Creative ray tracing

Bernhard Michel, Eberhard Lange, Simuloptics GmbH, Schwabach, Germany

For virtual prototyping in the industry, non-sequential ray tracers are becoming more and more common. They can do much more, however: they enable detailed analyses of optical systems and thus help to understand their characteristics and behaviour. This is usually the basis for a target-oriented optimisation. Ray tracing with advanced software features, controlled and creatively utilised by the user, can lower the computing time by many orders of magnitude, and thus for the first time makes certain problems accessible to optics simulation.

The standard method for real world simulation of optical systems is non-sequential ray tracing: this means the order in which a ray hits objects is not pre-defined, but calculated during the ray propagation. This way one obtains a physical model that is correct within the framework of geometrical optics: Fresnel reflection and transmission, scattering from surfaces and volumes, extended light sources, but also the interaction with layer systems, gratings and diffractive optical elements can be simulated herewith. Taking into account the optical path length, non-sequential ray tracing can also be applied to coherent optical systems [1]. With accurate input data, non-sequential ray tracing is ideally suited for creating virtual prototypes and is used for this purpose on a large scale in industry.

This shifts the development, and with it most development *problems*, from the laboratory to the computer. If a real prototype does not meet system specifications, the virtual prototype won't do it either. If, for example, a prototype of an automotive headlamp exceeds the limit value for glare, the virtual prototype will merely confirm this, without giving additional hints *why* glare takes place and *how* one could fix this problem.

In fact, non-sequential ray tracers have many more capabilities than just *verifying* the performance of optical systems: based on elaborate analysis methods, they help the user to *understand* an optical system. Analysis means: asking the optical system questions. As a response to clever questions, the ray tracer will return answers that help substantially when fixing problems or optimising the system. A necessary condition for this is, however, that the software used permits asking such questions at all.

In this paper, we provide intentionally simple examples to demonstrate how ray tracing can be applied in a smart way to get a maximum of answers within a minimum of time. The simulations were performed using the optical software ASAP by Breault Research Organization (the respective ASAP commands are stated in parentheses).

1 Model of a flashlight

Consider an idealised flashlight (**figure 1a**), consisting of an incandescent filament and a parabolic mirror, illuminating a distant wall (all other parts of the flashlight are unimportant for this study). Examples such as this are ideally suited for demonstrating the capabilities of optical analysis.

If the centre of the filament is placed in front of the focus of the reflector, we expect a diverging beam such that a large area of the wall is illuminated. The simulation produces the irradiance distribution shown in figure 1b. There is a central spot surrounded by a ring and a diffuse background. Naively, one might expect that defocused optics only produce a blurred spot, so where does the ring come from? In order to find this out, one could try to display selected ray paths graphically. With a little bit of luck one could find some of the rays that cause the ring. We suggest a different approach, which is notably also successful for complex systems.

First of all, we make a survey of all ray paths (PATHS) that hit the wall:

- 5% of the rays hit the wall without having interacted with the reflector at all. This is the direct light, irradiated from the source into forward direction.
- 40% of the light that is irradiated by the filament undergoes one reflection from the reflector
- 55% – and this is a surprisingly high amount – gets reflected from the mirror twice. This is already a hint at the origin of the ring.

Figure 1c is a graphical representation of the ray paths (HISTORY PLOT) mentioned above, indicating that the double reflection path is indeed responsible for the central ring. For a more detailed analysis we select only the rays hitting the ring (SELECT), reverse their directions (REVERSE), and propagate them back into the system using single ray tracing steps. Figure 1d shows the strike points of the rays on the reflector. Starting from the light source, these rays initially undergo a reflection from the rear and a second reflection on the front section of the reflector before striking the wall. This model of a flashlight

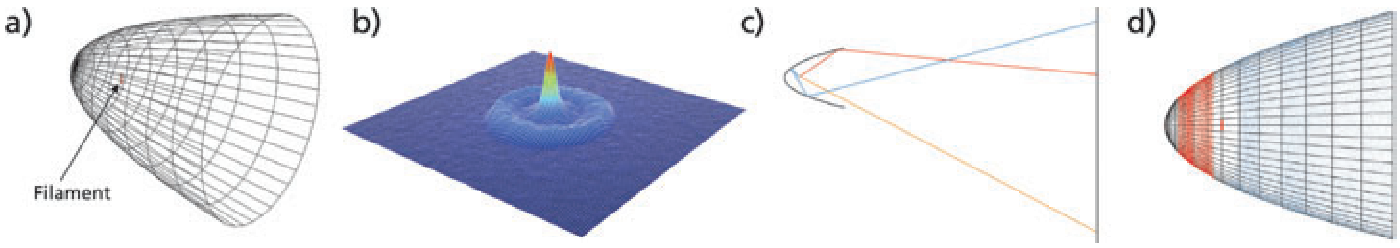


Figure 1: a) Simple model of a flashlight: parabolic reflector and incandescent filament (away from reflector focus); b) irradiance distribution of the flashlight on a distant wall; c) some representative ray paths: direct light (yellow), single-reflected light (red) and double-reflected light (blue); d) Analysis of the double reflected light, which causes the ring in b): light originating from the filament hits the red marked region of the reflector, gets reflected a second time from the blue marked region and is then radiated onto the wall

Originally published in German in Photonik 3/2006

exemplifies a typical problem of many illumination systems: stray light caused by unwanted reflections. In the case of automotive headlamps, this could, for example, lead to glare towards the oncoming traffic. The method just described allows for very efficient problem analysis – the first step towards fixing the problem.

This example also begs the question as to how one could illuminate the wall without generating the ring? One obvious solution would be to cut off the rear part of the reflector, but this would result in lost illumination power. Changing the basic shape of the reflector would make more sense, and the most appropriate shape can be found using suitable optimisation algorithms; more about this in section 4.

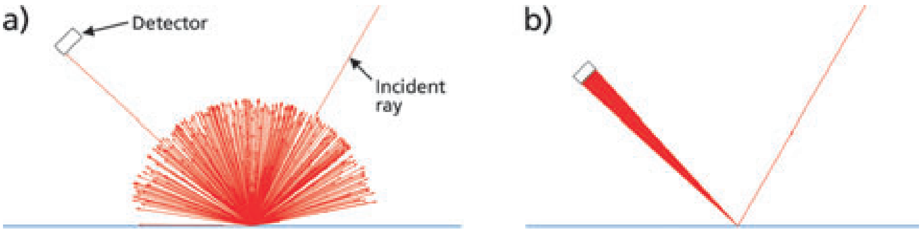


***Figure 2**: Modelling light scattering: a) one incident ray generates many child rays which are radiated into random directions. Only a few rays hit the detector. b) Using importance sampling only those scattered rays are produced that are directed towards a pre-defined target – in this case the detector*

2 Importance sampling

Ray tracers often have to struggle with an unnecessarily large number of rays. As an example, **figure 2a** shows a simplistic set-up for a light scattering measurement. A collimated beam illuminates a spot on a light scattering surface. A detector measures the light scattered into a certain angular region. Ray tracers often model light scattering using "ray-splitting": one incident ray generates many scattered rays with random directions, the flux and distribution of which is chosen according to the angular scattering characteristic of the surface (BRDF – Bidirectional Reflectance Distribution Function). Assuming the detector to subtend a solid angle of $1(^\circ)^2$ then only each 20600th ray on average will hit the detector[1], all others get lost and only waste computation time. So-called "importance sampling" offers a solution to this problem: only those scattered rays are created that are directed towards a certain target region – in this case towards the detector. Figure 2b shows the same simulation as figure 2a, but now employing importance sampling: all scattered rays now hit the detector, because only those rays have been created. With the same statistical error level, the computation runs orders of magnitude faster than before – in our case ca. 1000 times. In practical applications a speed gain of a factor of 10^{12} and more is often encountered.

3 Stray light analysis

Figure 3a shows a typical Cassegrain telescope used for astronomical observation. The tube around the telescope and the cone-shaped baffle protect the detector (e.g., a CCD camera) reasonably well, but not completely, from stray light originating from sources other than the observation direction. In reality, all physical surfaces scatter light – the tube, the mirrors etc. – and scattered light can thus still hit the detector indirectly through scattering. For a quantitative computation of the stray light intensity on the detector using conventional ray tracing, a huge number of rays would be necessary as the likelihood that a scattered ray reaches the detector is very small. The time effort needed for a sufficiently accurate calculation on a standard PC, tracing 10^4 rays per second, lies in the order of several tens of thousands of years and is of course completely impractical. By using a clever combination of several ray tracing steps together with importance sampling, the calculation can be accomplished at the same level of accuracy within a couple of minutes.

For stray light analysis, a systematic procedure, which has proven useful and which will be presented in the following, is well-documented in the literature [2,3]. It consists of a sequence of steps, all of them using "creative ray tracing". We consider a stray light source (for example the moon or street lighting), which is incident on the telescope under 60° (figure 3a).

In the first two steps we neglect scattering. We trace the light from the stray light source to the telescope and list all surfaces that were hit directly or indirectly via mirrors, lenses etc. (TRACE STATS) – these objects are called illuminated objects. The next ray trace is done in the opposite direction: we consider the detector as a light source and investigate which objects are hit by the rays emitted from the detector. These are the objects that are in the field of view of the detector – the so-called critical objects (figure 3b).

As an example, we demonstrate this procedure for one select stray light path: "scattering from the tube and subsequent scattering from the primary mirror" (**figure 4a**). At first, we assign scattering properties to the tube and mirror – preferentially based on experimental data. In the next step we specify the target regions for the importance sampling. The light scattered by the tube should hit the primary mirror. The latter will therefore be used as a target region for importance sampling. The light from the primary mirror should then be scattered to the detector via the secondary mirror. A suitable aiming region for importance sampling is therefore the *virtual image* of the detector in the secondary mirror.

We can determine this virtual image easily – this being another application of "creative ray tracing" (figure 4b): we create a light source at one corner of the detector and trace the rays from there until it gets reflected from the secondary mirror. The rays now point into the directions indicated by arrows. We determine the best focus of these rays, i.e., that point in space with minimum distance from the rays. This point constitutes one "corner" of the virtual

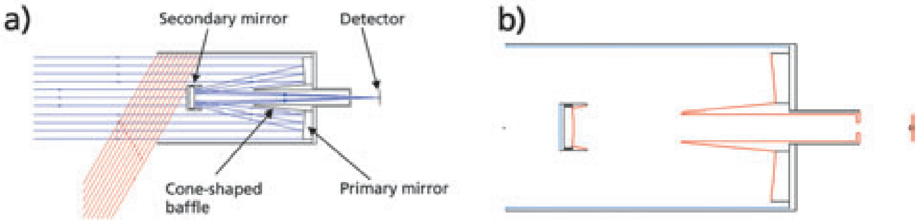


***Figure 3**: Cassegrain-telescope: a) ray paths for "wanted light" is shown in blue, those of stray light (unwanted light) in red. Scattering from the tube is not yet taken into account. b) The objects, which are illuminated directly by the stray light source (without light scattering), are marked blue, those objects that lie within the field of view of the detector (critical objects) are marked red*

---

[1] Comment: A hemisphere subtends a solid angle of $360^2/2\pi$, which is ca. 20600 square degrees. In SI-units 1 square degree corresponds to $(2\pi/360)^2$ sr or ca. 0.00030422 steradian.

Originally published in German in Photonik 3/2006

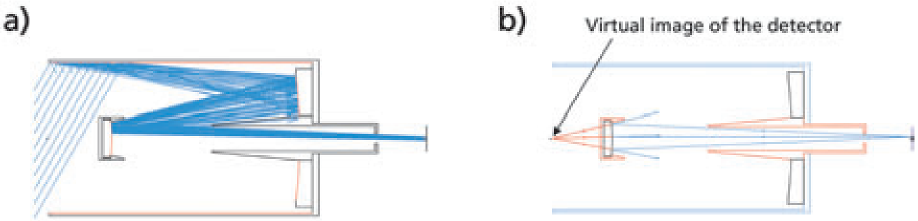


***Figure 4**: Cassegrain-telescope: a) stray light path which connects the stray light source with the detector via scattering from the tube and the primary mirror; b) how to construct the virtual image of the detector in the secondary mirror*

image of the detector. The other corners are constructed similarly. Now, after having prepared everything for an efficient simulation, we can start ray-tracing. Because of importance sampling, almost all rays created at the stray light source will actually reach the detector (figure 4a). Computing the exact contribution of the stray light path to the detector signal now takes only a few minutes. The remaining stray light paths (which result from the remaining combinations of illuminated and critical objects), can be analysed in a similar fashion.

4 Thick Rays

Ray tracers typically model the light distribution of expanded sources with random-number generators. Simulation results are thus afflicted with statistical noise. If N rays hit a detector, the relative statistical error amounts to approx. 1/sqrt(N). In order to compute a picture with a resolution of 100 x 100 pixels with an accuracy of one percent, 100^4 = 100 million rays are required. Particularly for optimisation tasks, those requring re-running the simulation over and over, a simulation with too high a ray count is out of question because of prohibitively large computing times.

A less well-known and little used tool offers a remedy: the simulation with "thick" rays in ASAP. These are Gaussian beams, more commonly utilised in coherent optics [1]. They can be used likewise for incoherent, geometrical-optical simulations, however. Except for some tiny remaining ripples, decomposition into Gaussian beams typically yields smooth intensity distributions independent of the number of beams used. With declining beam number, the picture merely becomes somewhat blurred.

Figure 5a illustrates the concept on the basis of a very simple example. An isotropically radiating light source sits in a simple reflector. With one million regular ("thin") rays we obtain the light distribution in figure 5b, with 5400 thick rays the distribution in figure 5c. Despite a reduction in computation time by a factor of 25, the distribution in figure 5c is much smoother. Only a few details, such as the sharp peaks at the edge, are somewhat flattend due to the lower resolution. If these details are of practical relevance, then one could gradually increase the number of thick rays in the final phase of an optimisation, until finer details are sufficiently captured.

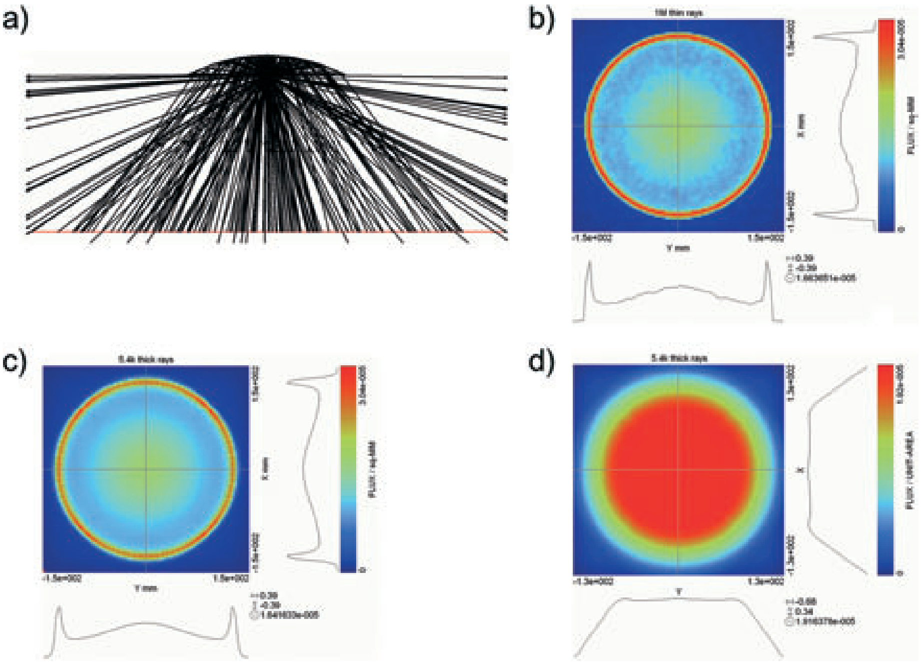


Figure 5: *"Thick" and "thin" rays compared side by side. a) Isotropically radiating light source with simple reflector; b) light distribution on the detector surface, computed with one million thin rays; c) the same with only 5400 thick rays; d) reflector shape that optimises the uniformity of the light distribution: result after 100 optimisation steps in less than 2 minutes*

Figure 5d shows the result of such an optimisation. We vary three parameters of the reflector model and optimise the uniformity of the light distribution. Owing to ray tracing times of less than one second, the 100 optimisation steps took up less than two minutes. The true value of such techniques becomes evident when optimising more complex systems with a higher number of free parameters.

5 Conclusion

Ray tracing – used in a flexible and creative way – helps to study and understand the performance of optical systems qualitatively and quantitatively. Intelligent software features enable the handling of tasks that are either too time-consuming or are not accessible at all with conventional ray tracing.

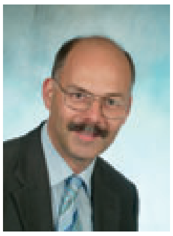
Acknowledgements

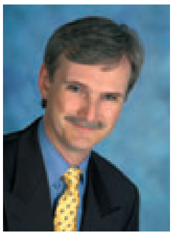
We thank the Breault Research Organization for the model of the Cassegrain telescope prepared by Gary L. Peterson.

Literature:

[1] S. Miller, J. Pond, B. Michel, *Raytracing meets Maxwell's Equations: integrating micro- and macro-optical design*, Photonik International 2006, p. 76-79
[2] G.L. Peterson, *Stray Light Calculation Methods with Optical Ray Trace Software*, Proc SPIE Vol. 3780, p. 132-137, Sep. 1999
[3] R.P. Breault, *Control of Stray Light*, in: *Handbook of Optics*, Vol. 1 (McGraw-Hill, New York, 1995), Chapter 38

Author contact:





Dr. Bernhard Michel
Eberhard Lange

Simuloptics GmbH
O'Brien Str. 2
91126 Schwabach, Germany
Tel. +49/9122/830300
Fax +49/9122/830303
eMail: info@simuloptics.de
Internet: www.simuloptics.de

Originally published in German in Photonik 3/2006