# Physically based Volume Rendering

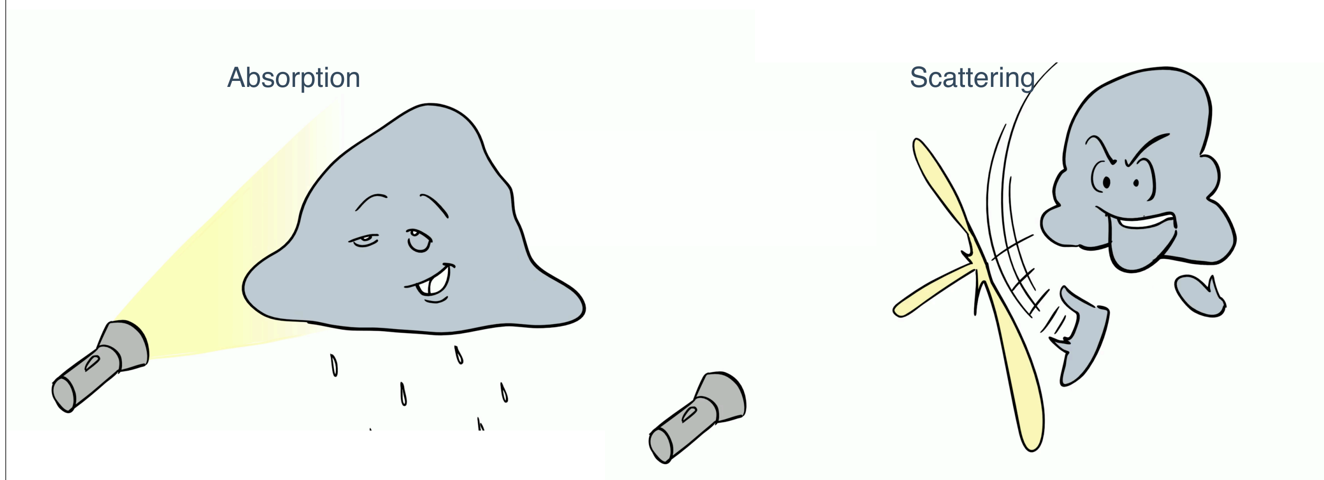
*Philipp Slusallek*    *Karol Myszkowski*
Gurprit Singh

# Monte Carlo Methods for Physically based Volume Rendering

*Slides borrowed from SIGGRAPH 2018 Course by*

*Jan Novák, Iliyan Georgiev, Johannes Hanika, Jaroslav Křivánek, Wojciech Jarosz*

# FUNDAMENTALS





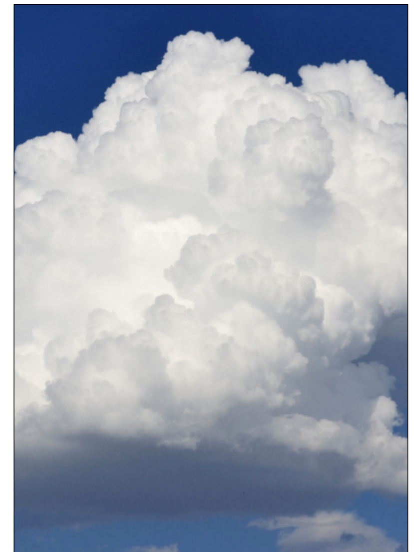

In this part, we will look at the basic interactions between light and matter and how to describe them mathematically, deriving the so-called Volume Rendering Equation. And then we'll talk about how to solve it.

## FUNDAMENTALS

Absorption

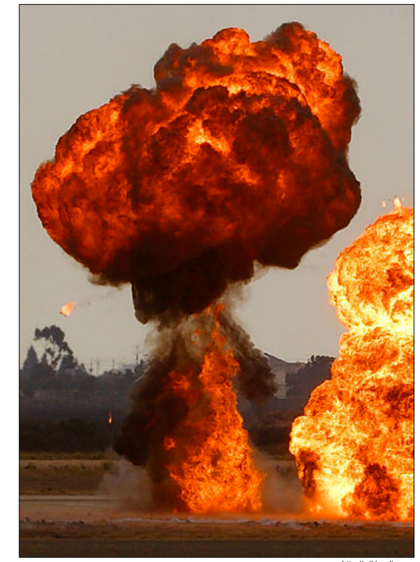
http://commons.wikimedia.org

Scattering



http://coclouds.com

Emission



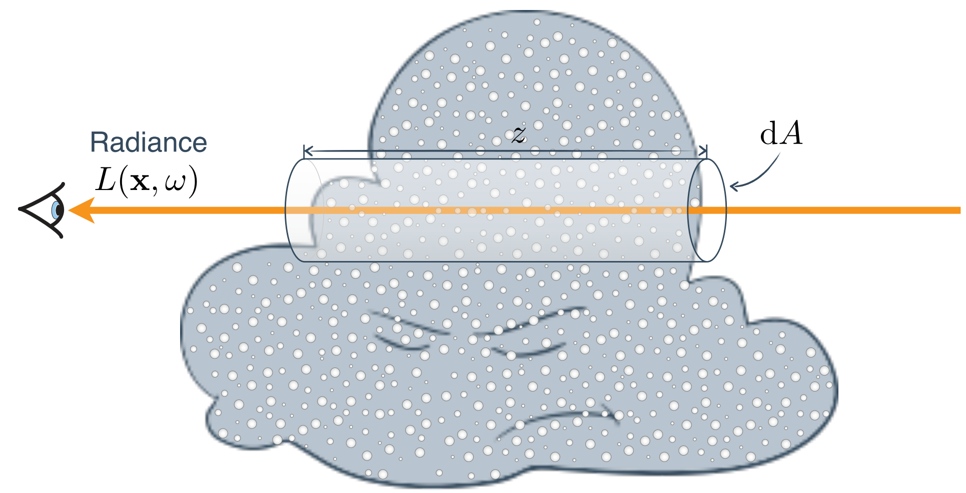
http://wikipedia.org





Here are a few examples that show the three main process that we will discuss:
absorption, scattering, and emission

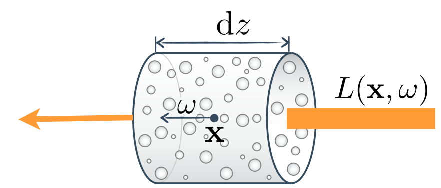
# RADIATIVE TRANSFER






These processes impact radiance, i.e. the radiative flux traveling through a differential beam.
The flux will be increased or decreased depending on the optical properties of the medium.

# ABSORPTION




$$\frac{\mathrm{d}L}{\mathrm{d}z} = -\mu_{\mathrm{a}}(\mathbf{x})L(\mathbf{x}, \omega)$$

$\mu_{\mathrm{a}}$ - absorption coefficient

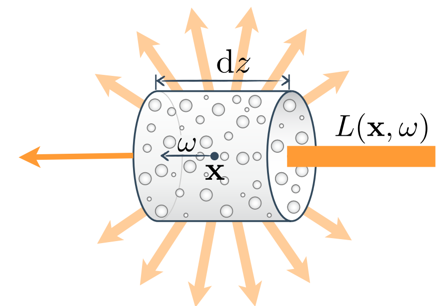


In order to formalize these changes mathematically, we will start with a differential beam segment.

I have a few particle here, but these is just for illustration.
We typically do not represent the material particles explicitly, we rather model the medium statistically quantifying the density of collisions per unit of travelled distance.

Some of these particles may be absorptive. If we look at the radiance passing through the differential segment along a given direction omega, it will be reduced, and the rate of change depends on the absorption coefficient, which quantifies the probability density of light being absorbed per unit distance.

# OUT-SCATTERING




$$\frac{\mathrm{d}L}{\mathrm{d}z} = -\mu_\mathrm{s}(\mathbf{x}) L(\mathbf{x}, \omega)$$

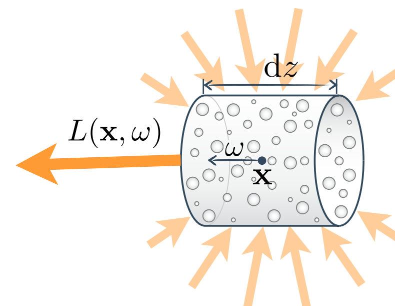
$\mu_\mathrm{s}$ - scattering coefficient



If there are scattering particles, some of the incident light will be scattered out, away from the direction of its initial travel.

Analogously to absorption, these losses are proportional to a coefficient—the scattering coefficient—which quantifies the probability density of a scattering collision per unit distance.

# IN-SCATTERING




In-scattered radiance

$$L_s(\mathbf{y},\omega) = \int_{S^2} f_p(\omega,\bar{\omega})L(\mathbf{y},\bar{\omega})\mathrm{d}\bar{\omega}$$

$$\frac{\mathrm{d}L}{\mathrm{d}z} = \mu_s(\mathbf{x})L_s(\mathbf{x},\omega)$$

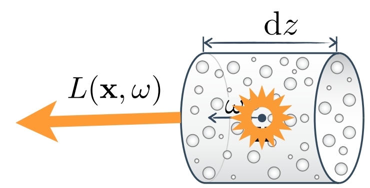
$\mu_s$ - scattering coefficient



Whenever light can out-scatter, there is a chance that some may also scatter into the direction omega.

The main difference to out-scattering is the sign, in-scattering is positive. The other difference is the radiance function. Here we have the so-called in-scattered radiance. We will discuss it in detail a bit later.

# EMISSION




$$\frac{\mathrm{d}L}{\mathrm{d}z} = \mu_\mathrm{a}(\mathbf{x}) L_\mathrm{e}(\mathbf{x}, \omega)$$

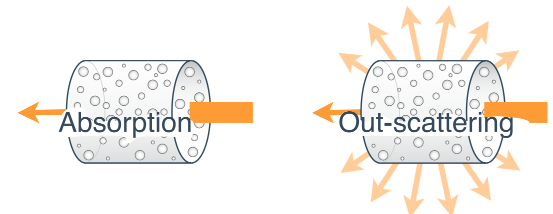
$L_\mathrm{e}$ - emitted radiance



Finally, the radiance may also increase due to emission from the volume, for instance as a result of incandescent or luminescent processes.

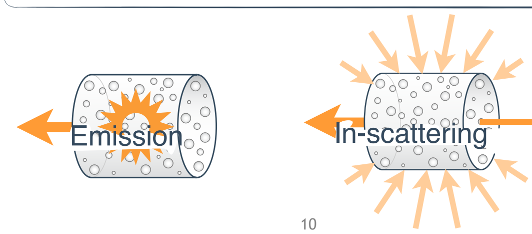
Here we made the choice to model the emission as a product of emitted radiance and absorption coefficient. This has two motivations, although other definitions are justifiable as well and may indeed be better in specific situations. The first reason is the notational convenience in later part of the talk. The second reason is based on a physically motivated argument. For emission to exist, there should be some material that has the ability to absorb energy and release it in the form of photons—e.g. through the process of incandescence.

# RADIATIVE TRANSFER EQUATION




$$\frac{\mathrm{d}L(\mathbf{x},\omega)}{\mathrm{d}z} = -\mu_\mathrm{a}(\mathbf{x})L(\mathbf{x},\omega) \quad -\mu_\mathrm{s}(\mathbf{x})L(\mathbf{x},\omega) \qquad \text{Losses}$$

$$+\mu_\mathrm{a}(\mathbf{x})L_\mathrm{e}(\mathbf{x},\omega) + \mu_\mathrm{s}(\mathbf{x})L_\mathrm{s}(\mathbf{x},\omega) \qquad \text{Gains}$$




[Chandrasekhar 1960]



We can combine all the losses and the gains into a single differential equation—the so-called *radiative transfer equation* (RTE).

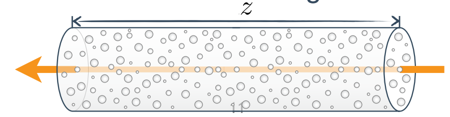
# RADIATIVE TRANSFER

Extinction coefficient $\mu_\mathrm{t}(\mathbf{x}) = \mu_\mathrm{a}(\mathbf{x}) + \mu_\mathrm{s}(\mathbf{x})$

$$\frac{\mathrm{d}L(\mathbf{x},\omega)}{\mathrm{d}z} = -\mu_\mathrm{t}(\mathbf{x})L(\mathbf{x},\omega) \quad \text{Losses}$$

$$+\mu_\mathrm{a}(\mathbf{x})L_\mathrm{e}(\mathbf{x},\omega) + \mu_\mathrm{s}(\mathbf{x})L_\mathrm{s}(\mathbf{x},\omega) \quad \text{Gains}$$

What about a finite-length beam?




[Chandrasekhar 1960]



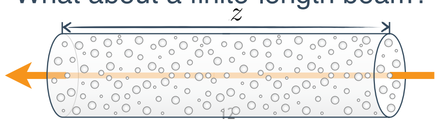
Since both absorption and out-scattering affect the same radiance function L, we can merge them into a single term. Here, mu_t is the sum of the absorption and scattering coefficient, and it is referred to as the extinction coefficient. It describes the probability density of any collision per unit distance.

The RTE is a differential equation operating on a differential beam segment. In rendering, however, we are typically working with beams with finite length. As such, we integrate both sides of the equation spatially, along direction omega, to obtain the integral form of RTE, which is more suitable for our needs.

# RTE — INTEGRAL FORM

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

What about a finite-length beam?



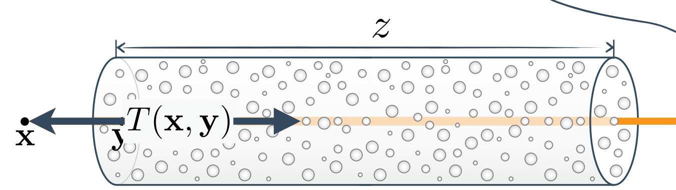



The integral form tells us how to compute radiance traveling through...

# RTE — INTEGRAL FORM

$$L(\mathbf{x},\omega) = \int_0^z T(\mathbf{x},\mathbf{y})\left[\mu_a(\mathbf{y})L_e(\mathbf{y},\omega) + \mu_s(\mathbf{y})L_s(\mathbf{y},\omega)\right]\mathrm{d}y$$

Transmittance $T(\mathbf{x},\mathbf{y}) = e^{-\int_0^y \mu_t(\mathbf{s})\mathrm{d}s}$ is the fraction of light that makes it from y to x




Optical thickness

$$\tau(\mathbf{x},\mathbf{y}) = \int_0^y \mu_t(\mathbf{s})\mathrm{d}s$$



... point x in direction omega: we need to integrate “back” along the ray and at each point y, we need to evaluate the gains and weight them down by the transmittance function that takes into account the losses on the way towards the point x.

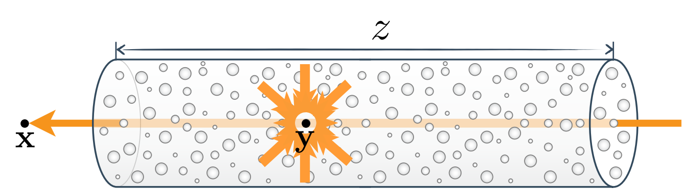
The transmittance function quantifies the fraction of light that makes it from one point to another. It’s derivation is know as the Beer-Lambert law, which tells us that light is reduced exponentially as a function of negative optical thickness.

The optical thickness is the integral of the extinction coefficient along the ray and it represents all the material that light could potentially interact with in-between x and y.

# RTE — INTEGRAL FORM

$$L(\mathbf{x},\omega) = \int_0^z T(\mathbf{x},\mathbf{y}) \left[\mu_\mathrm{a}(\mathbf{y}) L_\mathrm{e}(\mathbf{y},\omega) + \mu_\mathrm{s}(\mathbf{y}) L_\mathrm{s}(\mathbf{y},\omega)\right] \mathrm{d}y$$

Emission In-scattering






Let’s now have a closer look at the gains.

The first term here represents volumetric emission. The function L_e is typically defined by the artist, e.g. by measuring real objects, or using a fluid simulation.

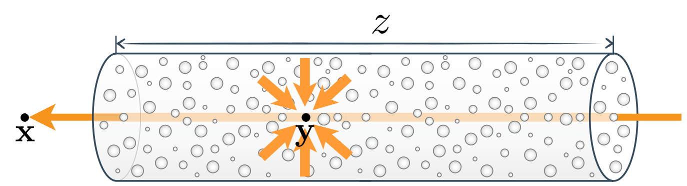
The second term is a little more complicated. It accounts for the effects of in-scattering.

# RTE — INTEGRAL FORM

$$L(\mathbf{x}, \omega) = \int_0^z T(\mathbf{x}, \mathbf{y}) \Big[ \mu_\mathrm{a}(\mathbf{y}) L_\mathrm{e}(\mathbf{y}, \omega) + \mu_\mathrm{s}(\mathbf{y}) L_\mathrm{s}(\mathbf{y}, \omega) \Big] \mathrm{d}y$$

$$L_\mathrm{s}(\mathbf{y}, \omega) = \int_{S^2} f_\mathrm{p}(\omega, \bar{\omega}) L(\mathbf{y}, \bar{\omega}) \mathrm{d}\bar{\omega}$$

Phase function

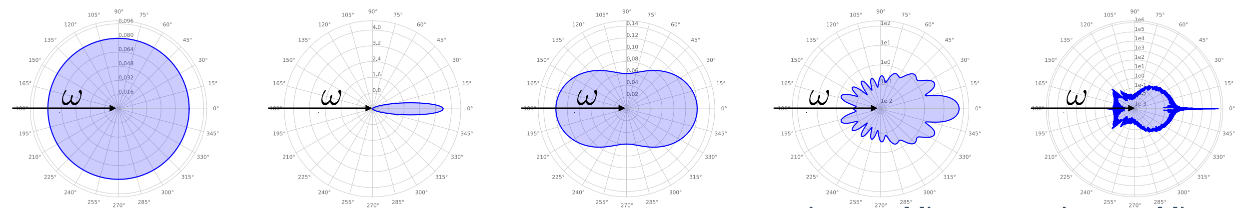





The in-scattered radiance function L_s is computed by integrating incident light over the sphere of all directions modulated by the phase function. The phase function captures the directional distribution of scattered light, and it is the volumetric analog of the BRDF.

# PHASE FUNCTION

$$f_{\mathrm{p}}(\omega, \bar{\omega})$$




Isotropic | Henyey-Greenstein | Rayleigh | Lorenz-Mie small particles | Lorenz-Mie large particles



Here are a few different models used for expressing the phase function.

# PHASE FUNCTION

Backward scattering PF

Smoke

Forward scattering PF

Steam





These images compare the visual appearance of predominantly backward- and forward-scattering media—smoke and steam.

# PHASE FUNCTION

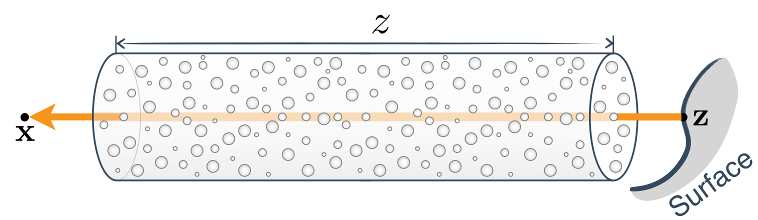
Isotropic PF

Forward scattering PF

## RTE — INTEGRAL FORM

$$L(\mathbf{x},\omega) = \int_0^z T(\mathbf{x},\mathbf{y})\Big[\mu_\mathrm{a}(\mathbf{y})L_\mathrm{e}(\mathbf{y},\omega) + \mu_\mathrm{s}(\mathbf{y})L_\mathrm{s}(\mathbf{y},\omega)\Big]\mathrm{d}y$$
$$+T(\mathbf{x},\mathbf{z})L_\mathrm{o}(\mathbf{z},\omega)$$

Background radiance






For completeness, we should also add a term that accounts for the surface at the end of the ray. We simply take the outgoing radiance from the surface and multiply it by the transmittance along the ray.

This equation here is often referred to as the *volume rendering equation* (VRE).

# VOLUME RENDERING EQUATION

$$L(\mathbf{x},\omega) = \int_0^z T(\mathbf{x},\mathbf{y})\Big[\mu_\mathrm{a}(\mathbf{y})L_\mathrm{e}(\mathbf{y},\omega) + \mu_\mathrm{s}(\mathbf{y})L_\mathrm{s}(\mathbf{y},\omega)\Big]\mathrm{d}y$$
$$+T(\mathbf{x},\mathbf{z})L_\mathrm{o}(\mathbf{z},\omega)$$

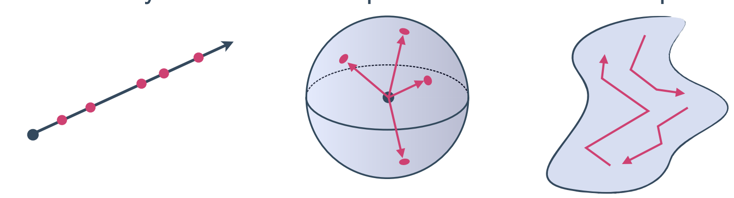
How do we solve it?



The main question that remains now is, how do we solve it?

# MONTE CARLO INTEGRATION

$$F = \int_{\mathcal{D}} f(x)\,\mathrm{d}x$$




$$\langle F \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}$$

Probability density function (PDF)



In this course, we focus on the various Monte Carlo methods developed in the past few decades.

Monte Carlo integration works in the following way: in order to integrate a function f over a domain D (e.g. a ray, sphere, path space), we generate a few points in D, evaluate the integrand at these points, and estimate the integral as a weighted average where each sample is divided by the probability density of drawing it.

In the following slides, we consider taking only one sample for notation brevity.

# VRE ESTIMATOR

$$\langle L(\mathbf{x},\omega)\rangle = \frac{T(\mathbf{x},\mathbf{y})}{p(y)}\left[\mu_{\mathrm{a}}(\mathbf{y})L_{\mathrm{e}}(\mathbf{y},\omega)+\mu_{\mathrm{s}}(\mathbf{y})L_{\mathrm{s}}(\mathbf{y},\omega)\right] + \frac{T(\mathbf{x},\mathbf{z})}{P(z)}L_{\mathrm{o}}(\mathbf{z},\omega)$$

$p(y)$ - probability density of distance y

$P(z)$ - probability of exceeding distance z



Applying such a one-sample estimator to the VRE is straightforward: we evaluate the integrand at a single randomly chosen distance y divided by the probability density of sampling the distance.

If the sampled distance happens to be past the nearest surface, we evaluate the second term (the radiance from the surface) divided by the *probability* of sampling a position behind the surface.

# VRE ESTIMATOR

Transmittance estimation

$$\langle L(\mathbf{x},\omega)\rangle = \frac{T(\mathbf{x},\mathbf{y})}{p(y)}\left[\mu_{\mathrm{a}}(\mathbf{y})L_{\mathrm{e}}(\mathbf{y},\omega)+\mu_{\mathrm{s}}(\mathbf{y})L_{\mathrm{s}}(\mathbf{y},\omega)\right]$$

$$+\frac{T(\mathbf{x},\mathbf{z})}{P(z)}L_{\mathrm{o}}(\mathbf{z},\omega)$$

Distance sampling

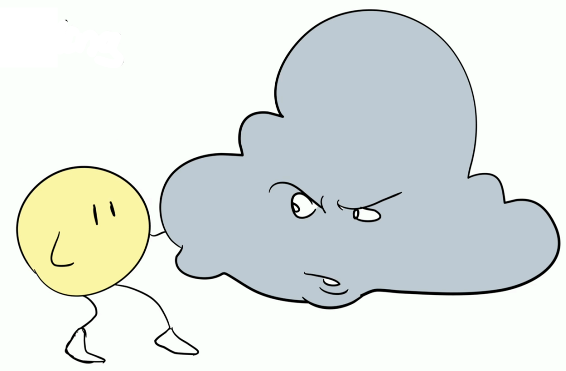


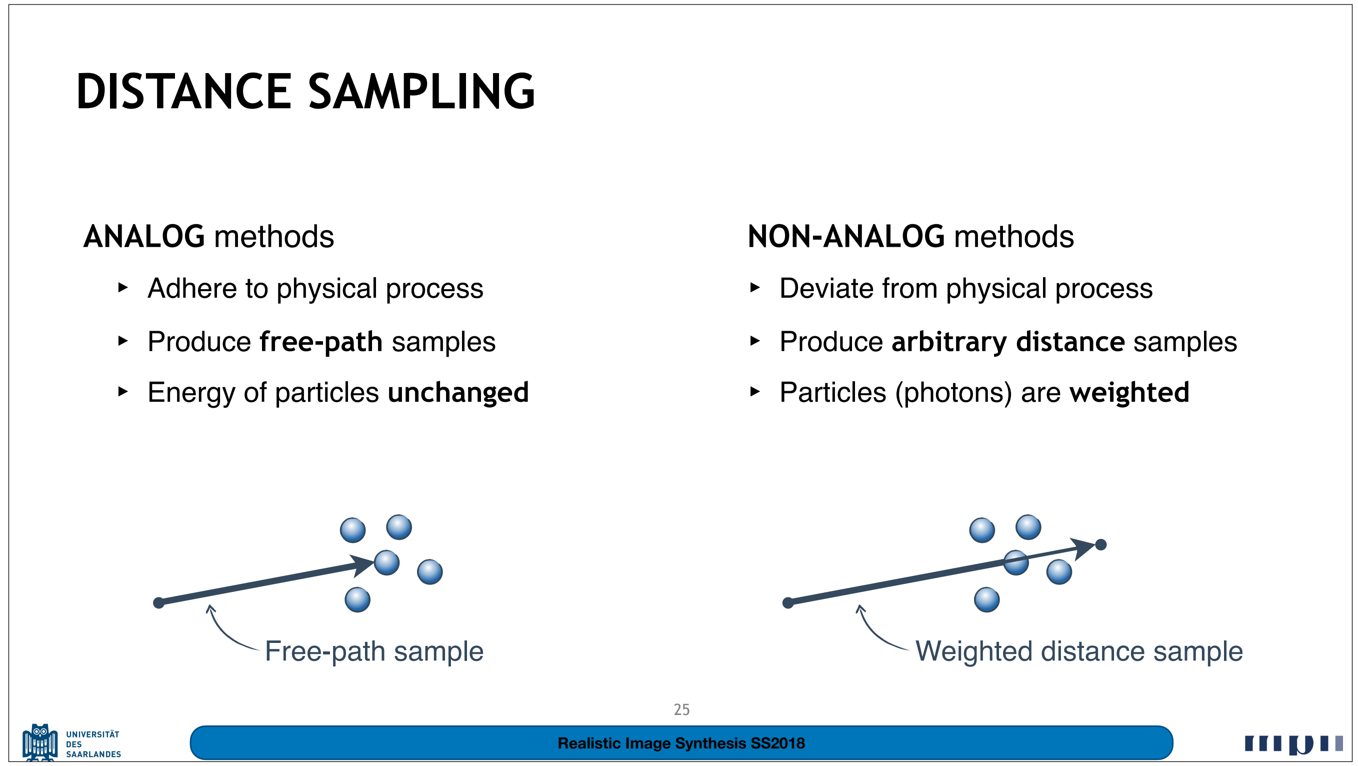
Sampling a distance and evaluating transmittance are the key building blocks of most volume-rendering algorithms.

Next, we discuss methods for sampling distances and then we describe approaches for estimating transmittance.

# DISTANCE SAMPLING

How far will photon travel before interacting with the medium?



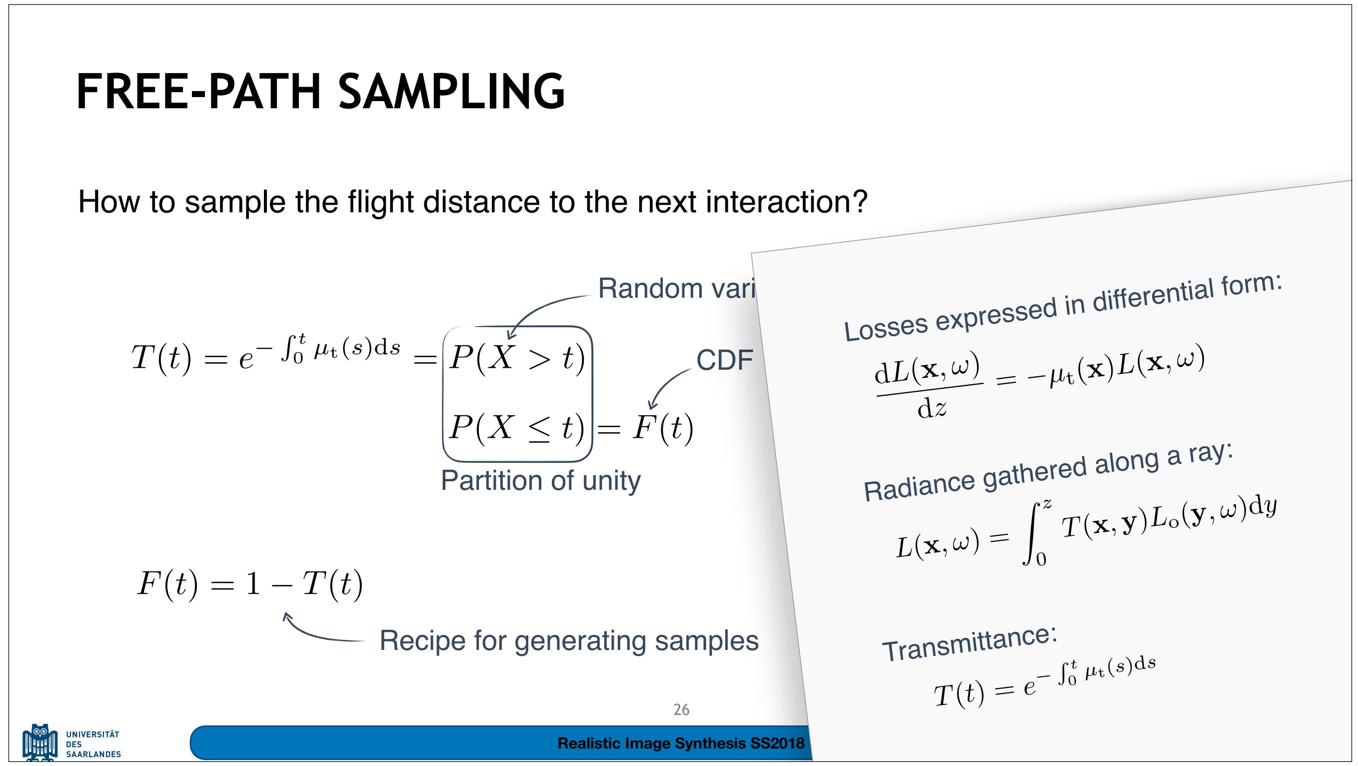



Distance-sampling methods can be classified as either *analog* or *non-analog* methods.

The terminology comes from neutron transport and refers to whether the method adheres to the physical process. A particle traced by an analog method is either scattered or absorbed strictly according the optical properties of the material. Such analog methods produce so-called free-path samples that are analogous to photon trajectories in the real world.

In contrast, non-analog methods deviate from the physical process producing distances that can be very different from real free paths. These methods can still achieve unbiased results if the samples are appropriately weighted to counteract the deviations from the physical process.

We will start with analog methods…




How would one go about sampling the free-flight distance to the next interaction?

If you recall, the interactions of radiance with the medium are described by the differential RTE (in box on the right).
[CLICK] Integrating it spatially gives us the relation between the incident and integrated outgoing radiance.
[CLICK] And these are related using transmittance, which quantifies the probability that a photon will travel beyond a given distance t.

How can we use this for sampling distances?

Notice, that the complementary probability that a photon will [CLICK] interact before reaching t meets the definition of a cumulative distribution function. [CLICK] Since these probabilities partition unity, we can define the CDF for sampling distances as the complementary probability to transmittance.

# FREE-PATH SAMPLING

Cumulative distribution function (**CDF**)

$$F(t) = 1 - T(t) = 1 - e^{-\tau(t)}$$

Probability density function (**PDF**)

$$p(t) = \frac{\mathrm{d}F(t)}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\left(1 - e^{-\tau(t)}\right) = \mu_\mathrm{t}(t)e^{-\tau(t)}$$

Inverted cumulative distr. function (**CDF$^{-1}$**)

$$\xi = 1 - e^{-\tau(t)}$$

Solve for t

$$\int_0^t \mu_\mathrm{t}(s)\mathrm{d}s = -\ln(1-\xi)$$

Approaches for finding t:

1) ANALYTIC (closed-form CDF$^{-1}$)
2) SEMI-ANALYTIC (regular tracking)
3) APPROXIMATE (ray marching)



We need two more things…

We need to express the PDF of individual samples, which is achieved by differentiating the CDF.

We also need to invert the CDF to obtain a recipe how to get from a random number to a sampled distance t. This leads to an equation, where we have the optical-thickness integral on the LHS, and a natural logarithm on the RHS. In other words, we are looking for a distance t at which the optical thickness equals to the negative logarithm.

Depending on the extinction function mu_t, we can find the distance either analytically, semi-analytically, or in an approximate manner. We will now look at these approaches in detail.

# ANALYTIC APPROACH

Inverted cumulative distr. function (**CDF**$^{-1}$)

$$\int_0^t \mu_\mathrm{t}(s)\mathrm{d}s = -\ln(1-\xi)$$

Some simple volumes permit closed-form solutions

Example: **homogeneous** medium ($\mu_\mathrm{t}(\mathbf{x}) = \mu_\mathrm{t}$)

Opt. thickness

$$\int_0^t \mu_\mathrm{t}(s)\mathrm{d}s = t\mu_\mathrm{t}$$

$\Rightarrow$

Inverted CDF

$$F^{-1}(\xi) = -\frac{\ln(1-\xi)}{\mu_\mathrm{t}}$$



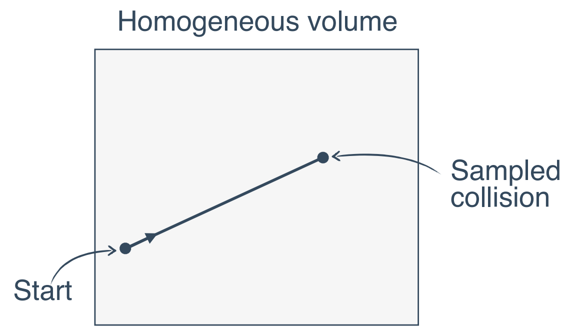
Some extinction functions allow expressing the optical thickness in closed form.

One such examples is a homogeneous medium, where the extinction function is simply a constant. The optical thickness is thus a simple linear function of t. When we plug this back to the CDF equation, we can solve it easily by dividing the RHS by the constant extinction. This yields the inverted CDF.

## ANALYTIC APPROACH

Inverted cumulative distr. function (**CDF**$^{-1}$)

$$\int_0^t \mu_t(s)\mathrm{d}s = -\ln(1-\xi)$$




Sampling in homogeneous vol:

1) Draw a random number $\xi$
2) Set $t = -\frac{\ln(1-\xi)}{\mu_t}$
3) Set $p(t) = \mu_t e^{-t\mu_t}$



Sampling the free-flight distance in homogeneous volumes is therefore straightforward: given a position, we draw a random number and compute the free-flight distance to the first collision and the PDF. Everything is analytic and easy to implement.

Unfortunately, most volumes do not permit closed-form free-path sampling.

# REGULAR TRACKING (SEMI-ANALYTIC)

For piecewise-simple (e.g. piecewise-constant), summation replaces integration

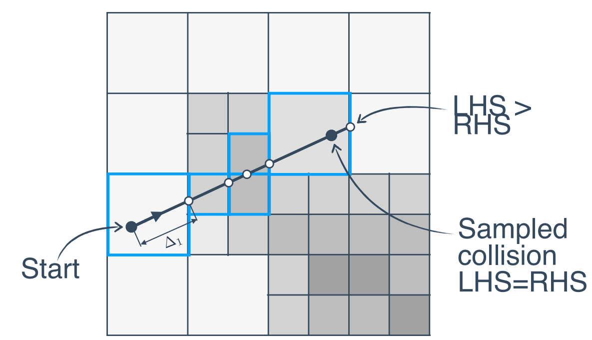
$$\int_0^t \mu_\mathrm{t}(s)\mathrm{d}s = -\ln(1-\xi)$$

$$\sum_{i=1}^{k} \mu_{\mathrm{t},i}\,\Delta_i = -\ln(1-\xi)$$

Regular tracking:

1) Draw a random number $\xi$
2) While LHS < RHS
   move to the next intersection
3) Find the exact location
   in the last segment analytically

(Hierarchical) voxel grid






If the volume is can be represented as a collection of simple volumes, for instance a voxel grid where each cell has constant extinction, the integration on the LHS will be replaced by a summation.

This leads to an iterative algorithm, called regular tracking: we first sample the optical thickness on the RHS, and step through the homogeneous partitions accumulating the optical thickness until the LHS exceeds the sampled value. Once this happens, we know the free-path sample is somewhere within the last partition and since the volume is simple, we can find the exact location analytically.

The main drawback of regular tracking is the necessity to find all the intersections with interfaces that separate individual homogeneous regions—this can be fairly expensive.

# REGULAR TRACKING (SEMI-ANALYTIC)

For piecewise-simple (e.g. piecewise-constant), summation replaces integration

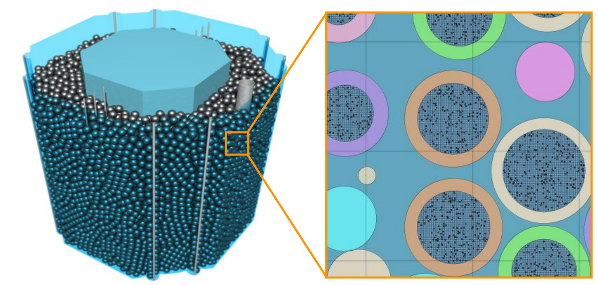
$$\int_0^t \mu_{\mathrm{t}}(s)\mathrm{d}s = -\ln(1-\xi)$$

$$\sum_{i=1}^{k} \mu_{\mathrm{t},i}\,\Delta_i = -\ln(1-\xi)$$

Regular tracking:

1) Draw a random number $\xi$
2) While LHS < RHS
   move to the next intersection
3) Find the exact location
   in the last segment analytically

Pebble-bed reactor



Images courtesy of Rintala et al. [2015]

Finding the intersections can be expensive…



Regular tracking is often used in simulations of nuclear reactions pebble-bed reactors, where the environment can be modeled as a collection of spherical homogeneous volumes. It is sometimes referred to as surface tracking in neutron transport literature.

The main drawback of regular tracking is to necessity to find all the intersections with interfaces that separate individual homogeneous regions—this can be fairly expensive.

# RAY MARCHING

Find the collision distance approximately

$$\int_0^t \mu_t(s)\mathrm{d}s = -\ln(1-\xi)$$

$$\not\Downarrow$$

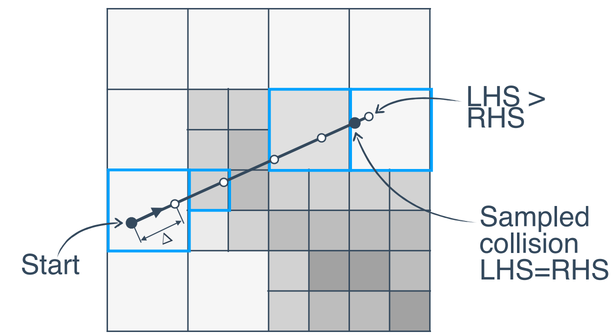
$$\sum_{i=1}^{k} \mu_{t,i}\,\Delta = -\ln(1-\xi)$$

Constant step

Ray marching:

1) Draw a random number $\xi$
2) While LHS < RHS
   make a (fixed-size) step
3) Find the exact location
   in the last segment analytically

(Hierarchical) voxel grid






Sometimes, we can justify sampling free paths only approximately, in favor of higher speed. This bring me to the next algorithm, called ray marching.

The idea of ray marching is to ignore the interfaces, and march along the ray with a constant step size. The sum on the LHS thus no longer corresponds exactly to the integral, but it is easy to implement and avoids the need to find the interfaces.

Going back to the voxel-grid example, we march along the ray touching only some of the voxels and assume the extinction is constant (or polynomial) along each step. Once the sum exceeds the RHS, we again retract back finding the location within the last segment analytically.

# RAY MARCHING

Find the collision distance approximately

$$\int_0^t \mu_t(s)\mathrm{d}s = -\ln(1-\xi)$$

$$\neq$$

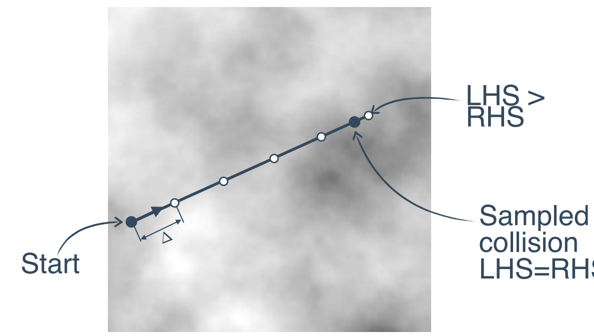
$$\sum_{i=1}^{k} \mu_{t,i}\,\Delta \;= -\ln(1-\xi)$$

Constant step

Ray marching:

1) Draw a random number $\xi$
2) While LHS < RHS
   make a (fixed-size) step
3) Find the exact location
   in the last segment analytically

General volume






As long as approximate free paths are acceptable, ray marching can be used with almost any kind of medium.

# RAY MARCHING

Find the collision distance approximately

$$\int_0^t \mu_t(s)\mathrm{d}s = -\ln(1-\xi)$$

$$\not\parallel$$

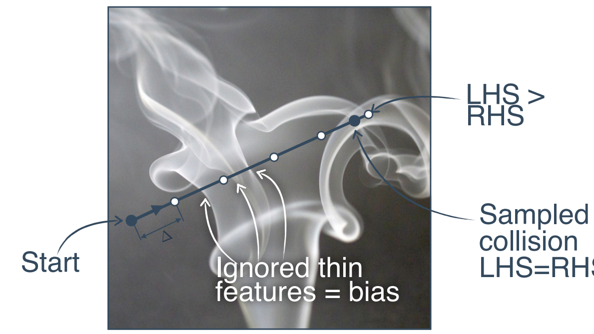
$$\sum_{i=1}^{k} \mu_{t,i}\,\Delta \;\; = -\ln(1-\xi)$$

Constant step

Ray marching:

1) Draw a random number $\xi$
2) While LHS < RHS
   make a (fixed-size) step
3) Find the exact location
   in the last segment analytically

General volume






Ignoring the variation of the extinction function between the steps, however, biases the distribution of free paths—it is no longer a true free-path distribution.

# FREE-PATH SAMPLING

| ANALYTIC CDF$^{-1}$ | REGULAR TRACKING | RAY MARCHING |
|---|---|---|
| ‣ Efficient & simple, limited to few volumes | ‣ Iterative, inefficient if free paths cross many boundaries | ‣ Iterative, inaccurate (or inefficient) for media with high frequencies |
| ‣ Simple volumes (e.g. homogeneous) | ‣ Piecewise-simple volumes | ‣ Any volume |
| ‣ Unbiased | ‣ Unbiased | ‣ Biased |

Common approach: sample optical thickness, find corresponding distance



Let's quickly summarize all the analog methods that we discussed until now.

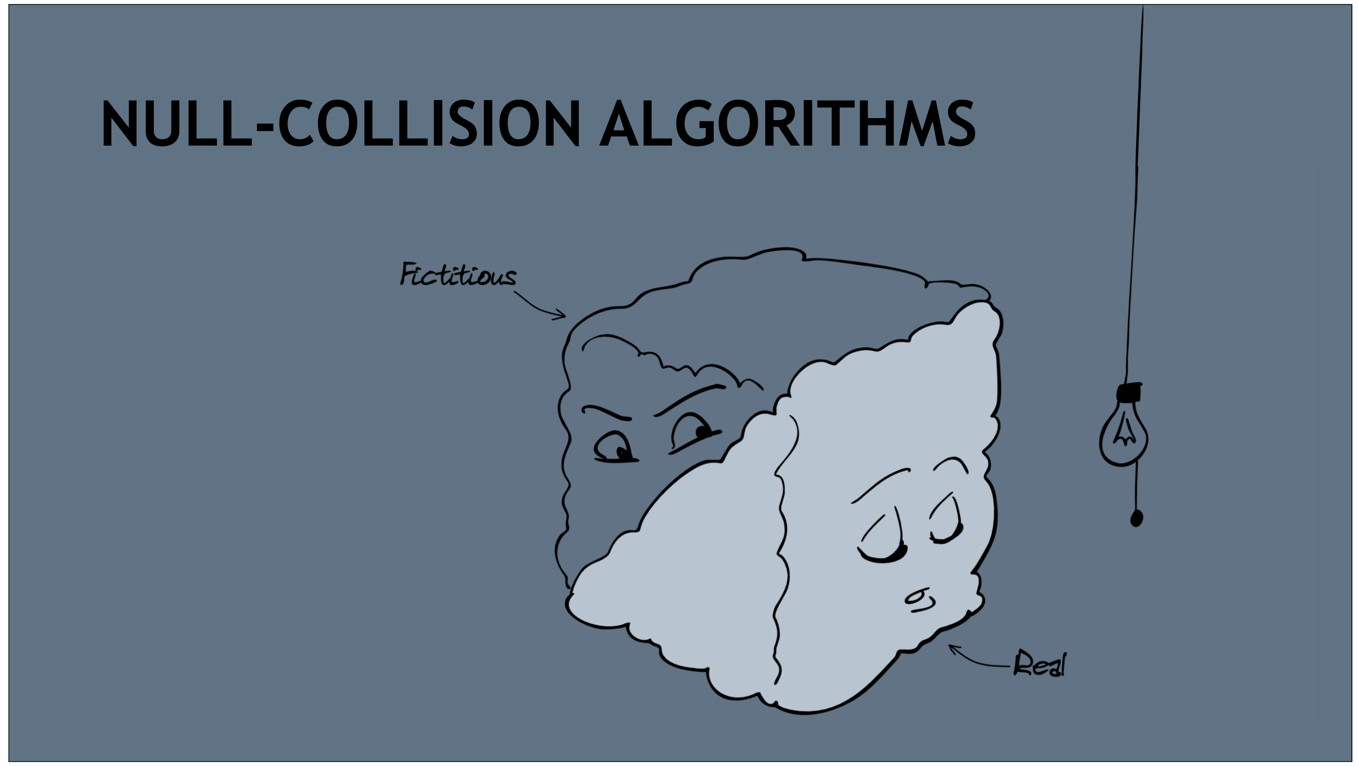
Analytic inversion sampling is certainly preferred, but can be achieved only for very few, rather simple volumes.

If the volume is piecewise-simple, we can iterate through individual volumes along the ray and still find the free-path sample in an unbiased way, but at the cost of finding all the intersections along the ray.

Finally, ray marching avoids this cost by stepping with a constant stride, but introduces bias.

All these methods approach distance sampling in the same manner: they choose a random value of optical thickness and then sweep along the ray searching for the corresponding location.

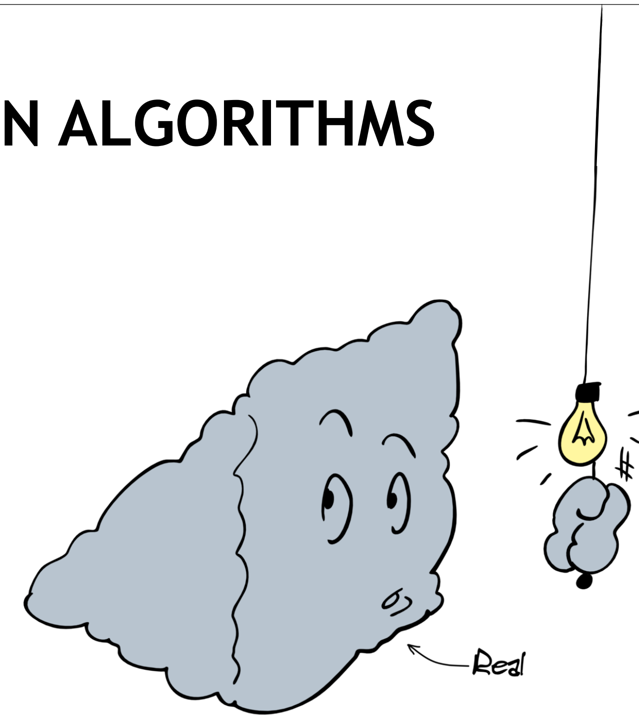
There is a very different approach, which allows sampling free paths in arbitrary volumes without introducing bias.




The approach is based on so-called null collisions.

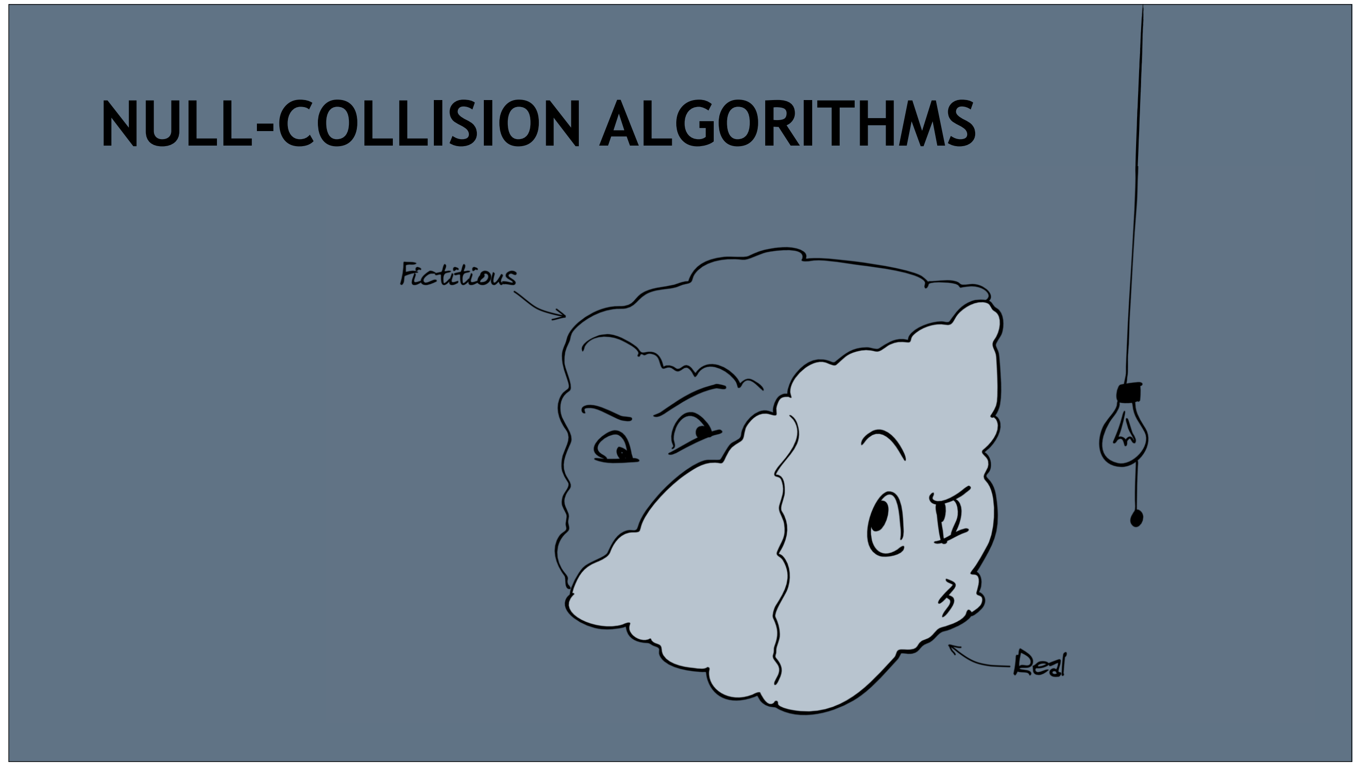
The idea of null-collision algorithms is to add a fictitious material, which is completely transparent to light, but its presence enables closed-form sampling by homogenizing the extinction function.

# NULL-COLLISION ALGORITHMS




The approach is based on so-called null collisions.

The idea of null-collision algorithms is to add a fictitious material, which is completely transparent to light, but its presence enables closed-form sampling by homogenizing the extinction function.




The approach is based on so-called null collisions.

The idea of null-collision algorithms is to add a fictitious material, which is completely transparent to light, but its presence enables closed-form sampling by homogenizing the extinction function.

# NULL-COLLISION ALGORITHMS

Origins in neutron transport and plasma physics, unbiased sampling

Applied in rendering since 2008 [Raab et al. 2008]

**FREE-PATH** sampling:

- **Delta tracking** (a.k.a Woodcock tracking)
- **Weighted delta tracking**
- **Decomposition tracking**
- Spectral tracking

**TRANSMITTANCE** estimation:

- Delta tracking
- (Residual) ratio tracking
- Next-flight delta/ratio tracking

Discussed together w/
other transmittance estimators



Null-collision methods were developed in neutron transport and plasma physics. They've been in use in computer graphics since 2008 and made it gradually all the way into production at some studios.

We will now discuss derivates of these algorithms for free-path sampling and detail the variants developed for estimating transmittance later.

## DELTA TRACKING

## WEIGHTED (DELTA) TRACKING

## DECOMPOSITION TRACKING

# DELTA TRACKING

a.k.a. Woodcock tracking, pseudo scattering, hole tracking, null-collision method,…

**PHYSICALLY-BASED** interpretation

- Correctness motivated by intuitive physical arguments: Butcher and Messel [1958, 1960], Zerby et al. [1961], Bertini [1963], Woodcock et al. [1965], Skullerud [1968], …

**MATHEMATICAL** formalisms

- Proofs: Miller [1967], Coleman [1968]
- Integral formulation: Galtier et al. [2013]



Delta tracking (also known as Woodcock tracking, pseudo scattering, hole tracking) is typically explained in papers using a set of physically-motivated arguments, where we reason about the presence of a fictitious matter and how light interacts with it.

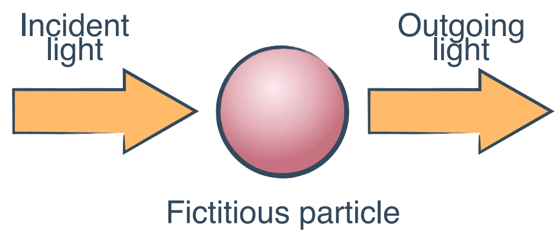
However, there is also a mathematical formalization of the method, which allows us to drop these physical arguments, and reason about the correctness from a purely mathematical standpoint.
The mathematical approach provides us with a very convenient framework for postulating new variants of null-collision algorithms.

We will look at the physically-based interpretation first, and then discuss the mathematical formalism.

# PHYSICAL INTERPRETATION

Add **FICTITIOUS MATTER** to homogenize heterogeneous extinction

- albedo $\alpha(\mathbf{x}) = 1$
- phase function $f_{\mathrm{p}}(\omega, \bar{\omega}) = \delta(\omega - \bar{\omega})$




Presence of fictitious matter does not impact light transport

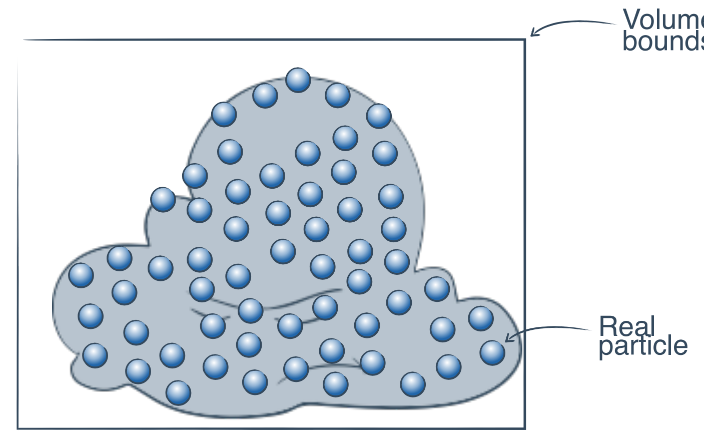


The idea of delta tracking is to add fictitious material, which homogenizes the volume and allows closed-form sampling of distances.

The fictitious material has albedo = 1, and its phase function is a delta function. A fictitious particle thus generates a so-called null-collision, after which all light continues forward, as if there was no collision.

# PHYSICAL INTERPRETATION

## HOMOGENIZATION

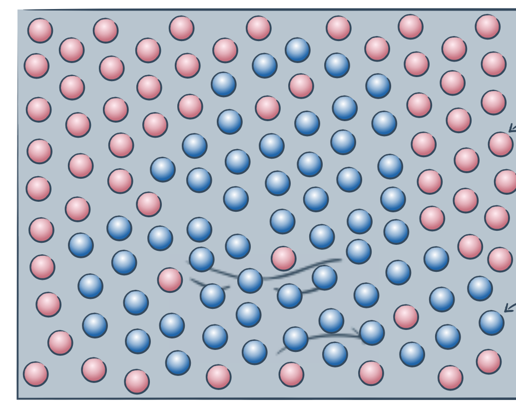





Here is a quick illustration of how the algorithm works. We take a heterogeneous volume, bound it by some geometry (in this example we use a box)…

# PHYSICAL INTERPRETATION

## HOMOGENIZATION



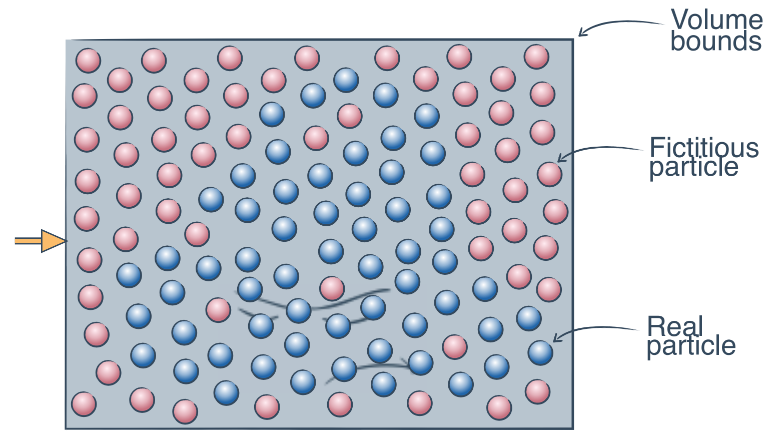





…and fill up the box with fictitious particles, so that the combined density of real and fictitious particles is the same everywhere.

# PHYSICAL INTERPRETATION

## HOMOGENIZATION

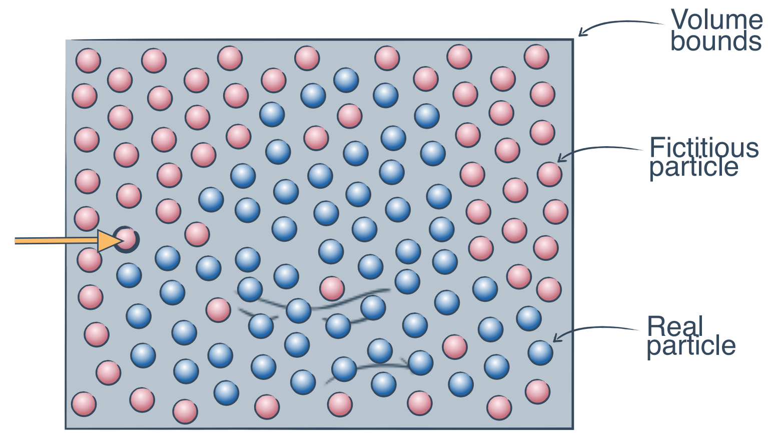





When a beam of light enters the box, it will hit some of the particles.

# PHYSICAL INTERPRETATION

## HOMOGENIZATION

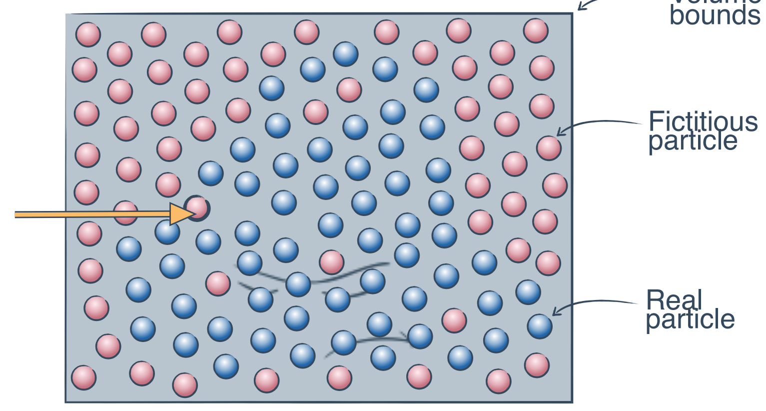





If the particle is fictitious, it produces a null collision upon which the light continues forward.

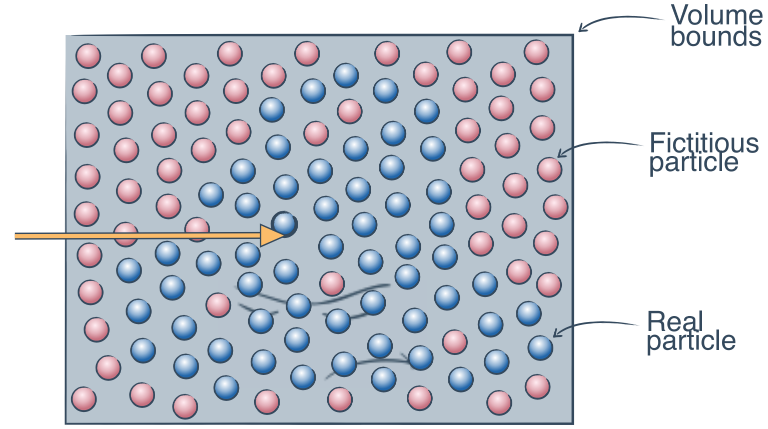
# PHYSICAL INTERPRETATION

## HOMOGENIZATION



# PHYSICAL INTERPRETATION

## HOMOGENIZATION



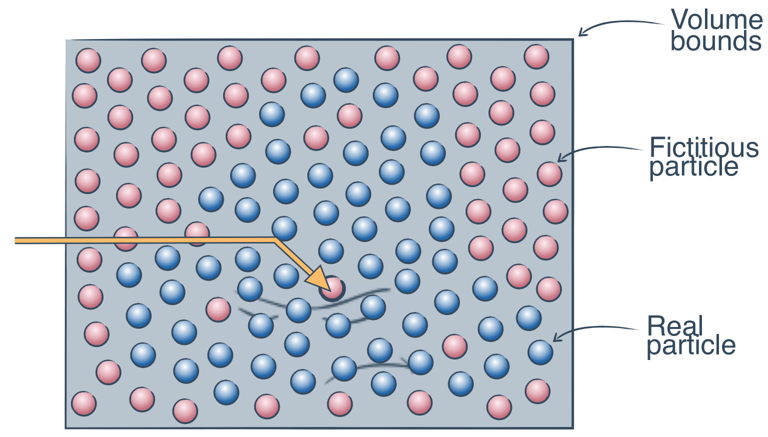





A real particle will absorb or scatter the light, as we see here.

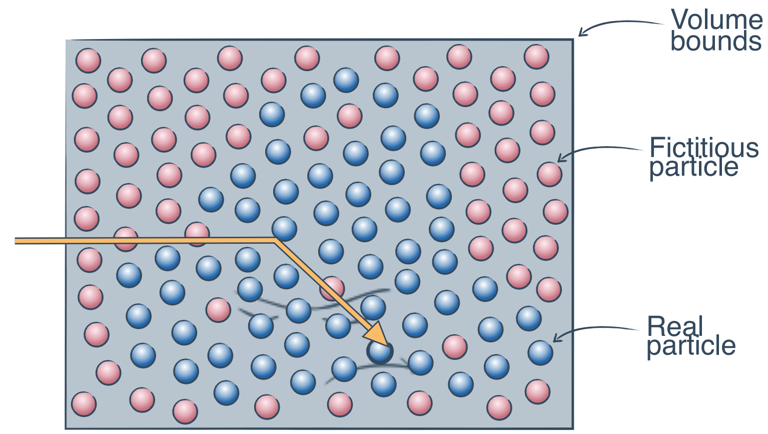
# PHYSICAL INTERPRETATION

## HOMOGENIZATION



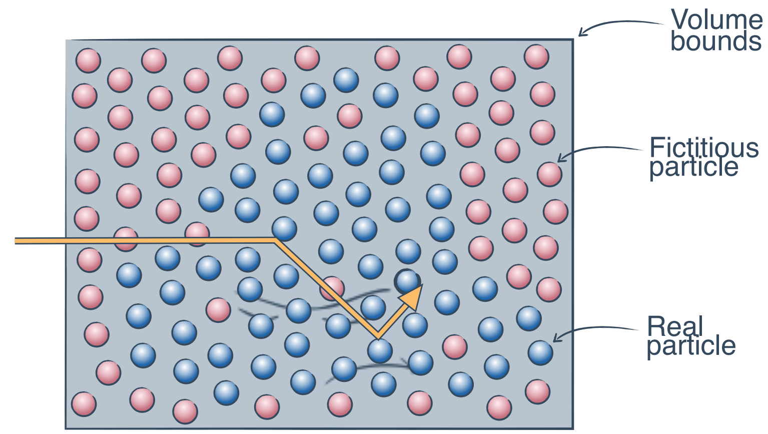
# PHYSICAL INTERPRETATION

## HOMOGENIZATION



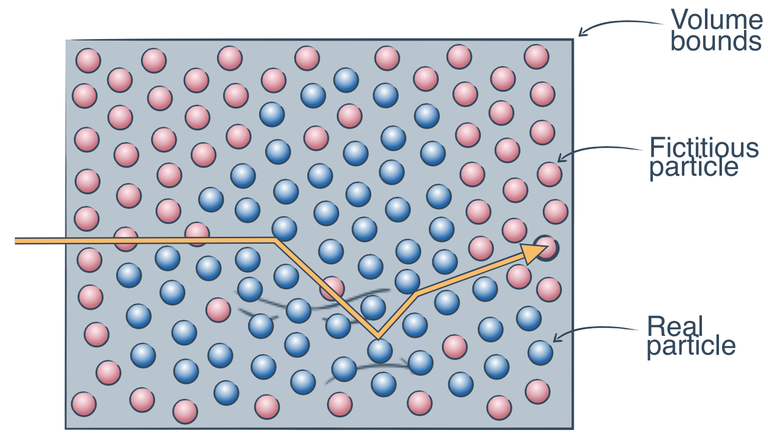
# PHYSICAL INTERPRETATION

## HOMOGENIZATION



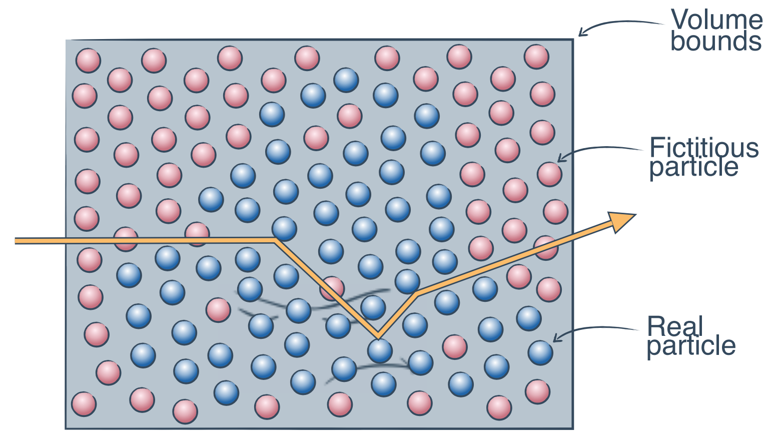
# PHYSICAL INTERPRETATION

## HOMOGENIZATION



# PHYSICAL INTERPRETATION

## HOMOGENIZATION

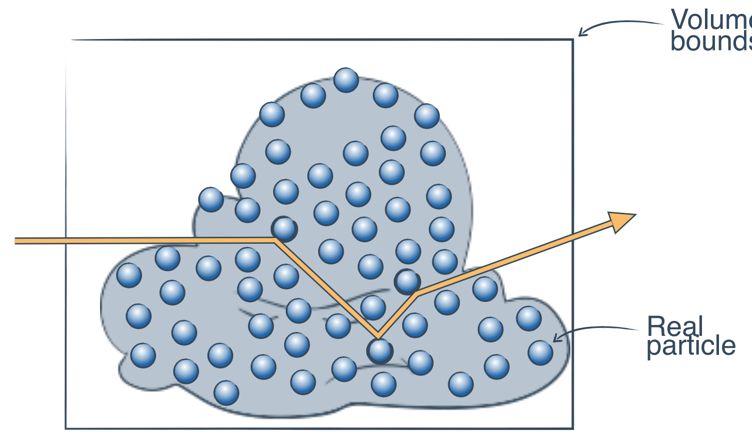





Notice that we have constructed a path through the medium, which, if we remove the fictitious particles…

# PHYSICAL INTERPRETATION

## HOMOGENIZATION

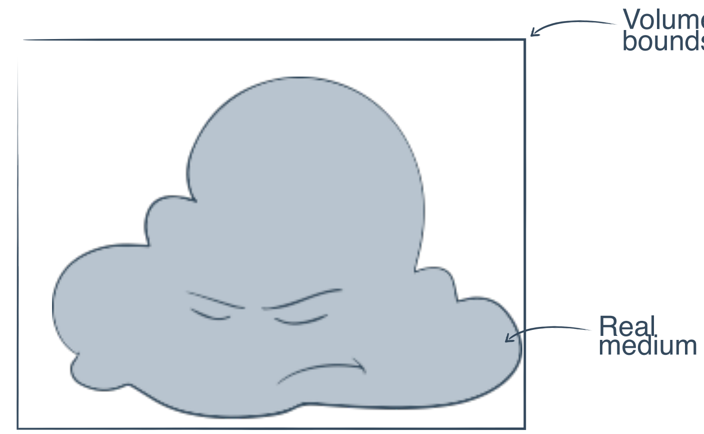





...appears to have interacted only with the real ones.

The fictitious particles did not anyhow alter the light path, but their presence is extremely convenient as they allow sampling distances to collisions analytically.

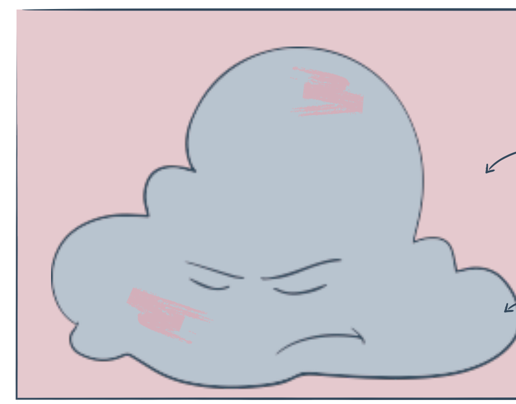
# STOCHASTIC SAMPLING






The particles here were only for illustration. In practice, the collisions are sampled stochastically on top of a statistical model of the medium.

# STOCHASTIC SAMPLING

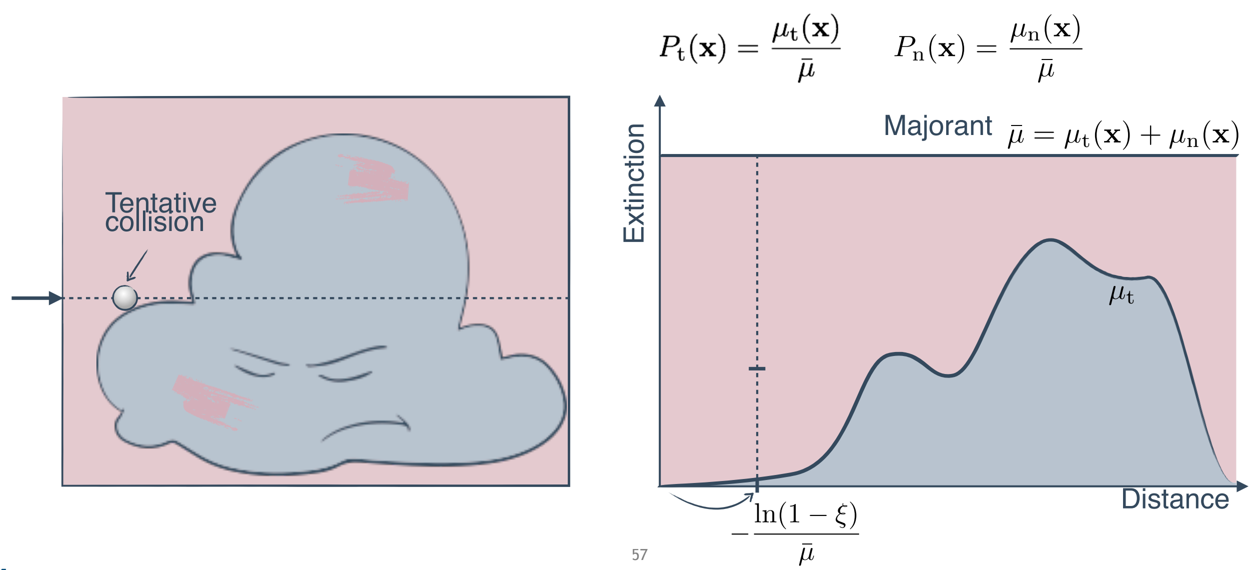





In addition to the extinction coefficient of the real medium, we will also have a null-collision coefficient mu_n representing the fictitious medium. We will adjust the null-collision coefficient spatially, so that the sum of the two coefficients is constant everywhere in the medium.

We refer to this sum as the “majorant of the extinction function”, or simply the “majorant”.

## STOCHASTIC SAMPLING





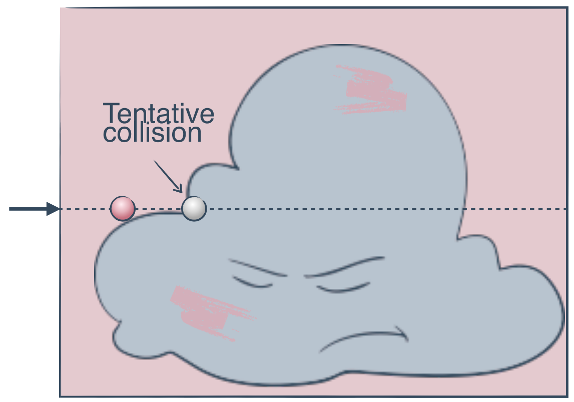

In order to sample the free path, we use the analytic inverse CDF derived from the constant majorant to obtain a distance to the first (tentative) collision.

Next, we need to decide whether the collision is real or null. This is done stochastically using the following probabilities. The probability of a real collision is proportional to the extinction coefficient, analogously, the probability of a null collision is proportional to the null-collision coefficient.

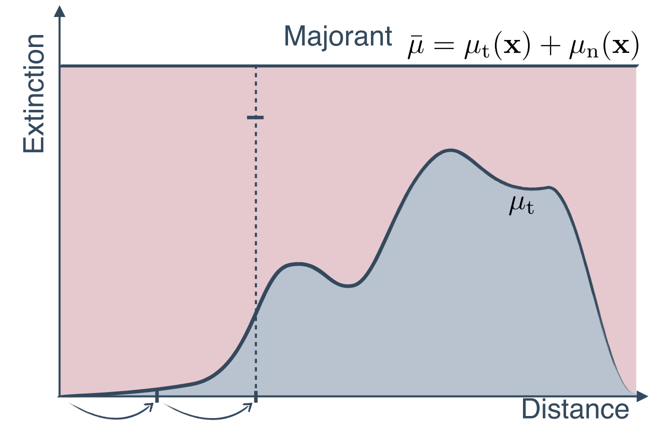
The probabilistic classification can be visualized as picking a random height along the dashed line. Here, the random number selected a null collision.

We repeat the distance sampling and probabilistic classification, until a real collision is found.

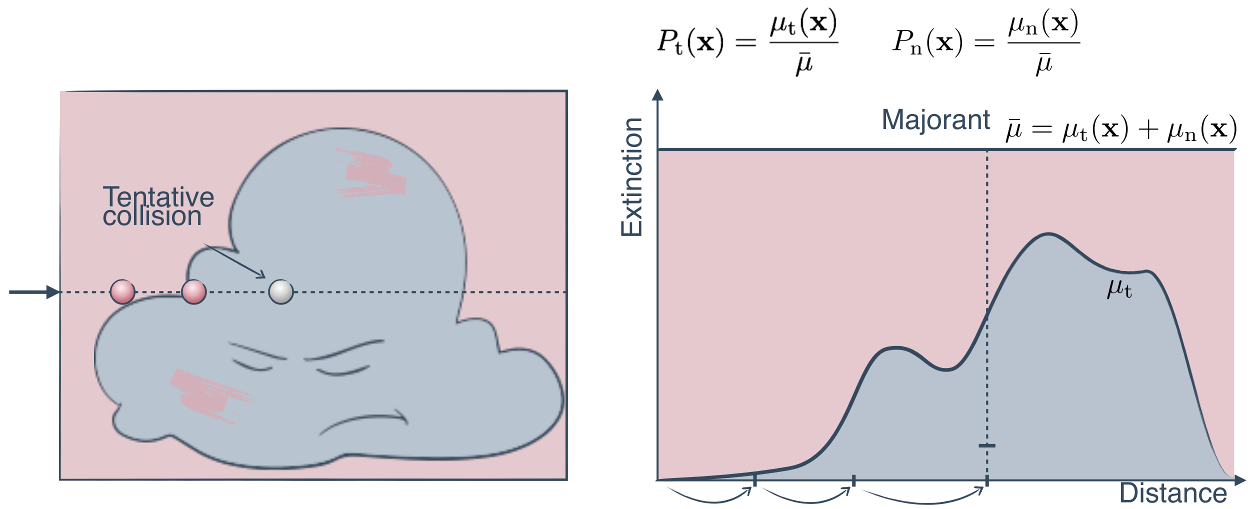
# STOCHASTIC SAMPLING




$$P_{\mathrm{t}}(\mathbf{x}) = \frac{\mu_{\mathrm{t}}(\mathbf{x})}{\bar{\mu}} \qquad P_{\mathrm{n}}(\mathbf{x}) = \frac{\mu_{\mathrm{n}}(\mathbf{x})}{\bar{\mu}}$$

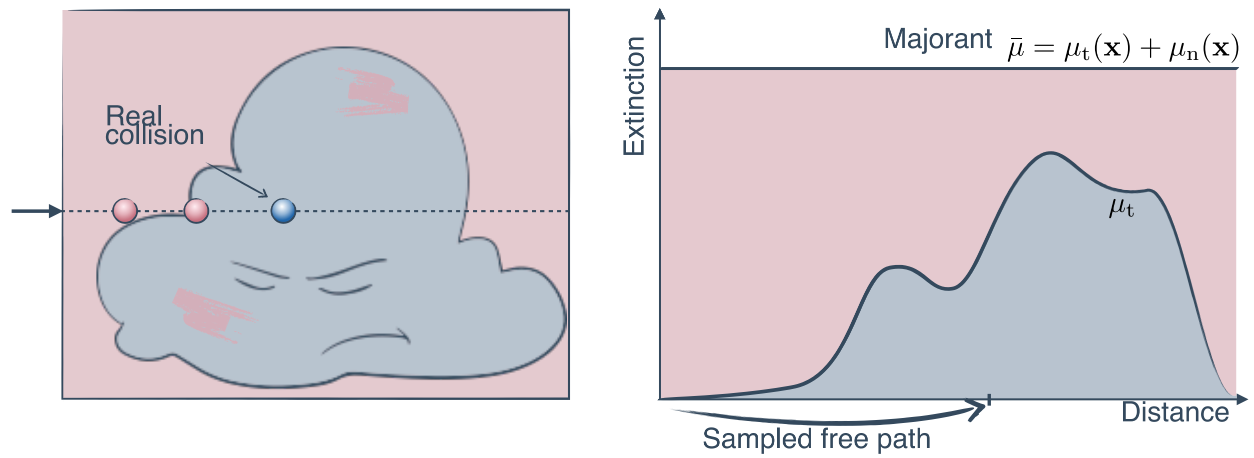


# STOCHASTIC SAMPLING



# STOCHASTIC SAMPLING

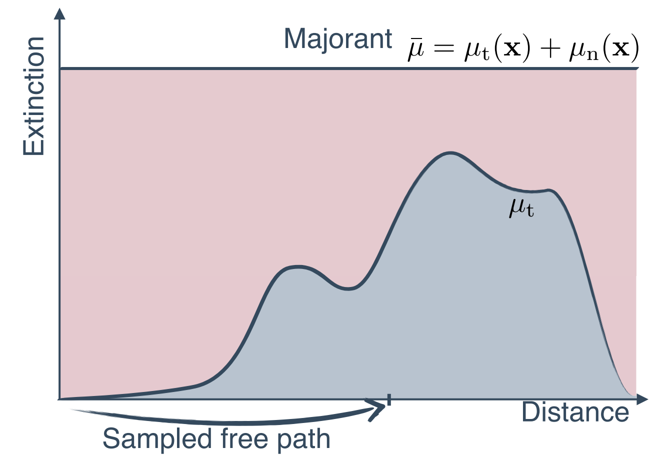
$$P_t(\mathbf{x}) = \frac{\mu_t(\mathbf{x})}{\bar{\mu}} \qquad P_n(\mathbf{x}) = \frac{\mu_n(\mathbf{x})}{\bar{\mu}}$$






The distance to the first real collision represents the free-path sample. This is how delta tracking works: we move forward sampling distances in the combined medium and probabilistically classify each tentative collision is either real or null. The algorithm can be interpreted as a form of rejection sampling.

# IMPACT OF MAJORANT



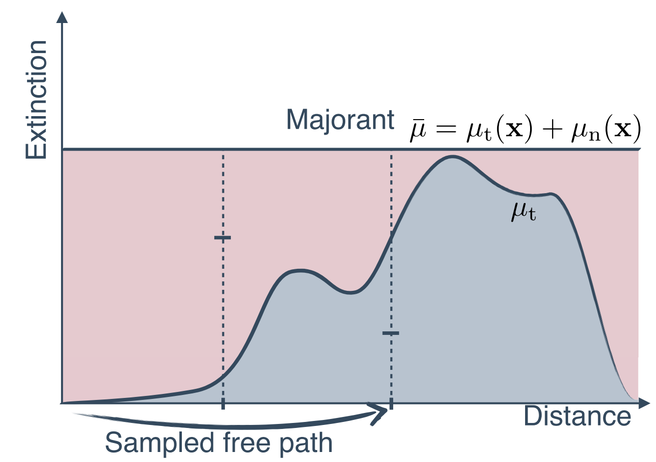





Let’s quickly look how the value of the majorant impacts the algorithm.

# IMPACT OF MAJORANT

Tight majorant = GOOD
(few rejected collisions)

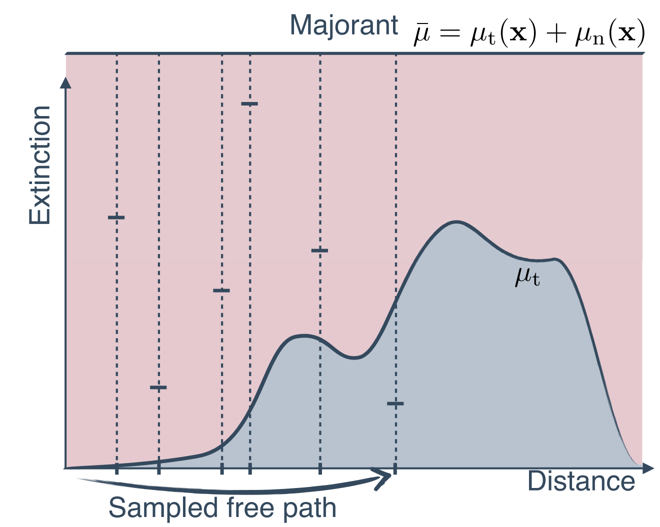





If the majorant is tight, there will be only very few null collisions… this is good.

# IMPACT OF MAJORANT

Loose majorant = BAD
(many expensive rejected collisions)






If the majorant is loose, there will be many null collisions, many rejections, and the free-path sample becomes expensive to construct. The majorant only impacts the efficiency though, the distribution of free paths is always correct with bounding majorants.

# DELTA TRACKING

**PHYSICALLY-BASED** interpretation

- Correctness motivated by intuitive arguments:
  Butcher and Messel [1958, 1960],
  Zerby et al. [1961], Bertini [1963],
  Woodcock et al. [1965], Skullerud [1968],
  …

**MATHEMATICAL** formalism

- Integral formulation: Galtier et al. [2013]



This was the physical interpretation of delta tracking: we add some matter, the photon interacts either with the original, or the new matter, but since the net impact of the new matter on light transport is null, we still obtain the correct free-path distribution. The physically-based interpretation can be somewhat limiting though.

We will now look at the algorithm from a rather mathematical perspective, which is inspired by the integral formulation of null-collision algorithms by Galtier and colleagues.

DELTA TRACKING

WEIGHTED (DELTA) TRACKING

DECOMPOSITION TRACKING

The mathematical formalization will essentially provide us with a family of weighted trackers, that can be more efficient and robust than delta tracking.

# MATHEMATICAL FORMALIZATION

**CHANGE OF RADIANCE** due to null collisions

$$-\mu_{\mathrm{n}}(\mathbf{x})L(\mathbf{x},\omega) + \mu_{\mathrm{n}}(\mathbf{x})\int_{\mathcal{S}^2} \delta(\omega - \bar{\omega})L(\mathbf{x},\bar{\omega})\,\mathrm{d}\bar{\omega} = 0$$

Losses — Gains ("in-scattering")

Cancel each other



based on Galtier et al. [2013]

Let's start by formalizing the change of radiance due to null collisions. We said that the losses and the gains due to the null-collisions perfectly cancel out—they have to add up to zero not to impact light transport.

The differential equation formalizing the null collisions can be added to the RTE, which then changes in two places.

# MATHEMATICAL FORMALIZATION

**CHANGE OF RADIANCE** due to null collisions

$$-\mu_{\mathrm{n}}(\mathbf{x})L(\mathbf{x},\omega) + \mu_{\mathrm{n}}(\mathbf{x})\int_{\mathcal{S}^2}\delta(\omega-\bar{\omega})L(\mathbf{x},\bar{\omega})\,\mathrm{d}\bar{\omega} = 0$$

**INTEGRAL RTE** with null collisions

$$L(\mathbf{x},\omega) = \int_0^\infty T_{\mathrm{\bar{\mu}}}(\mathbf{y})\left[\mu_a(\mathbf{y})L_e(\mathbf{y},\omega) + \mu_s(\mathbf{y})L_s(\mathbf{y},\omega) + \underbrace{\mu_n(\mathbf{y})L(\mathbf{y},\omega)}_{\text{Null-collided radiance}}\right]\mathrm{d}y$$

Transmittance through the combined (real+fictitious) medium

Null-collided radiance



The transmittance here is with respect to the combined medium. It will be a bit lower than in the real medium, but this is compensated by the null-collided radiance, which counteracts the lower transmittance.

# MATHEMATICAL FORMALIZATION

**INTEGRAL RTE** with null collisions

$$L(\mathbf{x},\omega) = \int_0^\infty T_{\bar{\mu}}(y)\Big[\mu_a(\mathbf{y})L_e(\mathbf{y},\omega) + \mu_s(\mathbf{y})L_s(\mathbf{y},\omega) + \mu_n(\mathbf{y})L(\mathbf{y},\omega)\Big]\,\mathrm{d}y$$

## MATHEMATICAL FORMALIZATION

**RTE ESTIMATOR** with null collisions

$$\langle L(\mathbf{x},\omega)\rangle = \frac{T_{\bar{\mu}}(y)}{p(y)} \left[\mu_a(\mathbf{y})L_e(\mathbf{y},\omega) + \mu_s(\mathbf{y})L_s(\mathbf{y},\omega) + \mu_n(\mathbf{y})L(\mathbf{y},\omega)\right]$$

Probabilistic evaluation using Russian roulette

$$\langle f(x)\rangle_P = \begin{cases} \frac{f(x)}{P(x)} & \text{if accepted} \\ 0 & \text{otherwise} \end{cases}$$



based on Galtier et al. [2013]

We now apply a Monte Carlo estimator and we will introduce one more concept here: the concept of probabilistic evaluation of the gains we have in the brackets, such that only one of them is selected for evaluation.

# MATHEMATICAL FORMALIZATION

**RTE ESTIMATOR** with null collisions

$$\langle L(\mathbf{x},\omega)\rangle = \frac{T_{\bar{\mu}}(y)}{p(y)} \left[ \left\langle \mu_a(\mathbf{y}) L_e(\mathbf{y},\omega) \right\rangle_{P_\mathrm{a}} + \left\langle \mu_s(\mathbf{y}) L_s(\mathbf{y},\omega) \right\rangle_{P_\mathrm{s}} + \left\langle \mu_n(\mathbf{y}) L(\mathbf{y},\omega) \right\rangle_{P_\mathrm{n}} \right]$$

Probabilistic evaluation
using Russian roulette

$$\langle f(x)\rangle_P = \begin{cases} \frac{f(x)}{P(x)} & \text{if accepted} \\ 0 & \text{otherwise} \end{cases}$$



This can be done by invoking the Russian roulette on each of the gains.

# MATHEMATICAL FORMALIZATION

**RTE ESTIMATOR** with null collisions

$$\langle L(\mathbf{x},\omega)\rangle = \frac{T_{\bar{\mu}}(y)}{p(y)} \left[ \left\langle \mu_a(\mathbf{y}) L_e(\mathbf{y},\omega) \right\rangle_{P_a} + \left\langle \mu_s(\mathbf{y}) L_s(\mathbf{y},\omega) \right\rangle_{P_s} + \left\langle \mu_n(\mathbf{y}) L(\mathbf{y},\omega) \right\rangle_{P_n} \right]$$

Represents an entire family of
(weighted) trackers that all solve RTE!

Delta tracking is just one specific instance.

…see EG STAR or
**Galtier et al. [2013]**
for complete derivation



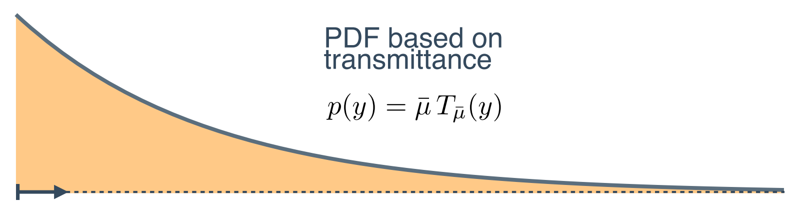
This mathematical formulation represents an *entire family* of weighted trackers that all solve the radiative transfer equation. The formalism here provides us with a convenient framework for developing new (weighted) trackers tailored to specific problems.

# WEIGHTED (DELTA) TRACKING

**RTE ESTIMATOR** with null collisions

$$\langle L(\mathbf{x},\omega)\rangle = \frac{T_{\bar{\mu}}(y)}{p(y)} \left[ \langle \mu_a(\mathbf{y}) L_e(\mathbf{y},\omega)\rangle_{P_a} + \langle \mu_s(\mathbf{y}) L_s(\mathbf{y},\omega)\rangle_{P_s} + \langle \mu_n(\mathbf{y}) L(\mathbf{y},\omega)\rangle_{P_n} \right]$$

Degrees of freedom: **DISTANCE SAMPLING** $p(y)$



PDF based on transmittance

$$p(y) = \bar{\mu}\, T_{\bar{\mu}}(y)$$



*some parameters dropped for brevity

In particular, the estimator exposes two key degrees of freedom.

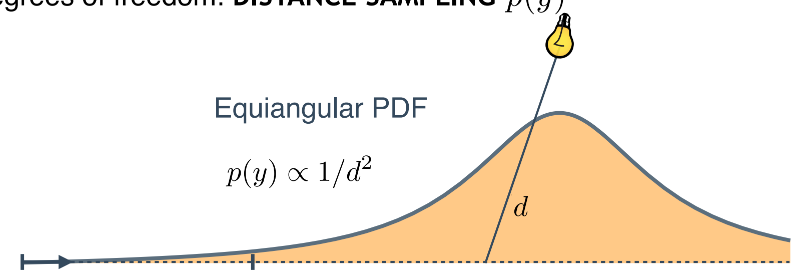
The first one is the distance sampling PDF for generating tentative collisions. Until now, we assumed the PDF to be derived from the homogenized combined medium.

# WEIGHTED (DELTA) TRACKING

**RTE ESTIMATOR** with null collisions

$$\langle L(\mathbf{x},\omega)\rangle = \frac{T_{\bar{\mu}}(y)}{p(y)} \left[\left\langle \mu_a(\mathbf{y}) L_e(\mathbf{y},\omega)\right\rangle_{P_\mathrm{a}} + \left\langle \mu_s(\mathbf{y}) L_s(\mathbf{y},\omega)\right\rangle_{P_\mathrm{s}} + \left\langle \mu_n(\mathbf{y}) L(\mathbf{y},\omega)\right\rangle_{P_\mathrm{n}}\right]$$

Degrees of freedom: **DISTANCE SAMPLING** $p(y)$








*some parameters dropped for brevity

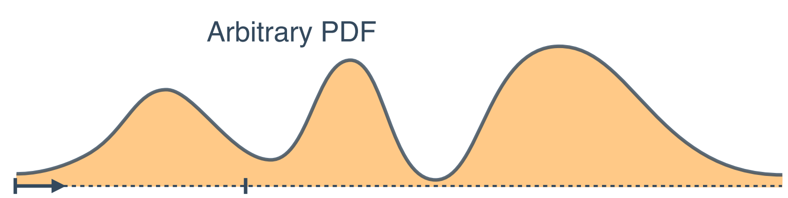
But the PDF can be arbitrary. It can take into account the light source, for instance, or some other knowledge we may have about the scene.

# WEIGHTED (DELTA) TRACKING

**RTE ESTIMATOR** with null collisions

$$\langle L(\mathbf{x},\omega)\rangle = \frac{T_{\bar{\mu}}(y)}{p(y)} \left[ \left\langle \mu_a(\mathbf{y}) L_e(\mathbf{y},\omega) \right\rangle_{P_a} + \left\langle \mu_s(\mathbf{y}) L_s(\mathbf{y},\omega) \right\rangle_{P_s} + \left\langle \mu_n(\mathbf{y}) L(\mathbf{y},\omega) \right\rangle_{P_n} \right]$$

Degrees of freedom: **DISTANCE SAMPLING** $p(y)$





*some parameters dropped for brevity

# WEIGHTED (DELTA) TRACKING

**RTE ESTIMATOR** with null collisions

$$\langle L(\mathbf{x},\omega)\rangle = \frac{T_{\bar{\mu}}(y)}{p(y)} \left[ \langle \mu_a(\mathbf{y}) L_e(\mathbf{y},\omega)\rangle_{P_a} + \langle \mu_s(\mathbf{y}) L_s(\mathbf{y},\omega)\rangle_{P_s} + \langle \mu_n(\mathbf{y}) L(\mathbf{y},\omega)\rangle_{P_n} \right]$$

Degrees of freedom: **DISTANCE SAMPLING** $p(y)$

**INTERACTION PROBABILITIES** $P_a, P_s, P_n$

Delta tracking

$$P_a = \frac{\mu_a}{\bar{\mu}} \qquad P_s = \frac{\mu_s}{\bar{\mu}} \qquad P_n = \frac{\mu_n}{\bar{\mu}}$$

*some parameters dropped for brevity



The second degree of freedom are the interaction probabilities. As long as we do not violate the definition of probability, we can set these arbitrarily.

Delta tracking derives them from collision coefficients using physical arguments. This limits the applicability of delta tracking, for instance we cannot handle negative values of these coefficients. While negative collision coefficient may not sound physically plausible (which should we care then?), they may occur in practice. For instance, when the “majorant” is computed only approximately (e.g. for procedural volumes) it may occasionally underestimate the extinction coefficient. The density of fictitious particles is then negative and the ratios of collision coefficients to the majorant (as defined in delta tracking) are outside of the [0,1] range—the do not represent valid probabilities.

# WEIGHTED (DELTA) TRACKING

**RTE ESTIMATOR** with null collisions

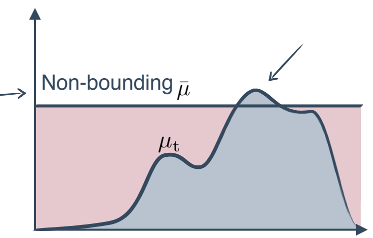
$$\langle L(\mathbf{x},\omega)\rangle = \frac{T_{\bar{\mu}}(y)}{p(y)} \left[ \langle \mu_a(\mathbf{y}) L_e(\mathbf{y},\omega)\rangle_{P_a} + \langle \mu_s(\mathbf{y}) L_s(\mathbf{y},\omega)\rangle_{P_s} + \langle \mu_n(\mathbf{y}) L(\mathbf{y},\omega)\rangle_{P_n} \right]$$

Degrees of freedom: **DISTANCE SAMPLING** $p(y)$

**INTERACTION PROBABILITIES** $P_a$, $P_s$, $P_n$

Weighted tracking that handles

$$P_a = \frac{\mu_a}{\mu_t + |\mu_n|} \quad P_s = \frac{\mu_s}{\mu_t + |\mu_n|} \quad P_n = \frac{|\mu_n|}{\mu_t + |\mu_n|}$$






*some parameters dropped for brevity

We can adjust the probabilities to accommodate for such situations and make the algorithm more robust.

# WEIGHTED (DELTA) TRACKING

**RTE ESTIMATOR** with null collisions

$$\langle L(\mathbf{x},\omega)\rangle = \frac{T_{\bar{\mu}}(y)}{p(y)} \left[ \langle \mu_a(\mathbf{y}) L_e(\mathbf{y},\omega) \rangle_{P_a} + \langle \mu_s(\mathbf{y}) L_s(\mathbf{y},\omega) \rangle_{P_s} + \langle \mu_n(\mathbf{y}) L(\mathbf{y},\omega) \rangle_{P_n} \right]$$

Degrees of freedom: **DISTANCE SAMPLING** $p(y)$

**INTERACTION PROBABILITIES** $P_a, P_s, P_n$

Disabled absorption/emission sampling

$$P_a = 0 \qquad P_s = \frac{\mu_s}{\mu_s + |\mu_n|} \qquad P_n = \frac{|\mu_n|}{\mu_s + |\mu_n|}$$



*some parameters dropped for brevity

Another option is, for instance, to disable absorption/emission sampling if we know there is no emission in the volume.

One can imagine many other schemes of biasing the collision probabilities in order to reduce estimation variance in various situations; this is an active area of research.

# WEIGHTED (DELTA) TRACKING

**RTE ESTIMATOR** with null collisions

$$\langle L(\mathbf{x},\omega)\rangle = \frac{T_{\bar{\mu}}(y)}{p(y)} \left[ \langle \mu_a(\mathbf{y}) L_e(\mathbf{y},\omega) \rangle_{P_a} + \langle \mu_s(\mathbf{y}) L_s(\mathbf{y},\omega) \rangle_{P_s} + \langle \mu_n(\mathbf{y}) L(\mathbf{y},\omega) \rangle_{P_n} \right]$$

**WEIGHT** due to multiple null collisions:

$$\prod_{i=1}^{k-1} \frac{T_{\bar{\mu}}(y_i)}{p(y_i)} \frac{\mu_n(y_i)}{P_n(y_i)}$$




*some parameters dropped for brevity

In all of these situations, the deviations from the analog process are compensated for by a weight at each collision. The weight amounts to the transmittance divided by the PDF, multiplied by the collision coefficient divided by the respective probability.

In the case of multiple null collisions, we then end up with a product of these local collision weights. This product is essentially the path throughput along a chain of null collisions.

# WEIGHTED (DELTA) TRACKING

- Integral framework for null-collision algorithms
  [Galtier et al. 2013]
- Handling of non-bounding “majorants”
  [Cramer 1978, Galtier et al. 2013, Eymet et al. 2013, Novák et al. 2014, Szirmay-Kalos et al. 2017, Kutz et al. 2017, Szirmay-Kalos et al. 2018]
- Improved transmittance estimation
  [Cramer 1978, Novák et al. 2014—Ratio tracking]
- Sample splitting
  [Eymet et al. 2013], [Szirmay-Kalos et al. 2017—Single vs. Double particle model]
- Spectral tracking
  [Kutz et al. 2017]



There are many approaches for setting these probabilities and PDFs. Some for handling non-bounding majorants, others that improve transmittance estimation or balance the evaluation of individual gains. Probably the most useful feature of weighted delta tracking is that it can handle multiple wavelengths at the same time, so we no longer need to execute delta tracking for each color channel independently.

# WEIGHTED (DELTA) TRACKING

## SUMMARY

- Non-analog tracker
- Distance distribution differs from free-path distribution, but…
  distribution of **WEIGHTED** distance samples is **IDENTICAL** to free-path distribution
- Allows handling non-bounding “majorants”
- Enables reducing variance by adjusting:
  - distance sampling of tentative collisions
  - collision probabilities



In summary, weighted tracking is a non-analog tracker, which weights distance samples to yield the correct distribution. In my opinion, there are still many unexplored variants of the algorithm that could be tailored to various problems, and the mathematical formalism that we reviewed is extremely useful—it provides us with a common language for discussing and comparing new algorithms across fields.

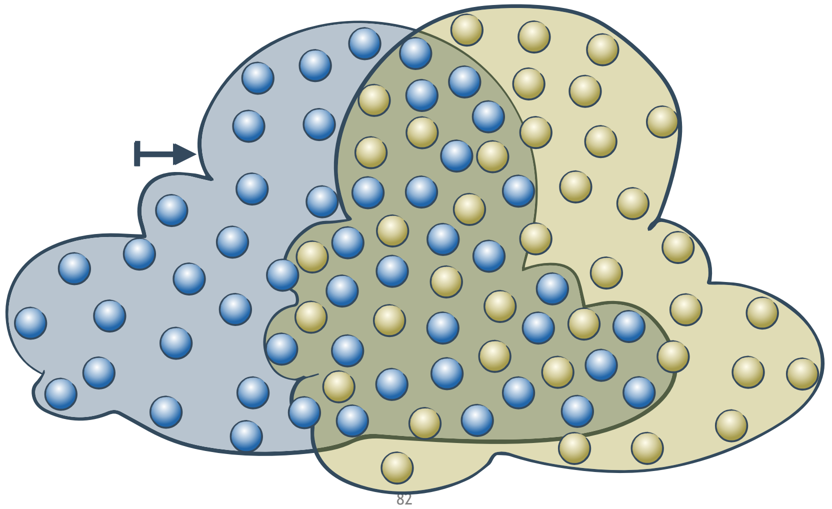
DELTA TRACKING

WEIGHTED (DELTA) TRACKING

**DECOMPOSITION TRACKING**

The last tracker that I will briefly discuss is called decomposition tracking.

# OVERLAPPING VOLUMES

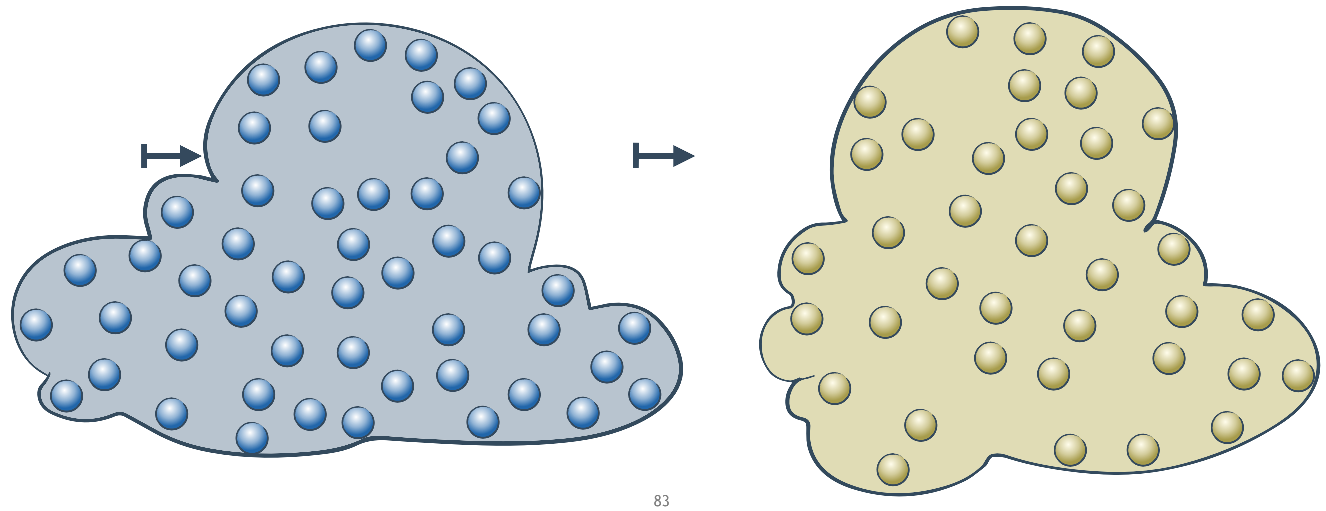






Before we look at the tracker, let's talk about how to sample free paths in overlapping volumes.

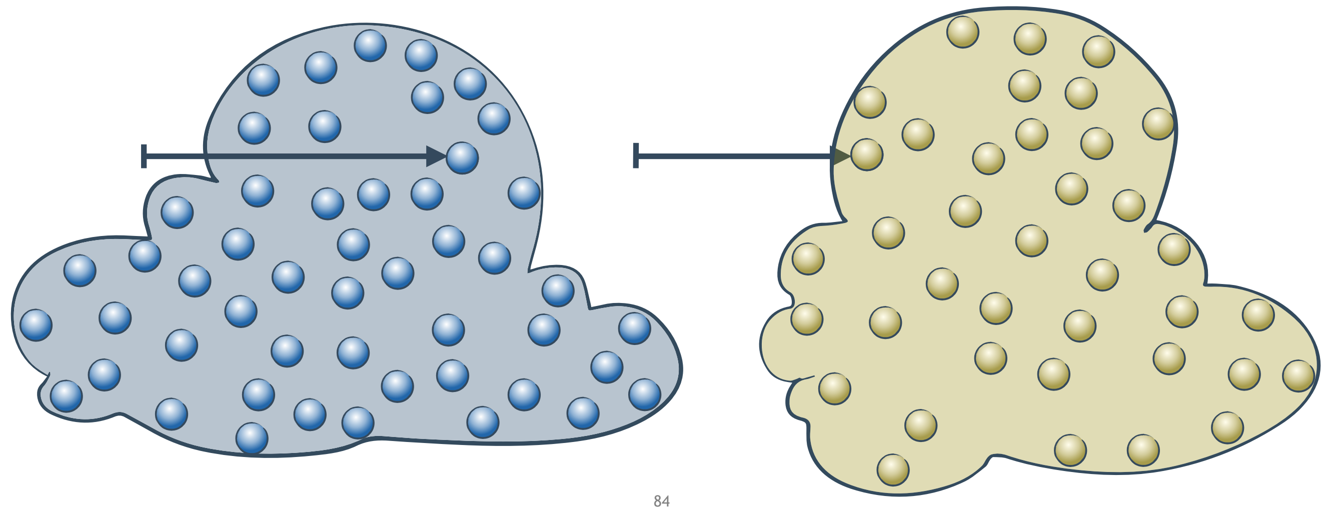
# OVERLAPPING VOLUMES





Let us assume we can only handle each volume in isolation.

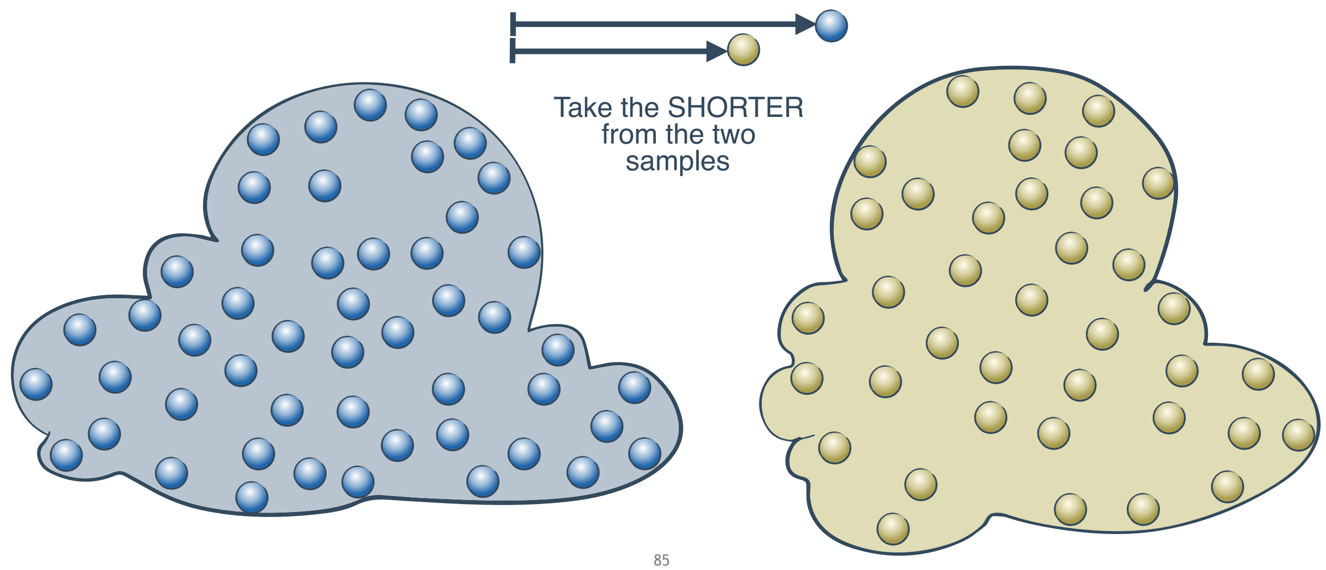
# OVERLAPPING VOLUMES





To sample a free path in the composite, we can sample a free path in each volume independently…

# OVERLAPPING VOLUMES

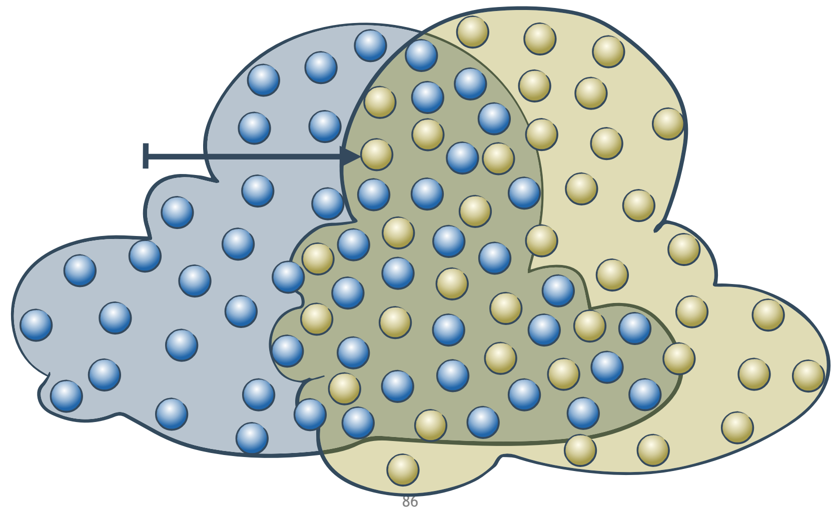







…and then take the shorter of the two samples.

# OVERLAPPING VOLUMES





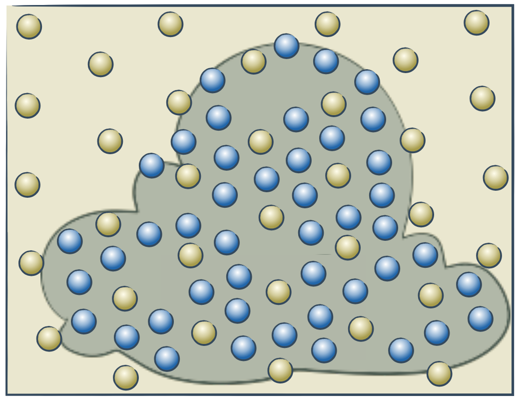


This will become the free path sample in the combined volume.

# DECOMPOSITION TRACKING

Accelerate free-path sampling by reducing expensive extinction evaluations

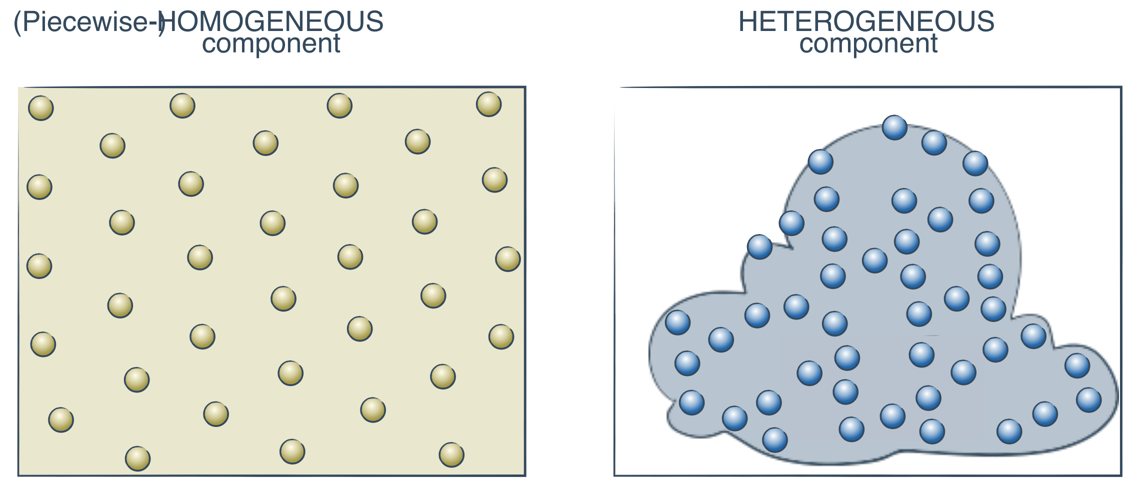
- [Kutz et al. 2017]





Peter Kutz at Disney Animation made the observation that decomposing the volume into overlapping volumes can actually be used to reduce the cost of tracking, specifically, we can lower the number of evaluations of spatially varying coefficients.

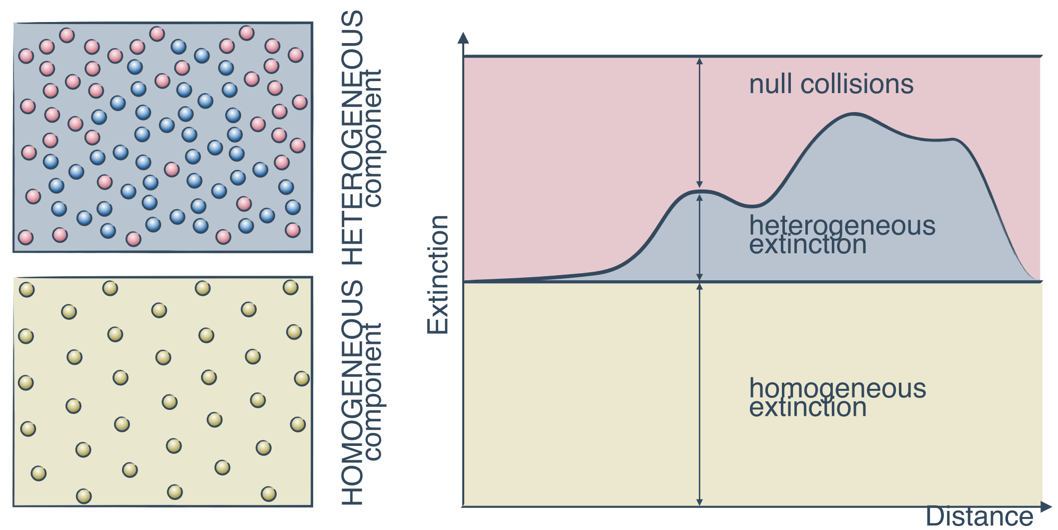
# DECOMPOSITION TRACKING






Let’s assume the volume can be decomposed into a homogeneous part, on the left, and a residual heterogeneous part on the right.

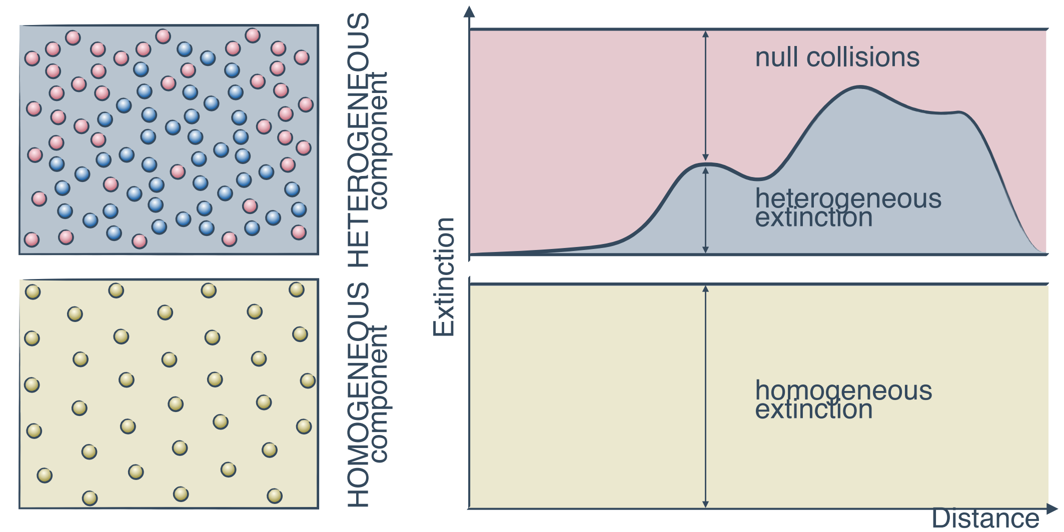
# DECOMPOSITION TRACKING






The extinction function then consists of a homogeneous component, a heterogeneous component, and fictitious matter to enable delta tracking.

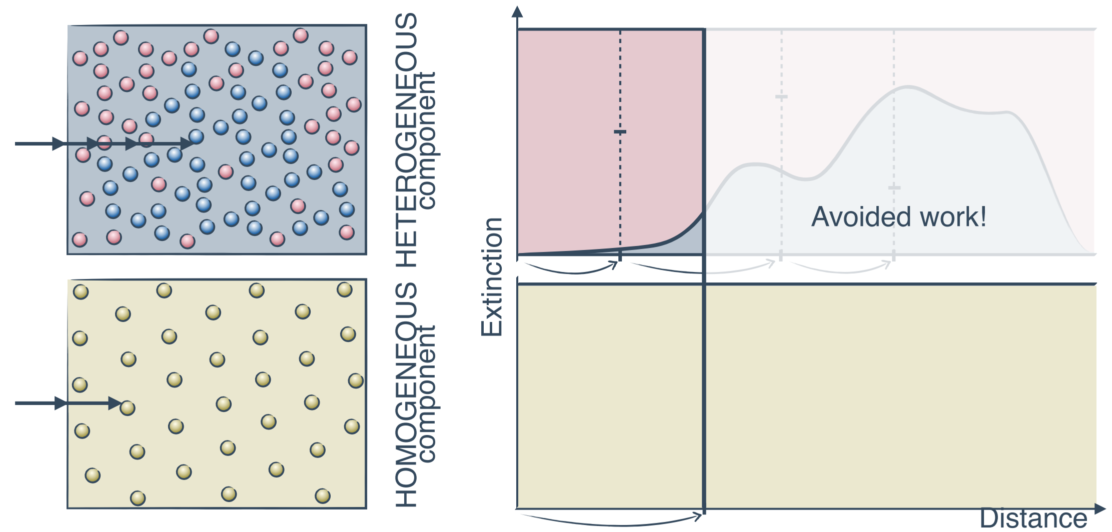
# DECOMPOSITION TRACKING






The idea of decomposition tracking is to sample each component independently and take the shorter of the two samples.

# DECOMPOSITION TRACKING






We start with the homogeneous component, obtaining a distance sample analytically. [CLICKx4] We then use this as the upper bound for delta tracking in the heterogeneous component. If the tracker is about to exceed this bound, then we terminate the tracking without continuing further since we know that the distance sample in the homogeneous volume is shorter and will be used as the free-path sample.

This way we can save a lot of delta tracking steps, which can be fairly expensive.

# DECOMPOSITION TRACKING

Decomposition tracking:

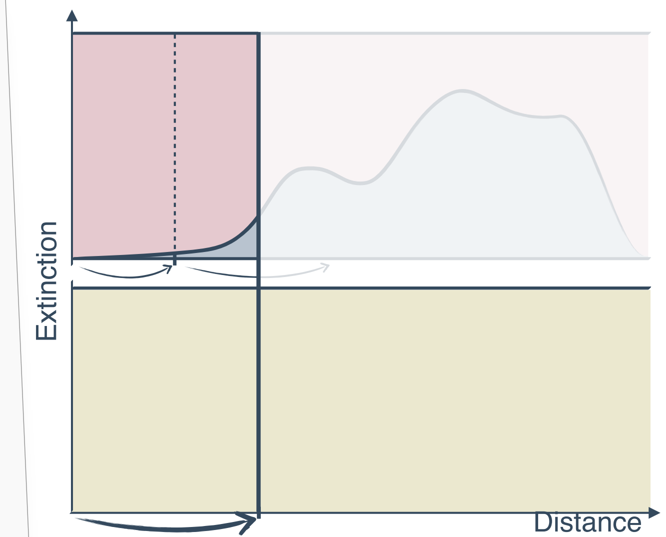
1) Decompose into control and residual
2) Sample control component

Repeat

3) Sample tentative free path in residual component
4) If beyond control sample
    5) Return control sample
6) Probabilistically classify collision

Until collision classified as real

7) Return residual sample



# DECOMPOSITION TRACKING


Kutz et al. [2017]



Here is an example. The cloud was subdivided using an octree where each cell contains a homogeneous and a residual heterogeneous component. Decomposition tracking reduces the number of extinction lookups by 42% yielding the same free-path samples as delta tracking.

# DECOMPOSITION TRACKING

**HOMOGENEOUS** and **RESIDUAL HETEROGENEOUS** components

- Reduces evaluations of spatially varying collision coefficients
- Requires a space-partitioning data structure (e.g. octree) to be practical
- Can be combine with weighted tracking to handle arbitrary decompositions

# MORE DISTANCE SAMPLING…

- Equiangular sampling
  [Kulla and Fajardo 2012]
- Joint-importance sampling
  [Georgiev et al. 2013]
- Tabulation approaches
  [Kulla and Fajardo 2012, Novák et al. 2012, Georgiev et al. 2013, Novák et al. 2014]