Ray Maps technique for effective interrogation of results of MCRT simulation

B.Kh. Barladyan, L.Z. Shapiro, A.G. Voloboy
Keldysh Institute of Applied Mathematics RAS, Moscow

Abstract

The new lighting simulation result interrogation technique called Ray Maps is described. These maps can be effectively used in physically accurate optic simulation system based on forward and backward Monte-Carlo ray tracing. In particular the Ray Maps can increase efficiency for photo-realistic images creation, fast analyze of light characteristics on the radiation detector with variable parameters, studying the fine details of the light propagation in the complex scenes and so on. The example of Ray Maps usage is Ray Visualization intended for the complex optical system design. This example is considered in details in the paper.

Keywords: *Monte Carlo ray tracing, Lighting simulation, Global illumination, Photo-realistic images, Interactive scene analysis, Ray Maps*

1. INTRODUCTION

Monte-Carlo ray tracing, both forward and backward ones, is widely used in various systems for physically accurate simulation of light propagation in complex scenes [Pharr 2004]. These methods are used for simulation of global illumination, photo realistic images generation and simulation of complex optical devices. Optical simulation of light transportation using Monte Carlo ray tracing usually requires considerable time. Long calculation time is especially critical if Monte Carlo ray tracing is used in rendering like described in [Hashisuka 2009]. In such a case the additional techniques should be elaborated to improve the effectiveness of the user interaction with such system.

The main idea of our approach is to store the set of rays traced by the Monte Carlo method together with the necessary supplementary data. After that the other subsystems of the optical simulation can be used for quick interrogation of calculated results as well as the ways of light propagation in a scene. This stored data can be called Ray Maps. It should be pointed that the set of the supplementary data to be stored together with the array of traced rays depends on the goal for which these data will be used. The volume of the stored data can be very large and the efficiency of the following processing strongly depends on this volume. The storing format of this data also is critical in the terms of further processing efficiency and therefore should be elaborated taking further processing into account. The present paper considers in detail only Ray Maps creation during forward Monte Carlo ray tracing and their post processing for the purpose of ray visualization system. Although the proposed approach can be used for other goals as well. Certainly the data set and the storage format depends on the certain purpose. More precisely it is determined by the post processing effectiveness. In the case when the stored data are used frequently (for example, the realistic images creation for several camera positions is the typical task) the access to the data should be accelerated by using special accelerated structures like, for example, BSP tree [Havran 2000].

2. DATA FOR RAY VISUALIZATION SYSTEM

The ray set in our Ray Visualization system [Voloboy at al 2009] is stored as segments set and contains the following data:

1. Segment color. The Monte-Carlo simulation in our system can be done in two color spaces – RGB and spectral ones. Even if the spectral color mode is used during simulation the user can request to store Ray Maps in the RGB format. This allows decreasing the stored data size. In the same time for ray visualization purpose the RGB format is sufficient. The spectral mode of segment representation which is also possible may be needed for more detailed investigation.
2. The light source emitted the given ray.
3. The geometry object from which the ray segment starts (if exists).
4. Triangle index for the geometry object from which the ray segment starts.
5. 3D point coordinates for the segments start and end points.
6. Normal to the surface in the ray/surface intersection points.
7. Optical surface properties for intersected surfaces.
8. Ray transformation type on the given surface – diffuse refraction/reflection, specular refraction/reflection, Fresnel refraction/reflection, BRDF/BTDF diffuse and specular refraction/reflection, absorption on the surface.
9. Transformation type in the medium – absorption or ray scattering.
10. Optical properties of the medium along the ray propagation.
11. Virtual radiation detectors intersected by the given ray segment and parameters of the given intersection.

This data is used for the two purposes:

1. For checking ray visualization criterion – only rays satisfied to the criterion defined by user should be visualized.
2. For Ray Tracing Report creation. User can investigate the history of the ray path in details in the report where information about all visualized rays satisfied to the given criterion is collected. In particularly the user can be interested in light source which emit the given ray, in transformation which occurred on the surface or in the media, in ray segment color (RGB and spectral), in surface/medium optical attributes and so on.

3. DATA STRUCTURES AND STORING

The number of ray segments created during Monte Carlo ray tracing can reach millions or even billions in some cases. So the Ray Maps created by Monte Carlo ray tracing have to be stored on the hard disk during simulation and then loaded for analyzing as needed. It should be pointed that while the most part of the segment data of the traced ray has fixed size, the data described the intersections with the radiation detector has variable length – it is possible that the given ray segment does not intersect the radiation detectors at all or does intersect several such detectors. The data describing the ray segment color has variable length also because the spectral color description as well as RGB color description is possible according user request. Moreover in common case the user can switch color mode storing in any moment during simulation. In the result one part of ray segments will contain the spectral data while another one – RGB data. Due to variable length of additional data each portion of traced rays is stored in the three parts. The first part contains basic ray segment descriptions of fixed length and links to variable additional data. The structure of basic description contains the following data:

1. Segment color in RGB format. The RGB color is needed to visualize the ray trajectory on display.
2. Index of light source emitted the ray.
3. The index of object from which the ray segment starts.
4. The index of the triangle for an object from which the ray segment starts.
5. 3D coordinates of the beginning of the ray segment.
6. Normal to the surface in the ray segment beginning point, if it corresponds to ray/surface intersection.
7. Ray transformation type on the surface or in the medium which cause the given ray segment (in particular, it can be absorbed on the surface, in the medium or the ray can left the scene).
8. Index of the first descriptor of the ray intersections with the radiation detectors.
9. The number of detectors intersected by the ray segment.

The second portion part contains a description of ray (segment) intersections with radiation detectors. It is an array of structures of fixed length. This structure contains the following data:

1. Index of radiation detector
2. Radiation detector cell index
3. The angle between the ray and the normal to the plane of the radiation detector.
4. The 2D coordinate of intersection of ray segment with radiation detector plane (local coordinates of the intersection point in the rectangle of the radiation detector coordinate system)
5. Was the intersection registered by the radiation detector or not.

This structure is used for each ray intersection with the each radiation detector. The array of these structures together with fields 8 and 9 in the previous structure provides information about ray/detector intersections in relatively compact form.

The last third part of the recorded data for each portion contains a description of a spectral color of ray segments in case of the spectral simulation was performed and storing of spectral data have been requested by the user.

The described above data provides storing the results of Monte Carlo ray tracing in compact form. In the same time the data contains the complete information necessary for the post processing. During ray visualization and analysis user can set different criterion for the ray visualization. The criterion consists of the following set of elementary events:

1. Ray was emitted by the specified light source.
2. Ray intersected the specified face of the optical device.
3. The ray intersected the pointed part of a geometric object of optical system, and the specified optical ray transformation took place.
4. The ray intersected the specified radiation detector or some its area.
5. The ray intersected any surface which have the specified set of optical properties, and the specified optical ray transformation took place.
6. The ray undertook the specified optical ray transformation (diffuse or specular reflection, absorption, etc.) on the one of the objects of the optical system.

User can apply for these events, their logical negation, intersection ($\cap$) and union ($\cup$) with an unlimited number of operands. In the result only the rays that satisfy to final logical expression will be visualized.

The user can also interactively change the conditions of registration of radiation at the radiation detector, for example, the resolution of a radiation detector, the angle of registration, show results only for the given light source. Also it is possible to obtain the different statistical data for rectangular and elliptical region of the radiation detector.

The changing of the ray visualization criterion or parameters of the radiation detector does not require new simulation, because the all necessary data are already presented in the saved data (Ray Maps) and can be processed relatively quickly to obtain the requested information.

There is an important aspect concerned to the effective simulation result saving in the form of Ray Maps – the size of file with stored data. For the 1 million rays (in average ~5 million segments) it requires ~220Mb if simulation was done in RGB space, and ~540Mb in spectral simulation mode with 33 wavelengths. Here the main problem is not in the required disk space, but in the time (speed) of the recording itself. It should be also taken into account that the Monte Carlo ray tracing typically uses multi-threading and distributed computing on multiple computers (the number of threads, used in Monte Carlo ray tracing is the total number of processors on all computers involved in simulation) while recording (and reading) typically uses the single thread. So in some cases storing the data to the disk becomes a bottleneck that limits the speed of simulation.

To overcome this problem in our system the data compression is used before writing it to disk. Directly Monte Carlo ray tracing is performed in our system by portions of a number of rays. The portion size is tuned during simulation depending on the different parameters such as the required accuracy, the ordered number of rays, number of available processors etc. Each portion is calculated in a separate thread (in general case on different computers) almost independently. In this situation we applied the compression for each portion of the traced rays in calculating threads. Already compressed data is transferred to the main thread for storing to the disk. As a result the storing the data to disk became shorter and does not de-accelerate lighting simulation, at least for the typical case of four calculations threads in one computer. After compressing the file size in the case of simulation in RGB space was decreased from 220Mb to 50Mb and in spectral case from 540Mb to 51Mb. The total simulation time practically was not affected in RGB case and accelerated in spectral case. The time spend on the compression performed in parallel in multiple threads were compensated by the acceleration of the data writing on the disk performed by the single thread. To compress the data the free cross-platform data compression library, created by Jean-Loup Gailly and Mark Adler, zlib [Zlib 2010] was used.

4. RESULTS

Ray Maps creation and further its processing in Ray Visualization module was implemented in our lighting simulation system [Voloboy, Galaktionov 2006].

During testing on many scenes it was confirmed that the time spent for the Ray Maps writing to the disk performed in a separate thread is almost negligible for the forward Monte Carlo rays tracing executed in parallel threads. For example if four threads (Intel Core2 Quad Q9550 2.83GHz) are used for Monte Carlo ray tracing and one more thread for data storing in uncompressed state then we often get situation when four calculation threads have to wait until storing thread has completed writing of previous data portion. As the compression reduces the size of recorded data in ~4.5 times for RGB mode and in ~10 times in spectral mode the data storing thread provides writing all necessary data to disk without suspending the work of simulation threads. The situation is explained by the fact that the processor spent small fraction of time for writing data to a disk.

It should be noted that the computational threads spend some time on compression. For very simple scenes and simulation in RGB mode compression time can be up to 50% of simulation time. But time spent for compression is determined mainly by number of segments and wavelengths for a ray portion while simulation time is determined by scene complexity. For example for simulation in spectral mode only 25% of CPU time is spent for compression. For complex real scenes this per cent decreases more. These costs are justified if four processors are used which is typical for modern computers. According to our estimates a recording thread will provide effective work till 6 - 8 simulation threads especially when the spectral simulation will be done with a large number of wavelengths.

The following results were obtained for typical scenes for simulation in RGB space (on 1000000 rays, ~5000000 segments):

1. Monte Carlo tracing time is and ~5-10 seconds in case of simulation of typical scene with surfaces described by complex optical attributes (BRDF).
2. Post processing. Loading time is 1.6 sec without compression. Compression (decompression is needed during loading) increases the whole loading time to ~2.3-2.4 seconds.
3. Post processing. Criterion analyzing time is ~1.2-1.3 seconds if a single elementary event is used. Each additional elementary event used in criterion increases processing time on ~0.5-0.6 seconds. This time is required if all 1000000 rays will be processed. Typically only small part of rays should be processed to obtain requested number of output rays satisfying to the specified criterion.

Therefore the proposed Ray Maps technique is an effective tool for interrogation of simulation results calculated by Monte Carlo ray tracing. This is important for design of complex optical devices. The biggest benefit of the proposed approach takes place in the spectral simulation and if optical properties of surfaces are described by the complex bidirectional reflectance distribution functions (BRDF) of large size. Another case when advantage of our approach is significant is volume scattering in a complex multi component media. In all cases when the time of simulation by the Monte Carlo ray tracing is large and the results should be used multiple times the usage of Ray Maps increases efficiency of the simulation system.

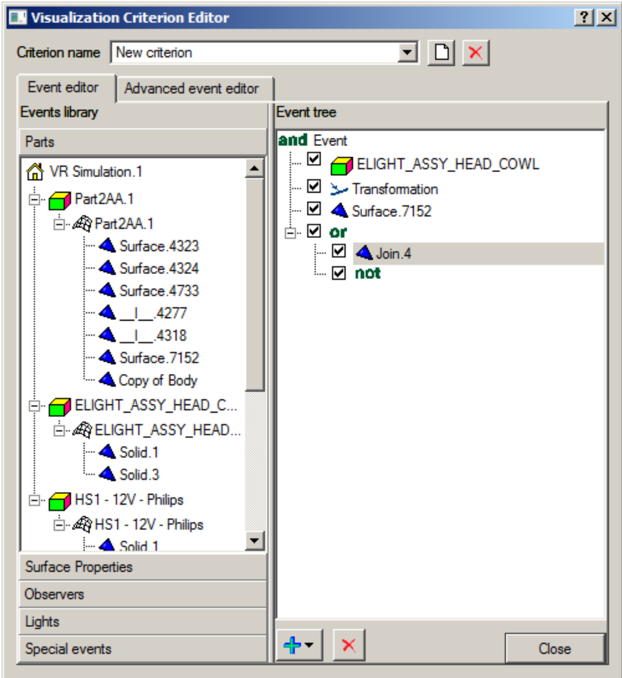

Figure 1: Visualization Criterion Editor.

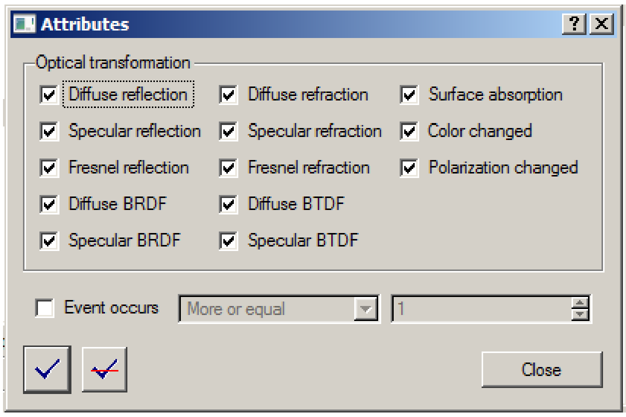

Figure 2: Optical Transformation Attributes.

The **Figure 1** and **Figure 2** illustrates the user possibilities to control the set of visualized rays by using the **Visualization Criterion Editor.** The user can set different criteria for ray visualization. The following set of elementary events can be selected in the dialog on **Figure 1:**

- light source emitted ray,
- intersection with specified geometry part,
- intersection with the surface with specified optical properties,
- special events along the ray path (killed ray or transformation).

For events it is possible to select special type of transformations by using the dialog presented on the **Figure 2**.

Also it is possible to construct logical expression from elementary event as it is shown in right part of dialog on **Figure 1**.

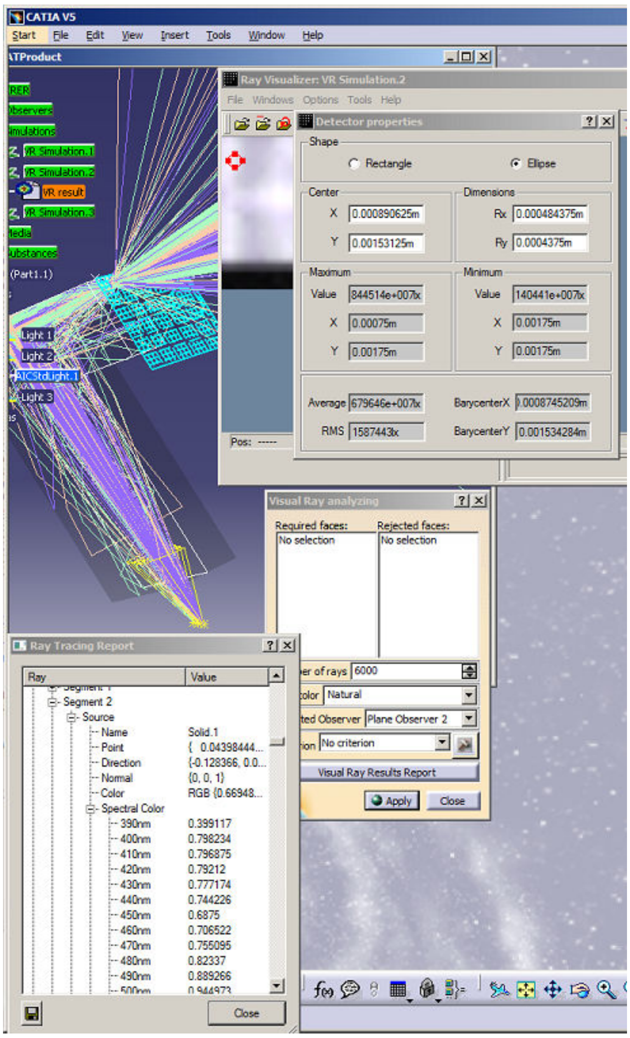

Figure 3: Our ray visualization system in CATIA.

The **Figure 3** illustrates the elaborated by us plugin for industrial CAD system CATIA [CATIA]. The dialog of radiation detector in the top-right corner lets you to choose the rays passing via the specified detector area additionally to the ray criterion specified as it is shown on the **Figure 1** and **Figure 2**. The **Ray Tracing Report** located at the lower left corner of the image provides detailed analyzing of the events, the optical properties of surfaces and media along the ray trajectory. All rays in this report correspond to the displayed rays and satisfy to the criterion set by user. The ray segment color can be represented both in RGB and spectral formats.

5. FUTURE WORK

Currently the suggested approach is used in the ray visualization subsystem. We suppose to adapt the Ray Maps technique for photo realistic images generation for scenes with complex optical surface and media properties. We also plane to use Ray Maps to allow post processing variation of radiation detector parameters such as detector resolution, the angles of registration, etc.

6. AKNOLEDGMENTS

This work was supported by the Russian Leading Scientific Schools Foundation, project NSh- 8129.2010.9, RFBR, grants 10-01-00302 and 11-01-00870, and by INTEGRA Inc. (Japan).

7. REFERENCES

[CATIA] CATIA - Virtual Design for Product Excellence, http://www.3ds.com/products/catia/welcome/

[Hashisuka 2009] Toshiya Hachisuka and Henrik Wann Jensen. Stochastic progressive photon mapping. ACM Transaction on Graphics, 28(5), pp. 1–8, 2009.

[Havran 2000] Vlastimil Havran. Heuristic Ray Shooting Algorithms: Ph.D. thesis / Department of Computer Science and Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague. November 2000 http://www.cgg.cvut.cz/~havran/phdthesis.html.

[Pharr 2004] Matt Pharr, Greg Humphreys, Physically Based Rendering. Elsevier 2004.

[Voloboy, Galaktionov 2006] А.Г. Волобой, В.А. Галактионов Машинная графика в задачах автоматизированного проектирования // "Информационные технологии в проектировании и производстве", № 1, 2006, с. 64-73

[Voloboy et al 2009] А.Г. Волобой, В.А. Галактионов, А.Д. Жданов, Д.Д. Жданов. Средства визуализации распространения световых лучей в задачах проектирования оптических систем // "Информационные технологии и вычислительные системы", № 4, 2009, с. 28-39.

[Zlib 2010] http://zlib.net/

About the authors

Boris Kh. Barladyan, PhD, senior researcher, Keldysh Institute for Applied Mathematics RAS.

E-mail: obb@gin.keldysh.ru

Lev Z. Shapiro, PhD, senior researcher, Keldysh Institute for Applied Mathematics RAS.

Alexey G. Voloboy, PhD, senior researcher, Keldysh Institute for Applied Mathematics RAS.

E-mail: voloboy@gin.keldysh.ru