved environment. The input of the system is a 3D model of a real scene and definitions of light sources (their position, intensity etc.). The emphasis is given on obtaining view-independent solution that visualises the light load, in the form of a 3D scene suitable for presentation purposes.

2. PREVIOUS WORK

Existing software for physically accurate solution of global illumination problems is typically hard to work with. It requires a lot of knowledge of the problem, it is usually very complex, expensive and it is intended mainly for indoor scenes. Majority of this software creates only images or animations and does not produce the output in a 3D scene format where we can investigate the situation more in detail, from different viewpoints.

For experimental and educational purposes we need a relatively simple, transparent and flexible system that is easy to maintain, modify and extend. This led us to development of our own system based on Java3D paradigm. Using Java makes the system portable to various hardware platforms and easy to modify.

3. METHODS FOR SOLVING THE LIGHT DISTRIBUTION IN A SCENE

Physically accurate global light propagation simulation requires solving the *rendering equation* [Kaj86a]. A solution may be obtained by two principal approaches. The first one could be classified as the group of ray tracing based methods, the second one is the group of finite element methods solving the rendering equation on a finite element mesh for purely diffuse scenes.

Basic ray tracing [Whi80a] is a point-sampling technique. It works with rays of the light that enter the observer's eye, which are traced in backward direc-

WSCG POSTERS proceedings
WSCG'2003, February 3-7, 2003, Plzen, Czech Republic.

tion, towards the light sources. Ray tracing based methods excel when computing scenes with point light sources, direct illumination, specular reflections and refraction through transparent materials. They typically give the result in the view-dependent 2D form and are relatively memory efficient.

Classical radiosity methods produce a view-independent 3D solution — rather then computing pixel values on the screen, these methods calculate intensities for elements in a 3D environment. Approaches like *progressive refinement* [Coh88a] are fast and they eliminate difficult and time-consuming pre-processing that was necessary with the first radiosity methods. The *hierarchical radiosity* [Han91a] and the *wavelet radiosity* [Gor93a] create a hierarchy of elements and the fine initial tessellation of the input scene is not necessary. Contemporary approaches (based either on stochastic radiosity [Gra01a] or on photon mapping [Jen02a]) to global light simulation join positive qualities of the methods of both mentioned groups.

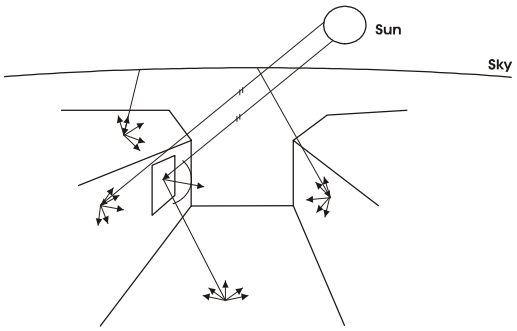

Figure 1. Light conditions in an outdoor architectural scene. The Sun and the sky are essential light sources. Diffuse reflections are prevalent, but specular reflections are not negligible.

4. SOLUTION

The basic observation is that the light load is view independent. This implies that we can get the solution in the form of a static 3D scene. Architectural scenes (see Fig. 1) contain many flat surfaces that can be approximated by diffuse elements. Light load for such scenes could be exactly determined with the radiosity method. However, it is necessary to represent also specular reflections and non-diffuse light sources which are not possible to neglect. The selected radiosity method should be therefore also suitable for incorporating these new extensions. We believe that the *hierarchical radiosity* based on *stochastic Jacobi iterative method* [Bek98] meets best these requirements. Other possibility would be using some kind of a multipass method.

Hierarchical Monte Carlo radiosity is a combination of two popular approaches in the radiosity calculation — *hierarchical radiosity* and *stochastic computation*. Hierarchical radiosity [Han91a] contributes with automatic adaptive subdivision of surfaces and with strong reduction of form factors that have to be computed. *Monte Carlo radiosity* excels with low memory requirements (it is not necessary to explicitly evaluate and store the form factors). It can achieve higher stability of the computation and is less sensitive to errors caused by missing or badly computed occlusion. The usual drawback of stochastic methods is the high frequency noise. As the stochastic part of the method, we selected the *stochastic Jacobi iterative method* [Bek99a]. This method is a good candidate to be used in practice, because the computational cost is related to the number of samples that need to be shot rather than to the number of elements in the scene.

Extensions to the hierarchical Monte Carlo method

Specular Reflections

Smooth surfaces reflect the incoming light in a single direction — at an angle equal to the incident angle at which they arrive to the surface. Specular surfaces like glass, chrome, metals and so on are very usual in architectural scenes. Is it therefore necessary to extend our pure diffuse method to handle specular reflections correctly: Every object in our scene has its own specular reflection coefficient ρ_S. For each light ray hitting an object with $\rho_S \neq 0$ the recursive *ray tracing* is performed. The ray tracing finishes when the ray reaches a surface with $\rho_S = 0$ or when the depth of recursion is equal to the predefined fixed value.

Parallel Light Sources

An extension that includes parallel light sources is necessary to allow the user to model very distant light sources. This extension is straightforward: a parallel light source emits particles in direction of its normal vector, while a diffuse light source emits photons according to $\cos\theta$ distribution.

Lighting Iteration

Sun and sky characteristics differ a lot from other synthetic light sources so they are treated separately in our system. For computation of effect of Sun and sky, we inserted new initial iteration into the iterative process. During this lighting iteration, the energy of all parallel light sources in the scene is shot out. There are different thresholds and other computational parameters involved, and the oracle function (see [Coh93a] for explanation) differs from the one used in normal iterations. The distance of interacting elements is not considered in the lighting oracle function and the division parameter used during the lighting iteration can have a different value. After all the energy of the global light sources has been shot out, the gathered energy on the elements is considered as the self-emission of the element. In the following iterations, the Sun and the sky are ignored. A

similar approach in context of progressive radiosity and pure hierarchical refinement is described by Mülle and colleagues [Mul95a] and Daubert et al. [Dau97a], respectively.

Approximation of the Sun's path

Integral part of our solution is investigation of the influence of the sun *trajectory* on the light load in the given scene configuration. The Sun's relative position to the place on Earth changes during the day. The form of sun trajectory is also dependent on the season of the year and latitude and longitude on Earth.

We use the CIE [CIE73] empirical formulas to construct Sun and sky elements. These elements are sampled with a quantum of time Δt for a given time interval. The result is approximated as the superposition of contributions of time samples. Thanks to this we can visualise the light load during some time interval for example during the whole day.

Visualisation

One of the most important tasks for a lighting designer is to determine the luminance levels in the environment that is being designed or measured. Hence, our primary goal is the visualisation of the light load. We display scene surfaces in false colours that corresponds to the level of irradiance (W/m^2) or light load (J/m^2) in the case of longer time period visualisation. Visualisation is performed in the 3D space and the scene can be saved in the VRML [ISO97a] format. It can be used later in order to investigate the results, perform walkthroughs, or compare different illumination configurations.

5. IMPLEMENTATION

Our experimental system was implemented in the Java programming language using the standardised Java3D application programming interface (API). This makes the system portable to various hardware platforms and easy to modify.

Java3D

All Java3D programs are based on the Scene Graph data structure. Scene Graph is a tree-like data structure that describes entire scene. It consists of two main branches – a viewing branch and a content branch. The content branch cares about geometry, appearance, behaviour and so on, while the viewing branch is intended to set up and change the viewing parameters.

Data structure of our System

While loading an input 3D scene, the scene graph is constructed. It consists predominantly of Shape3D objects representing shapes formed by triangle elements. As we need to store the hierarchy of elements that grows in consequence of subdivision during computation, we place it as a user data directly to Shape3D objects, see Fig. 2.

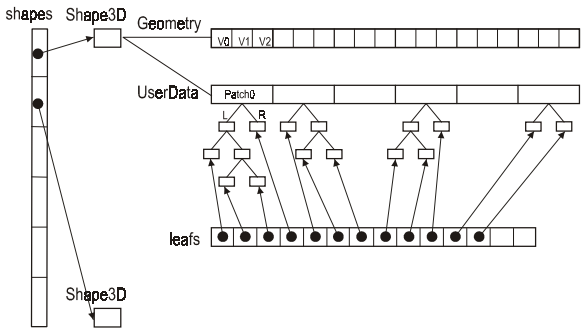

Figure 2. Data structure of the system. Shape3D represents a 3D object. It is a collection of triangles. Triangles are defined by vertices. Hierarchy is maintained as a user data.

Acceleration of the Java3D Ray Casting

As with any other computation method that is based on ray casting, the speed of a ray-primitive intersection test has an important influence on the overall speed of an implementation. Java3D native methods for ray-object intersections are quite inefficient. For us this meant writing an optimised ray-triangle intersection routine, such as the one of Badouel [Gla90a]. We also have to keep the time-critical sections free of object instantiation and allocation.

We have first replaced the original ray-picking tests by open-source routines for ray-triangle intersection available from org.j3d group [J3d02a]. However, ray intersection routines in org.j3d package need to recompute many object parameters from scratch, which still brings some performance penalty. Using a variant of Badouel's algorithm [Gla90a], where all important constants have been precomputed brought us an average speedup of 2.2 against original Java3D implementation.

6. RESULTS

In order to compare whether the convergence of the implemented method corresponds to the expected convergence rate of stochastic Jacobi relaxation we carried out several measurements on various scenes. These scenes included simple indoor scenes and also outdoor scenes.

Figure 3 demonstrates the capability of the system to account for specular reflections in outdoor scenes. The left image on Figure 3 shows how our system handles the computation of light load for a sun-lit scene where the sun moves over the sky for approximately one hour. One can see that the illumination during this rather short time period did not completely wash out the borders between reflections on the street.

7. CONCLUSIONS & FUTURE WORK

In this paper, we presented an experimental system for visualisation of the light load. We made several extensions to the hierarchical stochastic radiosity method: our method can handle parallel light sources and specular reflections. A new "lighting" iteration was added to computation phase to consider natural light sources such as the Sun and the sky. The system was implemented in Java programming language using the Java3D API.

The selection of the computation method gives some specific characteristics to our application. The result obtained is a view-independent representation of the whole 3D scene, VRML format is suitable for presentation and inspection purposes and is also acceptable for the web environment.

Possible extensions of the system are: output of the animation of light proportions during the day, participating media. The system will be extended to calculate thermal load of the buildings and to take into account other effects influencing the habitant's environment and their feelings.

Acknowledgements

This project has been partly supported by the Ministry of Education, Youth and Sports of the Czech Republic under research program No. Y04/98: 212300014 (Research in the area of information technologies and communications).

8. REFERENCES

[Bek98] Bekaert, P., Neumann, L., Neumann, A., Sbert, M., and Willems, Y. Hierarchical Monte-Carlo Radiosity. In Rendering Techniques '98, 1998.

[Bek99a] Bekaert, P. Hierarchical and Stochastic Algorithms for Radiosity. PhD thesis, Katholieke Universiteit Leuven, 1999.

[CIE73] CIE Technical Committee 4.2. Standardization of luminance on clear skies. CIE Publication No. 22, Commission International de l'Eclairaze, Paris, 1973.

[Coh88a] Cohen, M., Chen, S. E., Wallace, J. R. and Greenberg, D. P. A progressive refinement approach to fast radiosity image generation. Computer Graphics (SIGGRAPH '88 Proceedings), 1998.

[Coh93a] Cohen, M. F. and Wallace, J. R. Radiosity and Realistic Image Synthesis. Academic Press Professional, Boston, MA, 1993.

[Dau97a] Daubert, K., Schirmacher, H., Sillion, F. and Drettakis, G. Hierarchical Lighting Simulation for Outdoor Scenes. In Rendering Techniques'97, NY, 1997.

[Gla90a] Glassner, A. S. Graphics Gems. Academic Press, Inc., San Diego, CA, 1990.

[Gor93a] Gortler, S. J., Schröder, P., Cohen, M. F. and Hanrahan, P. M. Wavelet radiosity. Computer Graphics (SIGGRAPH '93 Proceedings), 1993.

[Gra01a] Granier, X. and Drettakis, G. Incremental Updates for Rapid Glossy Global Illumination. In Proceedings of EUROGRAPHICS'01, 2001.

[Han91a] Hanrahan, P., Salzman, D., and Aupperle, L. A rapid hierarchical radiosity algorithm. Computer Graphics (SIGGRAPH '91 Proceedings), 1991.

[Hav00a] Havran, V. Heuristic Ray Shooting Algorithms. PhD thesis, Faculty of Electrical Engineering, Czech Technical University in Prague, 2000.

[J3d02a] The Java3D Community Site. http://www.j3d.org, 2002

[Jen02a] Jensen, H. W. and Buhler, J. A Rapid Hierarchical Rendering Technique for Translucent Materials. (SIGGRAPH'02 Proceedings), 2002.

[Kaj86a] Kajiya, J., T. The Rendering Equation. In Computer Graphics (SIGGRAPH '86 Proceedings), 1986.

[Mul95a] Müller, S., Kresse, W., Gatenby, N. and Schöffel, F. A Radiosity Approach for the Simulation of Daylight. In Rendering Techniques'95, NY, 1995.

[Smi00a] Smits, B. and Jensen, H. W. Global Illumination Test Scenes. Technical Report UUCS-00-013, University of Utah, 2000.

[Whi80a] Whitted, T. An improved illumination model for shaded display. Communications of the ACM 23 (6): 343-349, 1980.

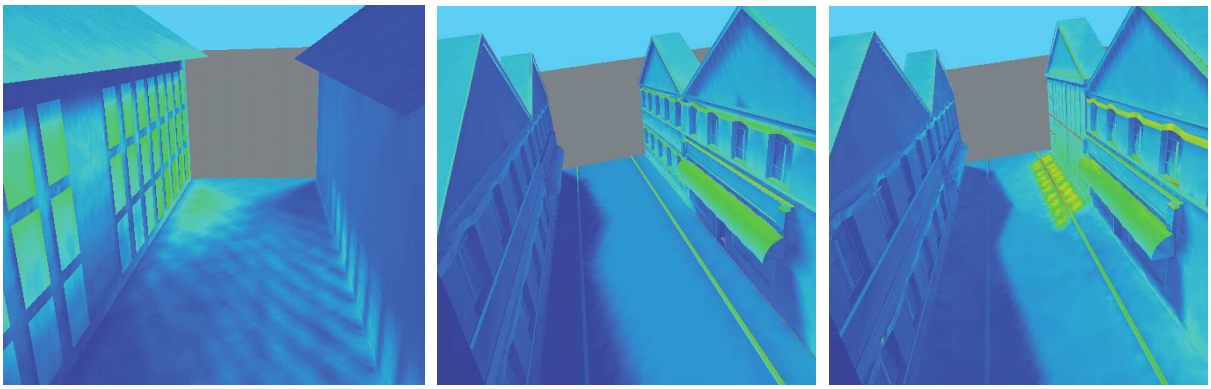

Figure 3. Light load for two simple architectural scenes.
On the left, we can see a cumulative result for 1 hour during a sunny day.
The middle image shows light load in another street, the right image shows light load in the same street after replacing an old building on the right side by the new one made of materials that are more reflective.