

Stray light calculation methods with optical ray trace software

Gary L. Peterson

Breault Research Organization
6400 East Grant Road, Suite 350, Tucson, Arizona 85715

Copyright 1999, Society of Photo-Optical Instrumentation Engineers (SPIE). This paper will be published in the proceedings from the July 1999 SPIE Annual Conference held in Denver, Colorado, and is made available as a preprint with permission of SPIE. Single print or electronic copies for personal use only are allowed. Systematic or multiple reproduction, or distribution to multiple locations through an electronic listserver or other electronic means, or duplication of any material in this paper for a fee or for commercial purposes is prohibited. By choosing to view or print this document, you agree to all the provisions of the copyright law protecting it.

ABSTRACT

Faster, better, and cheaper computers make it seem as if any optical calculation can be performed. However, in most cases brute force stray light calculations are still impossible. This paper discusses why this is so, and why it is unlikely to change in the near future. Standard software techniques for solving this problem are then presented, along with a discussion of how old techniques are used to take advantage of the new features that are available in the latest generation of optical analysis software.

Keywords: Stray light, scatter, optical analysis software, ray trace software

1. INTRODUCTION

Stray light calculations are done to find out how unwanted light propagates to a detector or focal plane and to determine if its irradiance is large enough to cause a problem. Faster, larger, and cheaper computers, as well as the advent of commercial optical analysis software have made these calculations faster and easier to perform. However, brute-force stray light calculations are still impossible. Fortunately, optical analysis software and standard stray light analysis techniques can be combined to simplify and automate identification of stray light paths. Importance sampling is then used to perform quantitative stray light calculations in a reasonable time.

2. BRUTE-FORCE STRAY LIGHT CALCULATIONS

One way of performing a stray light calculation is to do a direct computer simulation of the way light propagates through an optical system. This direct, or *brute-force* simulation consists of 1. constructing a computer model of the baffles, vanes, housings, struts, lenses, and mirrors of an optical system, and 2. tracing lots of rays through the system, allowing them to scatter, reflect and generally rattle around inside until some of them make it to the detector.

This approach has several attractive features:

1. It requires minimal labor on the part of the stray light analyst. Labor is expensive. Computers are cheap, large, and fast. So this seems to be a good allocation of resource.

2. It is a good simulation of the real world. Light really does scatter and reflect in all possible directions until a small fraction of it actually makes it to the detector.
3. It requires little knowledge of optics. The computer performs all of the imaging, propagation, and scatter calculations, so personnel without formal training in optics can be employed in the task.
4. It is unbiased. No preconceived ideas about what are the most important paths, or how light should propagate within the system are imposed on the calculation. The computer simply performs a random real-world simulation of how light actually propagates.

There is only one problem with this method: it doesn't work! To see why, consider a stray light analysis of a simple Cassegrain telescope that is illustrated in Figure (1). As shown in the figure, it is usually necessary to model stray light paths that have as many as two bounces, or scatters, from the source to the detector. How much time is needed to perform such a calculation? To find out, assume an initial sampling of the entrance aperture with a 1000×1000 grid of rays. Furthermore, assume each time a ray hits a surface that enough scattered rays are generated to sample the hemisphere above that surface with a density of one ray for each square degree. How many rays are needed and how long will it take to trace them? There are one million initial rays, and there a 20,600 square degrees in a hemisphere. So when each of the one million initial rays strikes a surface it generates 20,600 rays. Furthermore, since we want include as many as two scatters, each scattered ray generates 20,600 additional rays when it strikes a surface. Therefore, an estimate for the total number of rays required to perform this calculation is

$$1 \text{ million} \times (20,600)^2 = 4 \times 10^{14} \text{ rays,}$$

or 400 million million. At the present time, a typical ray trace speed through a complex system on a Pentium™ class computer is 1 million rays per hour. Dividing the number of rays by this rate gives a total ray trace time of 4×10^8 hours, or 46,000 years! And that is for only one source located at a single position or direction. Clearly, the brute-force approach is not a practical method.

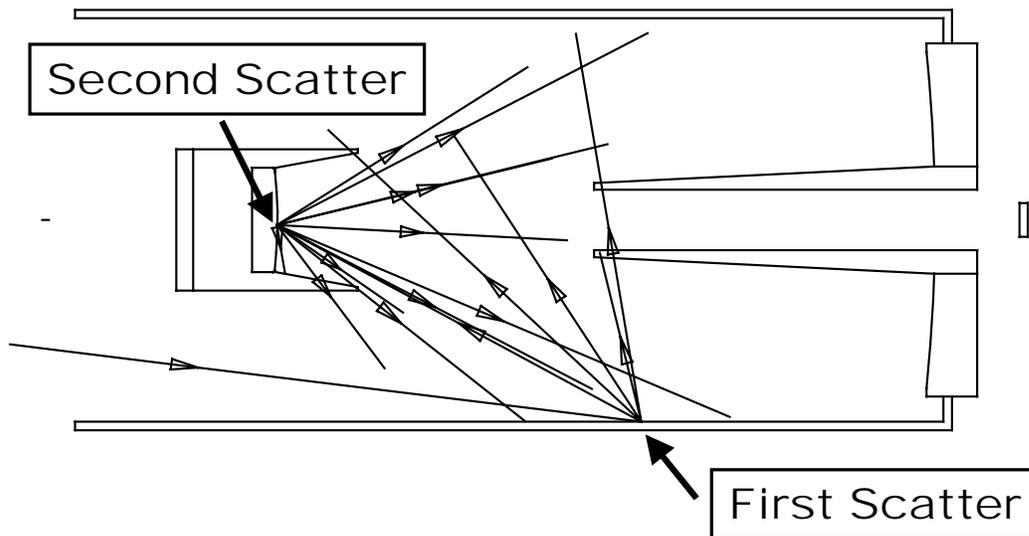


Figure 1: Example of two scatters inside a Cassegrain telescope.

3. SYSTEMATIC STRAY LIGHT CALCULATIONS

In practice, the time required for a stray light calculation is greatly reduced by abandoning the brute-force approach and applying two tools: 1. systematic identification of stray light paths, and 2.

importance sampling (or directed scatter). Both of these tools have been used in the stray light community for over twenty years, but modern computers and software allow even novice stray light analysts to apply them reliably.¹

Systematic identification of stray light paths starts at the detector to identify the *critical objects*. A *critical object* is any object or surface that is seen from the detector, including objects that are seen as images in mirrors and lenses. Critical objects are important in stray light analysis because 100% of the stray light comes from these objects. If the detector cannot see an object, then no stray light comes from it.

Critical objects are identified by tracing rays backward from the detector. This is illustrated in Figure (2). Any object or surface that is intersected by a ray is a critical object. The figure shows an in-plane ray trace (for clarity), but in practice a full three-dimensional ray trace is done to obtain a complete list of critical objects. Furthermore, an actual ray trace should include a series of point sources across the detector, or rays should be randomly distributed within the detector area and within the hemisphere above the detector. After the ray trace, a list of all of the objects intersected by rays is made for later reference.

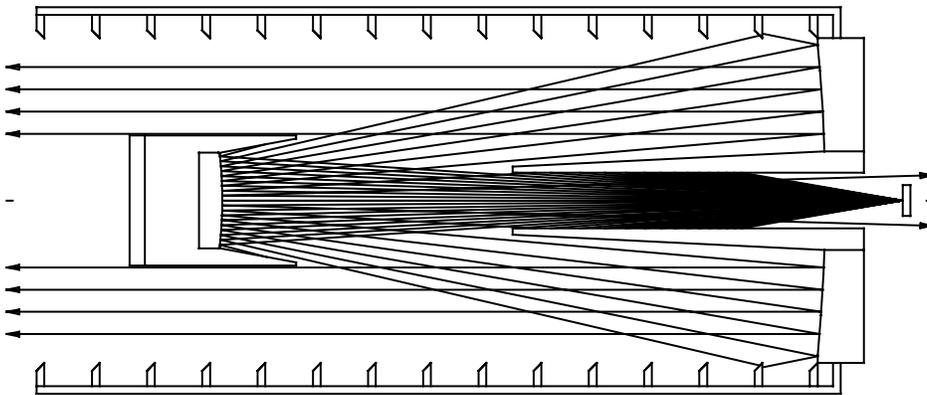


Figure 2: Illustration of a backward ray trace that is used to identify critical objects.

Note that using a computer to find critical objects preserves some of the advantages of the brute-force approach. It is unbiased. Given an accurate and complete model of the system and a dense collection of rays on the detector, the computer will find all objects that are visible from the detector without regard to the analyst's preconceptions about what should or should not be seen. The calculation can also be performed without an extensive knowledge of optics, as the computer handles all of the imaging during the ray trace.

After identifying the critical objects, the next step is to identify illuminated objects. For a given source and source location, an illuminated object is any object or surface that receives power from the source, either directly, or after reflection and refraction by the mirrors and lenses. Objects that are illuminated indirectly by scattered light are not included in this list.

Illuminated objects are identified by doing a forward ray trace through the system. Rays are traced forward from the source through the entrance of the optical system. Any object or surface that is struck by a ray is an illuminated object. This is again a fully three-dimensional ray trace so that all objects are identified. After the ray trace, a complete list of illuminated objects is made. In general, the list of illuminated objects changes with source position, so there are usually several of these lists: one for each source and source location. As with the backward ray trace that was used to identify critical objects, using a computer to identify illuminated objects is unbiased and does not require an extensive knowledge of optics.

Systematic identification of stray light paths is done by comparing the lists of critical and illuminated objects. Any object or surface that is on both the critical object list and the illuminated object list defines a first order stray light path. That is, the source and detector are connected by scatter from a single surface. At this time there is enough information to make design changes that will reduce the stray light background. In most cases, if first order stray light paths exist they are the dominant source of stray light. Stray light is reduced by either moving the object or surface so it is no longer seen from the detector or directly illuminated, or by inserting a baffle that blocks the detector's view of the object or blocks illumination of the object.

Second order, or two-scatter, stray light paths are found by connecting objects on the list of illuminated objects with the list of critical objects. That is, if an object on the critical-object list is seen from an illuminated object, then a second order path exists in which light from the source is scattered from the illuminated object to the critical object, and then scattered from the critical object to the detector. In most cases, it is obvious whether or not an optical path exists from the illuminated object to the critical object. But if there is doubt, ray tracing can be performed to find out. An extended source is placed over the illuminated object (or critical object) and a three dimensional ray trace is performed. If rays from the illuminated object strike the critical object (or vice versa), then a second order path exists. In general, the list of second order stray light paths consists of all possible connections between objects on the list of critical objects and objects on the list of illuminated objects. While the number of second order stray light paths is often large, the above procedure for identifying them is easy to apply and the chances of overlooking an important stray light path are small.

Note that no scatter calculations have been performed yet. The backward and forward ray traces consist entirely of refractions and reflections from the lenses and mirrors, and straight line transfers from one object to another. Because of this, a dense sampling of the system in the critical and illuminated object ray traces is practical: again, typically a million rays can be traced on Pentium™ class computers in about an hour.

Now that a complete list of first and second order stray light paths exists, it is time to perform quantitative stray light calculations. These calculations are made practical by using *importance sampling* or directed scatter. The vast majority of light that strikes an illuminated object never makes it to a critical object or the detector. Because of this, simulating scatter by tracing rays in all possible directions is a tremendous waste of computer time on rays that contribute absolutely nothing to the stray light calculation. The idea behind importance sampling is to avoid this waste by tracing only rays that propagate in interesting, or *important* directions. That is, towards critical objects or the detector.

This idea is illustrated for first and second order scatter in Figures (3) and (4). In Figure (3), rays from a source that strike the front surface of three vanes next to the primary mirror are scattered towards the image of the detector in the secondary mirror, which causes them to propagate directly to the detector. In Figure (4), rays that strike the housing around the secondary mirror are scattered towards the primary mirror, and are then scattered again towards the image of the detector in the secondary mirror. Vast savings in computer time are obtained by tracing only a small number of rays towards the areas of interest.

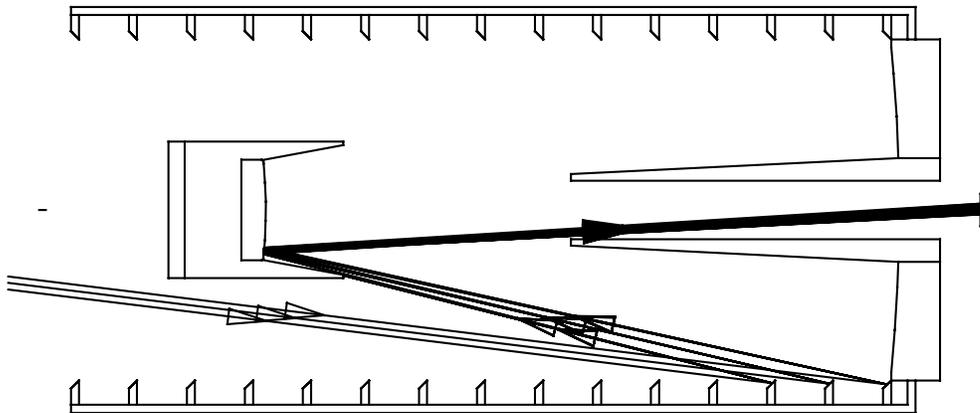


Figure 3: Illustration of a ray trace with importance sampling to calculate the stray light from a first order scatter path.

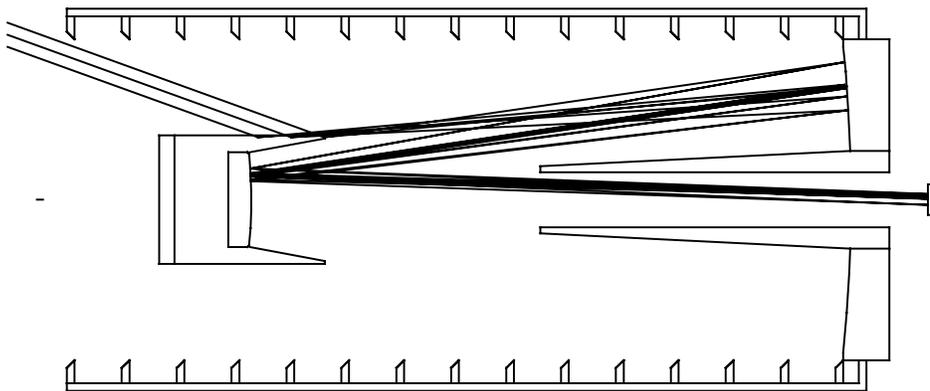


Figure 4: Ray trace in which rays are scattered towards the primary mirror, and the scattered from the primary mirror to the detector.

Naturally, in order to get the right answer it is not sufficient to simply send all of the scattered rays in interesting directions. The quantitative aspects of this approach must be accounted for. In particular, each ray must be assigned a power, and the power of scattered rays must be scaled in proportion to the power of the incident rays, the solid angle subtended by the important area, the scatter properties (Bidirectional Scatter Distribution Function, or BSDF) of the surface, and the cosine obliquity factor of the scattering surface. Fortunately, this quantitative aspect of the ray trace is generally handled entirely by the optical ray trace software.

Importance sampling works well when combined with the stray light paths that were identified in the backward and forward ray traces. For each first order stray light path, an important area is defined over the image of the detector as seen from each object that is both an illuminated and a critical object. For each second order stray light path, an important area is placed over each critical object (or image of the critical object) in addition to an important area over the detector image.

Once the important areas are defined quantitative stray light calculations begin. Rays are traced from the source into the system. Then rays are scattered from each illuminated object towards the image of the detector (first order stray light paths) or towards the critical objects (second order stray light paths), and from critical objects to the detector image. The end result is an accurate stray light calculation that avoids tracing rays propagating in uninteresting directions, and thereby reduces the computer time required by many orders of magnitude.

4. CONCLUSION

In most cases, brute-force stray light calculations are still well beyond the capabilities of computers that are currently available. However, using computers and optical ray trace software to implement standard stray light techniques allows systematic identification of stray light paths to be performed without an extensive knowledge of optics. In addition, importance sampling vastly reduces the computer time required to perform quantitative stray light calculations so that an accurate stray light irradiance can be obtained.

ACKNOWLEDGMENT

The author gratefully acknowledges Robert P. Breault for introducing him to the use of critical and illuminated objects to identify stray light paths.

REFERENCE

1. Robert P. Breault, "Control of Stray Light", in *Handbook of Optics*, Volume 1 (McGraw-Hill, New York, 1995), Chapter 38.