Implementing null-scattering path integral formulation in Mitsuba 2

Semester Project

JAN HOREŠOVSKÝ, EPFL, Switzerland

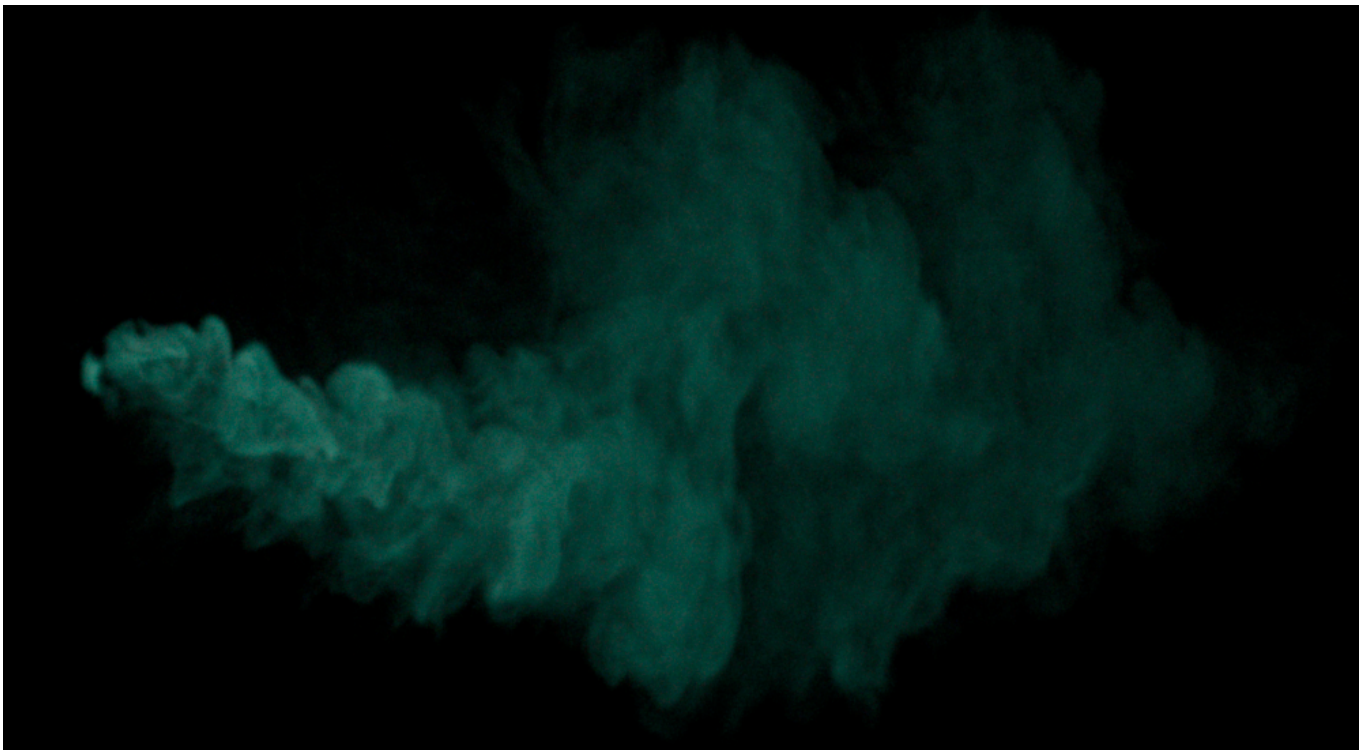


Fig. 1. An unbiased rendering of a spectrally varying heterogeneous medium using an integrator which combines hero wavelength sampling [Wilkie et al. 2014] with unidirectional and next-event estimation sampling strategies. It was made possible in Mitsuba 2 thanks to the new null-scattering path integral formulation of light transport by [Miller et al. 2019].

This report summarizes the implementation of null-scattering path integral formulation of light transport [Miller et al. 2019] in Mitsuba 2. This new path integral formulation introduces new possibilities in combining various sampling techniques via multiple importance sampling in heterogeneous media. This is achieved by providing full path pdfs. It also simplifies rendering software architecture by requiring a smaller number of more robust algorithms. We describe our implementation of this method and reproduce results presented in the original paper. In addition to that, we also discuss caveats related to the practical implementation of the method. One of them is increased noise due to path termination upon sampling an absorption event and the other is increased complexity of path throughput and pdf tracking and MIS weight calculation. These factors should be considered when deciding whether to use the new path integral in Mitsuba 2 by default.

Additional Key Words and Phrases: global illumination, light transport, participating media, null-scattering, Mitsuba 2

1 INTRODUCTION

When rendering images of virtual environments, special attention needs to be paid to any participating media such as clouds, fog or smoke. The reason for this is that such media violate our assumption of radiance constancy along straight lines between surfaces which is used in the basic *rendering equation* which stands at the heart of rendering. Volume rendering (i.e. rendering of participating media) has been thoroughly studied and an overview of recent advances and unsolved issues has been given in [Novák et al. 2018].

One of the issues mentioned there is related to rendering heterogeneous media which cannot be analytically sampled and as a result, the pdf of sampling a path through such media is unknown. This makes it impossible to combine various path sampling techniques via *multiple importance sampling* (MIS). For example, using *hero wavelength sampling* [Wilkie et al. 2014] in such cases is challenging because of its reliance on MIS. Because of this, many rendering systems have to switch between several rendering algorithms based on the type of media present in the scene.

A solution to this issue has been proposed in [Miller et al. 2019]. There, a new path integral formulation of light transport has been derived. This formulation allows for analytic sampling of heterogeneous media which brings the possibility of using MIS in new ways and also of simplifying rendering software so that a smaller number

of more robust algorithms is sufficient to render a large variety of scenes.

Mitsuba 2 [Nimier-David et al. 2019] is an upcoming research-oriented renderer which (among many other things which are not of interest here) is fully spectral. At the moment, however, it lacks support for rendering spectrally varying heterogeneous media and it employs separate algorithms for rendering homogeneous and heterogeneous media.

The goal of this project is to investigate how the new path integral formulation by [Miller et al. 2019] could be implemented in Mitsuba 2, then by implementing it to enable the rendering of spectrally varying heterogeneous media and finally to report on any shortcomings of the practical implementation. The ultimate goal is to assess whether the new path integral formulation should be used by default in Mitsuba 2 in the future.

2 BACKGROUND

In this section, we revise three important building blocks of this project for the sake of completeness and also in order to introduce terminology and notation which will be used in the rest of the report. These building blocks are:

- Volume rendering [Novák et al. 2018]
- Volume rendering in Mitsuba 2 [Nimier-David et al. 2019]
- Null-scattering path integral formulation [Miller et al. 2019]

2.1 Volume Rendering

We would like to evaluate how a certain participating medium influences the propagation of light. One option would be to simulate its every individual atom and how rays of light interact with these atoms. But because that would be intractable, it is common to approach this problem statistically, i.e. to evaluate what proportion of light will likely travel beyond a certain point in the medium.

For this purpose we introduce the *absorption* coefficient μ_a and *scattering* coefficient μ_s which determine the proportion of light which is absorbed and scattered in a medium, respectively. On a line through a medium, absorption and scattering describe the losses in radiance along that line. This leads to the *extinction* coefficient $\mu_t = \mu_a + \mu_s$. All these coefficients are constant in a homogeneous medium and spatially varying in a heterogeneous medium. Radiance can also increase due to *in-scattering* and *emission*, but we will only consider in-scattering because emission is more complicated to handle and is ignored in Mitsuba 2 as well.

The generalization of the *rendering equation* for participating media is the *volume rendering equation* (VRE):

$$L(x,\omega) = \int_0^d T(x,y)\Big[\mu_a(y)L_e(y,\omega) + \mu_s(y)L_s(y,\omega)\Big]\mathrm{d}y + T(x,z)L_o(z,\omega) \tag{1}$$

where

- d is the length of the intersection of a ray given by (x,ω) and the medium,
- z is a point on the surface intersected by that ray,
- $L_s(y,\omega) = \int_{S^2} f_p(\omega,\bar{\omega})L(y,\bar{\omega})\mathrm{d}\bar{\omega}$ is the in-scattered radiance function,
- $f_p(\omega,\bar{\omega})$ is the *phase function* which describes how light scatters in media (it is a medium equivalent of a BSDF),
- $T(x,y) = e^{-\int_0^y \mu_t(s)\mathrm{d}s}$ is the *transmittance* — the fraction of light that makes it from y to x and
- $\mu_a(y)L_e(y,\omega)$ is the emission term which we will ignore

The VRE (1) a recursive equation which makes it difficult to work with in a path tracing setting. A much more flexible way of describing light transport is expressing the VRE (1) using a path integral formulation as was introduced by [Pauly et al. 2000]:

$$L(p_1 \to p_0) = \sum_{n=1}^{\infty} \int_{P_{n-1}} L_e(p_n \to p_{n-1})T(\bar{p}_n)\mathrm{d}\gamma_{n-1}(p_2,\ldots,p_n) \tag{2}$$

where

- $\bar{p}_n = p_0, p_1, \ldots, p_n$ is a path from a point on the camera (p_0) to a point on the light source (p_n),
- P_{n-1} is the set of all paths of length $n-1$,
- $T(\bar{p}_n)$ is the throughput function which is the product of all the geometry terms, transmittance, phase functions and BSDFs that attenuate the light along the path and
- γ_{n-1} is the measure of our path space (for technical details see [Pauly et al. 2000])

This integral can be evaluated using the following Monte Carlo estimator:

$$\langle I \rangle(\bar{p}_n) = \frac{f(\bar{p}_n)}{p(\bar{p}_n)} \tag{3}$$

where $f(\bar{p}_n)$ is the contribution of path $\bar{p}_n$ and $p(\bar{p}_n)$ is its pdf.

When evaluating eq. 2 we need to sample points that will form a path and at the same time accumulate the throughput and the pdf of the path. The contributions of all possible paths which start with vertices p_0, p_1 together give $L(p_1 \to p_0)$ — the exitant radiance from p_1 towards p_0. This means that in the context of volume rendering, two key operations that we need to discuss which are not needed in basic surface rendering are *distance sampling* (to sample points inside a medium) and *transmittance evaluation* (to obtain throughput).

Sampling a distance t to the next interaction in a medium is easy in a homogeneous volume where $t = -\frac{\ln(1-\xi)}{\mu_t}$ with ξ being a random value from $[0,1]$ and $p(t) = \mu_t e^{-t\mu_t}$ being the pdf of the sampled distance t. In heterogeneous media, this analytical approach fails because $\mu_t(x)$ varies across the medium. A common workaround is to homogenize the medium by introducing null particles which do not alter light in any way and their sole purpose is to level medium density. A *null-scattering* coefficient μ_n is therefore assigned to every points in the medium in such a way that the *majorant* $\bar{\mu} = \mu_t(x) + \mu_n(x)$ is constant. When $\bar{\mu} = \max_x \mu_t(x)$, the majorant is tight. Distance is then sampled according to the majorant ($t = -\frac{\ln(1-\xi)}{\bar{\mu}}$). After sampling t it is necessary to stochastically determine whether a *real-scattering* event ($\xi < \frac{\mu_t(x)}{\bar{\mu}}$), or a *null-scattering* event was sampled. We keep adding t samples until a real-scattering event is sampled.

Similarly, transmittance is easy to evaluate in homogeneous media, but for heterogeneous media, techniques such as *delta tracking* or *ratio tracking* are often used to estimate the transmittance integral.

2.2 Volume Rendering in Mitsuba 2

The organization of Mitsuba 2 is very similar to PBRT [Pharr et al. 2016] in that it contains a small module for every piece of functionality that it provides. For our purposes, the most interesting modules are *integrators* and *media*. The task of an integrator is to evaluate eq. 2 and for that purpose it implements various path sampling strategies. Medium is used by the integrator and contains more specific operations, such as distance sampling and transmittance evaluation.

The volumetric integrator of Mitsuba 2 combines unidirectional and next-event estimation (NEE) path sampling strategies using MIS[1]. Algorithm 1 explains how it handles medium interactions when constructing a path.

Algorithm 1 Medium interaction handling in a Uni+NEE MIS volumetric integrator in Mitsuba 2. ϕ is throughput divided by the pdf, L_e is emitter value, f_p is phase function value and p_e, p_p are corresponding pdfs.

Require: *medium*, *ray*, ϕ
1: …
2: $x, \phi_m \leftarrow$ sample_distance(*ray*)
3: ϕ *= ϕ_m ▷ accumulate medium throughput divided by the pdf
4: **if** $x \in$ *medium* **then**
5: $\omega_e, L_e, p_e \leftarrow$ sample_emitter(x) ▷ NEE
6: $f_p, p_p \leftarrow$ eval_phase(ω_e)
7: *result* += $\phi \cdot L_e \cdot f_p \cdot \frac{p_e}{p_e^2+p_p^2}$ ▷ power heuristic
8: $\omega, f_p, p_p \leftarrow$ sample_phase(x)
9: update_ray(*ray*, ω) ▷ scatter
10: $L_e, p_e \leftarrow$ connect_to_light(*ray*) ▷ unidirectional path sampling
11: *result* += $\phi \cdot L_e \cdot \frac{p_p}{p_p^2+p_e^2}$ ▷ power heuristic
12: **else**
13: *medium* escaped
14: …

Function `connect_to_light()` in algorithm 1 checks whether there is an emitter in the given direction and if there is, it also scales its value by the transmittance of all media that are in the way. It does that by simply accumulating transmittance along the whole intersection between the ray and the media. As mentioned in the previous section, for homogeneous media transmittance is analytically evaluable and for heterogeneous media it needs to be estimated via ratio tracking.

Function `sample_distance()` in algorithm 1 always returns the sampled point and ϕ_m between the sampled point and the previous point. $\phi_m = \frac{T(x_{n-1},x_n)}{p_{\text{escape}}}$ if the sampled point is outside of the medium and $\phi_m \frac{\mu_s \cdot T(x_{n-1},x_n)}{p(x_n)}$ otherwise. The function has two implementations, one for homogeneous and one for heterogeneous media. The homogeneous version supports spectrally varying μ_t and is described in algorithm 2. The heterogeneous version only supports monochromatic μ_t and uses delta tracking which handles null-scattering events. It is described in algorithm 3.

Refer to appendix A for the `C++` code which describes the API of medium in Mitsuba 2.

Algorithm 2 Sampling distance in homogeneous media in Mitsuba 2. μ_t is spectrally varying and it is therefore an array of length 4. This corresponds to hero-wavelength sampling which Mitsuba 2 does. The first wavelength is the hero and the other three are sampled at fixed intervals from the hero such that the whole spectrum is uniformly covered. This integrator, however, does not sample according to the hero wavelength, but according to their average instead.

Require: *ray*
1: $t \leftarrow \frac{-\ln(1-\xi)}{\text{avg}(\mu_t)}$ ▷ ξ is a random value $\in [0, 1]$
2: $\phi \leftarrow \frac{e^{-t\cdot\mu_t}}{\text{avg}(\mu_t)\cdot e^{-t\cdot\text{avg}(\mu_t)}}$ ▷ $e^{-\int_0^t \mu_t \,\mathrm{d}s} = e^{-t\cdot\mu_t}$
3: **if** t is inside medium **then**
4: ϕ *= μ_s
5: **return** point(*ray*, t), ϕ

Algorithm 3 Sampling distance in heterogeneous media in Mitsuba 2 using delta tracking.

Require: *ray*
1: $t \leftarrow 0$
2: **loop**
3: t += $\frac{-\ln(1-\xi_1)}{\bar{\mu}}$ ▷ ξ_1 is a random value $\in [0, 1]$
4: **if** t is outside medium **then**
5: **return** point on the boundary, 1 ▷ $\frac{T(x_{n-1},x_n)}{p_{\text{escape}}} = 1$
6: **if** $\xi_2 > \frac{\mu_t}{\bar{\mu}}$ **then**
7: **continue** ▷ null event; ξ_2 is a random value $\in [0, 1]$
8: **return** point(*ray*, t), $\frac{\mu_s}{\mu_t}$

2.3 Null-scattering Path Integral Formulation

The main issue of the method described in the previous section is that transmittance is not analytically evaluable. As a result, we do not have access to path pdfs and a separate algorithm for heterogeneous media is needed. To solve this issue, [Miller et al. 2019] have presented a null-scattering path integral formulation of light transport in participating media. This path integral has the same form as eq. 2, but the T term and the γ measure are different. For technical details refer to the paper, we will only mention the implications and benefits of this new formulation.

The key difference is that null-scattering events are explicitly covered in the integral. They still do not alter light in any way, but now the paths that we integrate over can contain them and they can be measured along segments between two consecutive real scattering vertices by the redefined measure γ.

[1] This MIS only happens per bounce, not per full path which, as was mentioned in section 1, is not possible to realize in the current framework.

In practice, after sampling a vertex, we stochastically decide whether the vertex is a real scattering vertex, a null-scattering vertex, or an absorption vertex and then accumulate throughput and pdfs, add the vertex to the path and continue constructing it accordingly. Sampling is guided by the majorant and because in homogeneous media, $\bar{\mu} = \mu_t$, we can use the same algorithm for both homogeneous and heterogeneous media.

3 NULL-SCATTERING PATH INTEGRAL FORMULATION IN MITSUBA 2

We have implemented the path integral from [Miller et al. 2019] in Mitsuba 2 and in this section we overview what has changed in comparison with the functionality presented in section 2.2.

The basic idea when implementing the new path integral is that the integrator does more work than before and medium functionality is limited only to coefficient lookups and analytic evaluation of sampling and transmittance formulas. Another crucial difference is that throughput and pdfs are no longer accumulated together in an already divided form ϕ, but separately in order to have the full path pdf available for each sampling strategy. This can be for example sampling according to a specific color component or using NEE. Our new integrator does hero wavelength sampling [Wilkie et al. 2014] and combines it with unidirectional and NEE sampling. This means that we need to track 8 throughputs and pdfs (4 color components for each unidirectional and NEE strategy). The way our integrator handles medium interactions when constructing a path is described in algorithm 4.

Interaction sampling is simplified and a single implementation handles both homogeneous and heterogeneous media. It is described in algorithm 5. After an interaction is sampled, we only know the transmittance between this point and the previous one. We now have to explicitly find out in the integrator what kind of interaction we have sampled and what the corresponding throughput and pdf is. Interaction type is determined stochastically in `sample_event()` based on medium coefficients at the location of the interaction.

Function `connect_to_light()` is adapted to track throughput and pdfs correctly according to the new path integral formulation. It works in two modes: delta tracking and ratio tracking. The former is useful when connecting to light by unidirectional sampling and the latter is useful in NEE. The function is described in algorithm 6.

Refer to appendix B for the C++ code which describes the API of medium in our implementation.

4 RESULTS

We have reproduced the results of [Miller et al. 2019] to show that our implementation works properly. Firstly, we present images rendered only using hero wavelength sampling and unidirectional sampling. This integrator is called *Spectral MIS*. We compare this method with *independent tracking* which estimates each color component separately and this leads to excessive color noise. See figure 2 for the comparison.

Secondly, we show images from the integrator described in algorithm 4 which we call *Uni+NEE MIS*. The comparison in figure 3 shows two extreme cases where unidirectional and NEE each fail on their own.

Algorithm 4 Medium interaction handling in our null-scattering-aware Uni+NEE MIS volumetric integrator with hero wavelength sampling. f and p are arrays of length 4 containing path throughput and pdf respectively. For other notation, see algorithm 1

Require: *medium*, *ray*, f, p
1: ...
2: $x, f_m \leftarrow$ sample_interaction(*ray*)
3: $f\ {}^*\!= f_m, p\ {}^*\!= f_m$ ▷ f_m contains only medium transmittance
4: **if** $x \in$ *medium* **then**
5: $e \leftarrow$ sample_event(x) ▷ based on $\frac{\mu_s(x)}{\bar{\mu}}$, $\frac{\mu_a(x)}{\bar{\mu}}$ and $\frac{\mu_n(x)}{\bar{\mu}}$
6: **if** e is absorption **then**
7: $f \leftarrow 0, p \leftarrow 0$ ▷ path termination
8: **else if** e is scattering **then**
9: $f\ {}^*\!= \mu_s(x), p\ {}^*\!= \mu_s(x)$
10: $\omega_e \leftarrow$ sample_emitter_direction(x) ▷ NEE
11: *nee_ray* $\leftarrow$ spawn_ray(x, ω_e)
12: $f_{\text{nee}}, p_{\text{nee}}, f_{\text{uni}}, p_{\text{uni}}, L_e \leftarrow$ connect_to_light(*nee_ray*, nee)
13: $f_p, p_p \leftarrow$ eval_phase(ω_e)
14: *result* += $L_e \cdot \frac{f \cdot f_{\text{nee}} \cdot f_p}{\text{average}(p \cdot p_{\text{nee}} \cdot p_p, p \cdot p_{\text{uni}} \cdot p_p)}$ ▷ balance heuristic
15: $\omega, f_p, p_p \leftarrow$ sample_phase(x)
16: update_ray(*ray*, ω) ▷ scatter
17: $f_{\text{nee}}, p_{\text{nee}}, f_{\text{uni}}, p_{\text{uni}}, L_e \leftarrow$ connect_to_light(*ray*, uni)
18: $f_p, p_p \leftarrow$ eval_phase(ω)
19: *result* += $L_e \cdot \frac{f \cdot f_{\text{uni}} \cdot f_p}{\text{average}(p \cdot p_{\text{nee}} \cdot p_p, p \cdot p_{\text{uni}} \cdot p_p)}$ ▷ balance heuristic
20: **else if** e is null **then**
21: $f\ {}^*\!= \mu_n(x), p\ {}^*\!= \mu_n(x)$
22: move *ray* to x
23: **else**
24: *medium* escaped
25: ...

Algorithm 5 Sampling interactions in the new implementation. Distance is sampled according to hero wavelength.

Require: *ray*
1: $t \leftarrow \frac{-\ln(1-\xi)}{\bar{\mu}^0}$ ▷ hero wavelength has index 0
2: **if** t is inside medium **then**
3: $f \leftarrow e^{-t \cdot \bar{\mu}}$
4: **return** point(*ray*, t), f
5: **else**
6: $x \leftarrow$ point on the boundary
7: $d \leftarrow x -$ *ray* origin
8: $f \leftarrow e^{-d \cdot \bar{\mu}}$
9: **return** x, f

5 SHORTCOMINGS OF THE PRACTICAL IMPLEMENTATION

In this section, we will discuss two major issues with the implementation of the new path integral formulation. The first one is related to terminating paths when an absorption event is sampled and the second one is a numerical issue with accumulating throughput and pdfs for the full path separately.

Algorithm 6 Connecting to light in the new implementation.

Require: *ray*, *mode*

$f_{\text{nee}} \leftarrow 1, p_{\text{nee}} \leftarrow 1, f_{\text{uni}} \leftarrow 1, p_{\text{uni}} \leftarrow 1$
loop
  if *ray* origin is inside medium **then**
    $x, f_m \leftarrow$ sample_interaction(*ray*)
    $f_{\text{nee}} \mathrel{*}= f_m, p_{\text{nee}} \mathrel{*}= f_m, f_{\text{uni}} \mathrel{*}= f_m, p_{\text{uni}} \mathrel{*}= f_m$
    if $x \notin$ medium **then**
      if $x \in$ emitter **then**
        $p_{\text{nee}} \mathrel{*}= p_{\text{emitter\_dir}}$ ▷ pdf of sampling the emitter via NEE
        return $f_{\text{nee}}, p_{\text{nee}}, f_{\text{uni}}, p_{\text{uni}}, L_e$
      if *ray* escaped or occluded **then**
        return 0, 0, 0, 0, 0
    else
      if *mode* is uni **then**
        $e \leftarrow$ sample_event(x) ▷ based on $\frac{\mu_s}{\bar{\mu}}$, $\frac{\mu_a}{\bar{\mu}}$ and $\frac{\mu_n}{\bar{\mu}}$
        if e is not null **then**
          return 0, 0, 0, 0, 0
        $f_{\text{nee}} \mathrel{*}= \mu_n, p_{\text{nee}} \mathrel{*}= \bar{\mu}, f_{\text{uni}} \mathrel{*}= \mu_n, p_{\text{uni}} \mathrel{*}= \mu_n$
  else
    $x \leftarrow$ ray_intersect(*ray*)
    if $x \in$ emitter **then**
      $p_{\text{nee}} \mathrel{*}= p_{\text{emitter\_dir}}$ ▷ pdf of sampling the emitter via NEE
      return $f_{\text{nee}}, p_{\text{nee}}, f_{\text{uni}}, p_{\text{uni}}, L_e$
    if *ray* escaped or occluded **then**
      return 0, 0, 0, 0, 0
  move *ray* origin to x

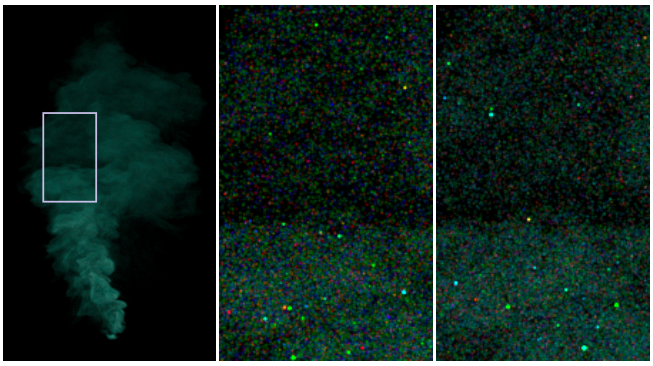


Fig. 2. Spectral MIS (which uses hero wavelength sampling) is good at eliminating color noise and doing so efficiently. The smoke plume is a heterogeneous medium with an isotropic phase function, $\frac{\mu_s}{\mu_t} = 0.99$ and spectrally varying $\mu_t = \langle 430 : 8, 540 : 40, 572 : 8 \rangle$ (the rest of the spectrum is interpolated by Mitsuba 2). Both images have been rendered at 64spp. Reference was rendered by Uni+NEE MIS at 1024spp.

Currently, the volumetric integrator in Mitsuba 2 uses every path that it constructs and scales its contribution appropriately with respect to $\frac{\mu_s}{\mu_t}$ (also known as the *albedo*). The new implementation is different in that in its basic monochromatic unidirectional form, paths which sample an absorption event are terminated and paths

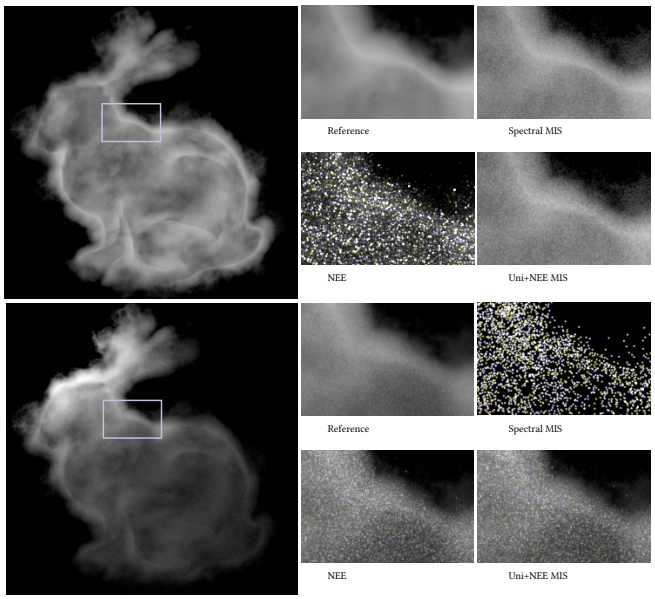
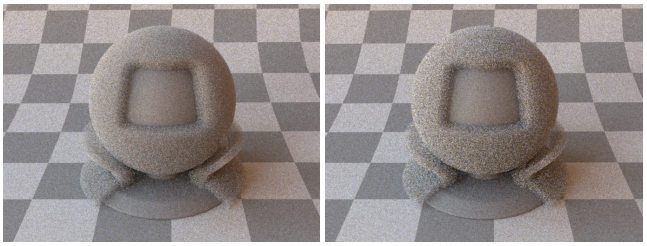


Fig. 3. Uni+NEE MIS handles both extreme cases robustly. The upper bunny cloud is back-lit by a large area light source, has Henyey-Greenstein phase function with $g = 0.99$, $\frac{\mu_s}{\mu_t} = 0.99$ and monochromatic $\mu_t = 4$. The lower bunny cloud is side-lit by a smaller area light source, has Henyey-Greenstein phase function with $g = 0.9$, $\frac{\mu_s}{\mu_t} = 0.99$ and monochromatic $\mu_t = 16$. All images have been rendered at 64spp except for the reference which was rendered at 1024spp. The upper reference was rendered by Spectral MIS and the lower by NEE.



(a) Current Mitsuba 2 (b) Our new implementation

Fig. 4. Both images have been rendered by unidirectional integrators and both show a homogeneous medium which eliminates the effect of null-scattering events. The same number of samples has been used in both cases. There is visibly more noise in the image on the right caused by path termination in the new path integral formulation.

that are completed are not attenuated by any medium coefficients at all. As a result, this increases the amount of noise slightly because many paths have zero contribution. See fig. 4 for an illustration of this.

The second issue is a numeric one. In the new implementation, we accumulate throughput and pdfs separately for the whole path and only divide the two at the end. This is necessary for various sampling strategies to be combined via MIS. The throughput variable, for

example, then looks like this: $f = \prod_{i=1}^{k} \mu_i$ where k is the length of the path and μ_i is a medium coefficient. In dense media, this value can easily grow above 10^{40} which cannot be represented by a single-precision floating point number. [Miller et al. 2019] introduce a simple trick in their implementation [Miller 2019]. They accumulate $\frac{p}{f}$ for each combination of color components, so in the case of RGB rendering, there would be $3 \cdot 3 = 9$ accumulators. Now consider the following equation:

$$\frac{3}{\frac{p_r}{f_r} + \frac{p_g}{f_r} + \frac{p_b}{f_r}} = \frac{3}{\frac{p_r + p_g + p_b}{f_r}} = \frac{3 \cdot f_r}{\sum_{c \in \{r,g,b\}} p_c} = \frac{f_r}{\text{average}(p)} \quad (4)$$

This is how we can then obtain the balance heuristic weight for the contribution in the r channel.

The problem with this solution is that it is time-consuming and also specific to the balance heuristic. It might depend on the use case whether we are willing to accept these limitations.

6 CONCLUSIONS

We have implemented a null-scattering path integral formulation introduced by [Miller et al. 2019] in Mitsuba 2. We have modified the code architecture and on top of that we have implemented an integrator which combines hero wavelength sampling [Wilkie et al. 2014] with unidirectional and NEE strategies. As a result, it is now possible to render spectrally varying heterogeneous media in Mitsuba 2 (see fig. 1). Our new implementation also simplifies classes related to homogeneous and heterogeneous media because they now share functions for sampling distance and evaluating transmittance.

We have discussed two drawbacks of the new path integral which were not obvious from the original paper. The first one is a slight increase in noise given by path termination when absorption events are sampled inside media. The second one is a lower time efficiency and flexibility with regards to MIS heuristics. This is given by the fact that full path throughput and pdfs are now accumulated separately and extra effort is required when keeping these values in single-precision floating point variables.

Future work related to this project should include precise time benchmarks and thus a more thorough comparison with the current volumetric integrator in Mitsuba 2. This would help conclude whether the simplified architecture and greater flexibility when combining various sampling techniques using MIS outweighs the increased noise and a more complicated throughput and pdf tracking. At the same time, more effort should be dedicated to investigating whether the drawbacks can be eliminated. Ultimately, this would help to decide whether the null-scattering path integral formulation should be the default choice in the rendering pipeline of Mitsuba 2 in the future.

ACKNOWLEDGMENTS

The author would like to thank Wenzel Jakob for the supervision of this project and Tizian Zeltner for patiently advising the author on the direction of the project and on the inner workings of Mitsuba 2.

Media used in figures 1, 2 and 3 come from OpenVDB [Museth et al. 2013] and the medium in figure 4 is from the Mitsuba scene repository.

REFERENCES

Bailey Miller. 2019. nullpath. https://github.com/baileymiller/nullpath. Commit: 7d54163b8b020d84eec454d3448a54c7d30953ff.

Bailey Miller, Iliyan Georgiev, and Wojciech Jarosz. 2019. A null-scattering path integral formulation of light transport. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 38, 4 (jul 2019). https://doi.org/10.1145/3306346.3323025

Ken Museth, Jeff Lait, John Johanson, Jeff Budsberg, Ron Henderson, Mihai Aldén, Peter Cucka, David R. Hill, and Andrew Pearce. 2013. OpenVDB: an open-source data structure and toolkit for high-resolution volumes. In *SIGGRAPH '13.*

Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. 2019. Mitsuba 2: A Retargetable Forward and Inverse Renderer. *Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 38, 6 (Nov. 2019). https://doi.org/10.1145/3355089.3356498

Jan Novák, Iliyan Georgiev, Johannes Hanika, Jaroslav Křivánek, and Wojciech Jarosz. 2018. Monte Carlo Methods for Physically Based Volume Rendering. In *ACM SIGGRAPH 2018 Courses (SIGGRAPH '18).* ACM, New York, NY, USA, Article 14, 1 pages. https://doi.org/10.1145/3214834.3214880

Mark Pauly, Thomas Kollig, and Alexander Keller. 2000. Metropolis Light Transport for Participating Media. In *Proceedings of the Eurographics Workshop on Rendering Techniques 2000.* Springer-Verlag, Berlin, Heidelberg, 11–22.

Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2016. *Physically Based Rendering: From Theory to Implementation (3rd ed.)* (3rd ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 1266 pages.

A. Wilkie, S. Nawaz, M. Droske, A. Weidlich, and J. Hanika. 2014. Hero Wavelength Spectral Sampling. In *Proceedings of the 25th Eurographics Symposium on Rendering (EGSR '14).* Eurographics Association, Goslar, DEU, 123–131. https://doi.org/10.1111/cgf.12419

A OLD MEDIUM API

Here is a snippet from the `medium.h` file in Mitsuba 2. It defines an abstract class `Medium` from which classes for homogeneous and heterogeneous media are derived:

```
virtual std::tuple<SurfaceInteraction3f, MediumInteraction3f,
    ↪ Spectrumf>
sample_distance(const Scene *scene, const Ray3f &ray,
        const Point2f &sample, Sampler *sampler) const = 0;

virtual Spectrumf eval_transmittance(const Ray3f &ray,
                                     Sampler *sampler) const = 0;
```

B NEW MEDIUM API

Here is a snippet from the `medium.h` file in our implementation. Notice that fewer random samples are now required and that the only virtual function which needs to be implemented in homogeneous and heterogeneous medium classes is `get_coefficient()`. This hides whether coefficients are constant, or spatially varying.

```
std::tuple<SurfaceInteraction3f, MediumInteraction3f,
    ↪ Spectrumf>
sample_interaction(const Scene *scene, const Ray3f &ray,
                   Float sample) const;

Spectrumf eval_transmittance(const Ray3f &ray) const;

uint32_t sample_event(const MediumInteraction3f &mi,
                      Float sample, int channel) const;

virtual Spectrumf get_coefficient(const MediumInteraction3f
                          &mi, uint32_t event_type) const = 0;
```