Good afternoon everyone! This project began early last summer during an internship at Walt Disney Animation,
Where we initially met with a team of VFX artists see what features they wanted from the next generation of Disney’s Hyperion rendering engine. One of the features which they requested
was for Hyperion to eventually support rendering general procedural media.
This would allow for supporting near infinite amounts of detail within the volumes used in their theatrical productions. However, due to algorithmic constraints this is not currently supported.
To render their productions, Hyperion currently relies on techniques based on the null-scattering paradigm,
Homogenization

Which requires that we first homogenize all volumes.
Take for example this heterogeneous volume,
Which has a spatially varying density function. For visualization purposes, I am going to replace the rendering of the volume,
With a visualization of the average density looking straight through the medium. Null-scattering techniques require us
to inject fictitious, or null density into the medium, such that if we took
the null density,
and real density
And then combined them by adding them together,
We would end up
With a constant density medium whose
total combined density is often referred to
As the majorant. Now, most modern algorithms allow us to specify the majorant (click) as any non-zero positive value, and no matter what majorant we specify,
Real Density

Majorant

Real Density

Density

X
Modern null-scattering algorithms should still give us the correct expected result. However, the choice of the majorant directly impacts the performance of our renders.
If the majorant is set too high (*click*),
then the cost of our renders will increase, and may even become too costly to in production.
Alternatively, if the majorant is set too low (*click*) such that the majorant no longer bounds the density, the renders might become fast,
But the variance of the renders will become so uncontrollable, that the renders will never converge in any reasonable amount of time. To have both low cost
and guarantee low variance, we ideally want a majorant which (*click*) bounds the density as tightly as possible. However, getting these tight majorants
Volume representation

Depends on how we represent our volumes.
if we only render volumes like the Disney cloud,
Which are explicitly stored as voxel density grids, then finding a tightly bounding majorant.
Is as easy (click) as iterating through all of the voxels,
Density

Real Density

X
And setting the majorant to the largest found density. However,
when dealing with purely procedural volumes,
Or complicated production workflows,
The underlying density (*click*) becomes a black box whose general shape is not known to the renderer.
Any majorant which we specify, becomes an approximation for a bounding majorant,
Density

Majorant
Real Density
Becoming very difficult to guarantee that we will ever have a tightly bounding one. And, without a tightly bounding majorant we can’t be sure that we will avoid,
uncontrollable variance,
Or too inefficient of renders
It is for this reason that Disney currently
bakes all volumes into voxel density grids before rendering them. Representing all volumes as grids allows null-scattering techniques to be (*click*) robust, since, having a density grid (*click*) guarantees that tight majorants can always be found. However,
Density

👍 tight

👍 robust

Real Density

X

Density
The process of baking reduces (*click*) the visual fidelity of all volumes, necessitates (*click*) preprocessing all volumes, and significantly increases (*click*) the storage requirements for all production scenes. We instead propose a solution which,
👍 tight 👍 robust

😢 visual quality
👍 tight 👍 robust

😢 visual quality
😢 preprocessing time
👍 tight
👍 robust

😢 visual quality
😢 preprocessing time
😢 storage / memory
Makes most null-scattering techniques resilient to non-bounding majorants. Our technique is (*click*) robust, discovers (*click*) tight majorants during render time,
👍 tight 🐝 👍 robust

😢 works for any majorant

😢 visual quality

😢 preprocessing time

😢 storage / memory
👍 tight 🐝
👍 robust
😊 works for any majorant
😢 visual quality
😢 preprocessing time
😢 storage / memory
👍 tight
👍 robust
😊 works for any majorant
😢 visual quality
😢 preprocessing time
😢 storage / memory
maintains the same visual quality as using bounding majorants in the converged renders,
👍 tight 🐝 👍 robust

😊 works for any majorant

😊 visual quality

🤔 preprocessing time

😢 storage / memory
require little to no preprocessing time
👍 tight 🍒       👍 robust
😊 works for any majorant
😊 visual quality
😊 preprocessing time
❓ storage / memory
And requires no extra storage on top of what is already needed in any practical implementation of null-scattering. However, in return for fixing all these prior issues,
👍 tight 🐜 💖 works for any majorant
👩‍💻 visual quality
💪 preprocessing time
😄 storage / memory
Our technique requires that we relax the (*click*) unbiased property of most existing methods,
👍 tight 🦄  👍 robust

😊 works for any majorant

😊 visual quality

😊 preprocessing time

😊 storage / memory

😊 unbiased
To instead settle for only being consistent.
👍 tight ☽ ☽ robust

😊 works for any majorant

😊 visual quality

😊 preprocessing time

😊 storage / memory

😐 consistent
Now, let us define what we mean by consistency.
Let's represent the first pixel sample in a render as $I$ of 1, and let's also assume that
This first pixel sample is very biased.
A full render effectively takes the average across many different pixel samples. A consistent algorithm

\[
\frac{1}{n} \sum_{k=1}^{n} I_k
\]
Is one which will guarantees that our render will eventually converge to the real solution,
in the limit. Regardless of how many individual pixel samples are biased.
Biased but consistent algorithms have appeared throughout graphics from photon mapping to many-light methods. And one idea, which has been used previously by virtual point lights,
Unbiased and consistent rendering using biased estimators

ZACKARY WISK, Dartmouth College, USA
KENDALL BETTENCOURT, Dartmouth College, USA and NVIDIA, USA
ILIANA GEORGIEVA, Aardvark, United Kingdom

Prior inspiration

Full support for procedural media

Progressive Photon Mapping

Takashi Tachibana
UC San Diego

Shadi Qaid
The University of Nottingham

Nick Warrener
UC San Diego

Scalable Realistic Rendering with Many-Light Methods

Carmen Debevec
Lawrence Berkeley National Laboratory

Martin Newell
MPI Computer Graphics, Germany

Emanuele Pardini
Uni North Carolina at Chapel Hill

Jan Novák
University of North Carolina at Chapel Hill

Abstract

This paper presents a method for scalable, real-time, realistic rendering of scenes with many light sources. The method is based on a fast approximation of the light source function, which is computed in a pre-processing step. The approximation is then used to construct a global illumination solution that is accurate and robust, even in the presence of a large number of light sources. The method is demonstrated on a variety of complex scenes, including a virtual city and a virtual room, where it produces high-quality results at interactive rates.

1 Introduction

The problem of realistic rendering of a scene with many light sources is a challenging problem in computer graphics. Most existing methods for light transport and global illumination are limited to scenes with a small number of light sources, due to the high computational cost of rendering.

Figure 1: A single scene illuminated by a large number of light sources. The method presented in this paper is able to render such scenes in real-time, with high-quality results.

The main advantage of this method is its scalability, which allows it to be used in a wide range of applications, including virtual environments, real-time rendering, and interactive visualization.

4 Conclusion

In conclusion, this paper presents a new approach for scalable, real-time, realistic rendering of scenes with many light sources. The method is able to produce high-quality results at interactive rates, making it suitable for a wide range of applications.

Acknowledgments

This work was supported by the National Science Foundation under grant number 0820151. We would like to thank John Porter, who provided valuable feedback on an earlier version of this paper.

References


Clamping to reduce variance

Is the idea of clamping to reduce variance. This is the first step in our technique.
Significantly non-bounding majorants lead to uncontrollable variance, HOWEVER, we can enforce our majorants to be always bounding,
by clamping the medium density to the specified majorant. This process will obviously
Make the medium itself biased. However, one thing to make note of is that if we
Increased the majorant, and thus clamped less of the density,
Our render would be less biased. And you can now start to see a thought experiment forming.
Density

X

Majorant
Real Density

Density

X
What if we had a sequence of pixel samples,
which all used monotonically increasing majorants. Meaning, the first few pixel samples will be biased,
But after some finite point a bounding majorant will be found and every subsequent pixel sample will be unbiased.
Majorant increases from left to right
The entire render would be the average of all these images.
\[ \sum_{k=1}^{j} \frac{I_k}{n} + \sum_{k=j+1}^{n} \frac{I_k}{n} \]

Which we can decompose into a finite number of (*click*) biased terms, while the remainder are all (*click*) unbiased.
\[ \sum_{k=1}^{j} \frac{I_k}{n} + \sum_{k=j+1}^{n} \frac{I_k}{n} \]

Biased
\[
\sum_{k=1}^{j} \frac{I_k}{n} + \sum_{k=j+1}^{n} \frac{I_k}{n}
\]

Biased  Unbiased
In the infinite limit, the biased contribution is going to converge to

\[
\lim_{n \to \infty} \left[ \sum_{k=1}^{j} \frac{I_k}{n} + \sum_{k=j+1}^{n} \frac{I_k}{n} \right]
\]

Biased Unbiased
Zero, while the contribution from the infinite remaining unbiased terms is going to converge to

$$\lim_{n \to \infty} \left[ 0 + \sum_{k=j+1}^{n} \frac{I_k}{n} \right]$$

Biased | Unbiased
The true solution. So, if we discover a bounding majorant in finite time, we can make most null-scattering algorithms consistent while avoiding uncontrollable variance.
Which brings us to the second step in our technique. Progressively updating the majorants.
Over the course of rendering, we will naturally evaluate the density
At many different points within the medium.
Density

Real Density

Majorant

X
Every single one of these density evaluations may or may not
get clamped. However, all of these evaluations give us direct,
Estimates for how non-bounding our majorant actually is. We can then choose
the largest difference between any of the density evaluations and our current majorant,
To directly set the majorant to use for the next render pass. We also add a small non-zero
Next Majorant = max(Density Evals)
Next Majorant = max\{Density Evals\} + C

constant to the updated majorant to guarantee that we will discover a bounding majorant in finite time. For brevity, we refer you to the paper for our explanation regarding this. The combination of clamping then progressively updating majorants fully summarizes our progressive null-tracking technique.
Next Majorant = \max\{\text{Density Evals}\} + C
Before we move onto results. We need to mention that while it is our intention to eventually incorporate this technique
Into Hyperion, most of our implementations and results are from PBRT.
Additionally, while we introduced the idea of a majorant as if it were a singular global constant. In practice, we store it
As a piecewise constant function to better locally fit the medium. Thus, we progressively update each majorant individually.
Now, on to some results.
For all results rendered using our technique,
We initialize our majorants to be near-zero to convey the robustness of our technique in the worst case scenario. Our progressive method then updates
Those majorants over the course of a render.
For this bunny scene, we compare
ratio plus weighted delta tracking which are given tightly bounding majorants ahead of time
To our technique, which also uses ratio and weighted delta tracking, except with our progressive clamping and updating. For this scene, our discovered majorants converge to become bounding very quickly so the bias seems visually imperceptible.
In terms of error, our technique converges fairly similarly to ratio tracking.
For the Disney cloud scene, which is a lot more dense, we performed
Equal extinction call comparisons between our method and a few state of the art transmittance estimators. In scenes like this where most of the variance comes from sources outside of transmittance estimation, low cost but higher variance estimators like ratio tracking are still preferable which is why we apply our progressive technique to ratio tracking. The point of this comparison is to show that even in these difficult scenes our progressive technique makes current methods resilient to non-bounding extinctions without taking a significant performance hit.
In conclusion, we have introduced a progressive method for making most null-scattering techniques resilient to non-bounding majorants. Our method imposes no significant performance loss, requires no major modification to any existing null-scattering algorithm, and can be implemented as a simple abstraction layer on top of a renderers medium interface.
In the paper

Progressive null-tracking for volumetric rendering
In the paper

- Full analysis of explosive variance
In the paper

- Full analysis of explosive variance
- Adaptive ratio tracking
In the paper

- Full analysis of explosive variance
- Adaptive ratio tracking
- Proofs and convergence rates
Future work
Future work

• Residual
Future work

• Residual
• Better majorant updating
Future work

• Residual
• Better majorant updating
• Full incorporation into Hyperion
Thank you!