

Rendering Participating Media

Barnabás Börcsök¹

Seminar: Data Visualization

barnabas.borcsok@gmail.com

Supervisor: Ludwig Leonard-Méndez¹

¹ Technische Universität München



Figure 1: Examples of participating media.

Abstract

Rendering participating media such as smoke, fog, clouds and fire is an important and active branch of computer graphics research. This essay aims to give an overview of rendering such volumetric phenomena by formalizing the problem statement, and building up to current advancements and directions of research in the field of rendering participating media.

CCS Concepts

•Computing methodologies → Rendering;

1. Introduction

Participating media affect the light as it tries to propagate through its volume. Some of the common examples include glass, water, smoke, and even clear air. (See Figure ??.) We approach the rendering of such phenomena as a collection of particles interacting with light rays. Chapter ?? defines these possible interactions, laying the foundation for Chapter ?? to formalize a possible extension of the widely used surface rendering equation, first introduced by [?].

The main contribution of this paper is an overview of rendering participating media, building up the necessary formulations, and also giving a stable starting point for the interested reader to build up and understand more advanced techniques.

This paper builds for the most parts on the 2017 SIGGRAPH course on *Production Volume Rendering* [?] and on the 2018 survey on *Monte Carlo Methods for Volumetric Light Transport Simulation* [?].

As the use of ray tracing and Monte Carlo rendering is prevalent

in current state-of-the-art production rendering engines [?], we also formalize the problem of rendering participating media as a ray-tracing problem, utilizing stochastic techniques to solve the equations.

The use of Monte Carlo techniques means that we model the interactions between particles and media as a probability field, modeling the "average effect" instead of the individual collision effects.

As photons, making up the light rays, propagate through participating media, the ray's direction and radiance (ultimately perceived as color) changes. This change in the light ray's direction and/or radiance is attributable to collisions with particles assumed to be infinitesimal volumes, that make up the whole of the participating media, as outlined in Section ?. In essence, as these modified light rays end up reaching our eyes (or more precisely, our virtual camera), we end up seeing the phenomena at hand.

We assume *statistically independent* collisions. This means that the methods discussed will work in any gaseous media (e.g. clouds,

smoke, fog, fire, etc), but not for dense granular media such as sand and snow, where these assumptions break down.

The prerequisite for the above assumption is that the size of the particles should be negligible compared to the average distance between them. If the particles are comparable to the average distance between them (which is the case for dense granular media), the assumption of statistically independent collisions breaks down, as a collision event results in a big deviation compared to a case where the event does not happen.

2. Properties of the medium

After a brief introduction to our goal and assumptions, we now dive deeper into how we think about participating media, and define important properties of the materials.

2.1. Collision coefficients

Participating media is a collection of microscopic particles.

As a photon travels through a medium (in the computer graphics sense), it interacts with material particles, which results in a loss of radiance due to the material absorbing the energy of the photon, or scattering it in some different directions. It is also possible for the photon particle to gain radiance due to the material's emission of energy, and also due to collecting the light scattered previously at some other point in the volume.

The *absorption coefficient* σ_a and *scattering coefficient* σ_s each quantify the local probability density of a photon undergoing the respective interaction per unit distance traveled $[\frac{1}{m}]$. For an overview of the possible interactions, see Figure ??.

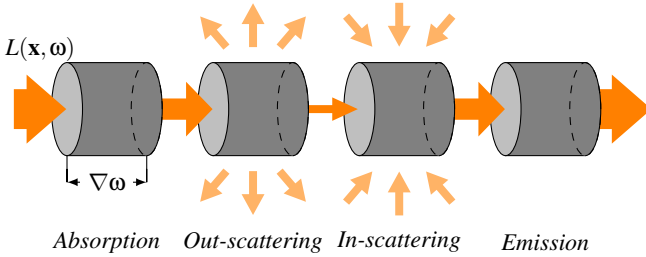


Figure 2: Possible interactions between the volume and the light traveling through the medium. In this example, it first loses $\sigma_a L(x, \omega)$ radiance due to the material absorbing a portion of the light, then loses further $\sigma_s L(x, \omega)$ due to out-scattering. Then it gains $\sigma_s L_i(x, \omega)$ radiance from light scattered at another part of the volume. Lastly, it gains $\sigma_a L_e(x, \omega)$ light due to the material's emission.

The *extinction coefficient* $\sigma_t = \sigma_a + \sigma_s$ indicates the probability density of either type of event happening per unit distance. It is also called the *attenuation coefficient*.

2.2. Phase function

The *phase function* $f_p(\mathbf{x}, \omega, \omega')$ describes how the volume scatters light at a given point \mathbf{x} , depending on the incident (ω) and outgoing (ω') directions.

In order to influence only the direction of the light (but do not influence the intensity of the light), f_p needs to be normalized over the unit sphere: $\int f_p(\mathbf{x}, \omega, \omega') = 1$.

Its use is analog to the BSDF function in the case of surface rendering. In most cases, f_p can be written as a function of the single angle θ between the two directions ω and ω' . If the medium scatters light uniformly in all directions, it is said to be *isotropic*, and the phase function is $f_{p, \text{isotropic}}(\mathbf{x}, \omega, \omega') = 1/(4\pi)$.

If the phase function is not isotropic, then it is *anisotropic*.

2.3. Directional dependence

In cases when the collision coefficients σ_a and σ_s do not depend on the direction of light propagation, the phase function can be parameterized only by the angle between incident and scattered light. A material is then called *isotropic*. (Note: for isotropic materials, the phase function can still be anisotropic – not scattering light uniformly.)

A material is *anisotropic*, if the collision coefficients or the phase functions depend on the direction of incident or scattered light, i.e. the response of the material varies with the direction of propagation.

2.4. Spatial dependence

The medium is *homogeneous* if all of the above medium properties are spatially invariant, and *heterogeneous* otherwise.

3. Mathematical foundations for radiative transport

The physical phenomenon of radiative transport is the transfer of energy in the form of electromagnetic radiation. In our case, we will deal with the transport of light, more precisely the interactions defining a photon traveling through participating media. For an overview of notation used, see Appendix ??.

3.1. Radiative Transfer Equation (RTE)

The Radiative Transfer Equation (RTE) gives us the change in radiance traveling along direction ω through a differential volume element at point \mathbf{x} . Appendix ?? gives a detailed derivation of the RTE function, and its components before arriving at the *integral form of the RTE*, giving us the explicit function

$$L(\mathbf{x}, \omega) = \int_0^\infty e^{-\int_0^s \sigma_t(\mathbf{x}-s\omega)ds} \left[\sigma_s(\mathbf{y}) L_s(\mathbf{y}, \omega) + \sigma_a(\mathbf{y}) L_e(\mathbf{y}, \omega) \right] d\mathbf{y}, \quad (1)$$

which is suitable for use in a path tracing setting. We call $e^{-\int_0^s \sigma_t(\mathbf{x}-s\omega)ds}$ the *transmittance coefficient* and denote it with $T(\mathbf{x}, \mathbf{y})$, summing up the loss of light intensity between \mathbf{x} and \mathbf{y} due to absorption and out-scattering by integrating a single differential process along ω . The exponent $\int_0^s \sigma_t(\mathbf{x}-s\omega)ds$ is called the *extinction coefficient* τ . This is the Beer-Lambert law [?], and we will often simplify the notation of the function to take only a single parameter t , denoting the distance between \mathbf{x} and \mathbf{y} : $T(t) = e^{-\tau(t)} = e^{-\int_0^t \mu_t(\mathbf{x}-s\omega)ds}$.

3.2. Volume rendering equation

As scenes are usually not made up only of participating media, but also solid objects with hard boundaries, we have to extend Equation (??) to also accommodate light interactions with object surfaces. The radiative equilibrium at a surface point \mathbf{z} is given by the surface rendering equation [?]:

$$L(\mathbf{z}, \omega) = L_e(\mathbf{z}, \omega) + \int_{S^2} f_r(\mathbf{z}, \omega, \omega') L_i(\mathbf{z}, \omega') |n(\mathbf{z} \cdot \omega')| d\omega', \quad (2)$$

where $L_e(\mathbf{z}, \omega)$ is the radiance emitted by the surface, $f_r(\mathbf{z}, \omega, \omega')$ is the BSDF (Bidirectional Scattering Function), relating the differential outgoing radiance $dL(\mathbf{z}, \omega)$ to the incident radiance $L_i(\mathbf{z}, \omega')$, and $|n(\mathbf{z} \cdot \omega')|$ is a foreshortening term depending on the surface normal $n(\mathbf{z}, \omega')$ at incident point \mathbf{z} and incident ray direction ω' .

We can combine Equations (??) and (??) into the Volume Rendering Equation (VRE):

$$L(\mathbf{x}, \omega) = \int_0^z T(\mathbf{x}, \mathbf{y}) [\sigma_a(\mathbf{y}) L_e(\mathbf{y}, \omega) + \sigma_s(\mathbf{y}) L_s(\mathbf{y}, \omega)] dy + T(\mathbf{x}, \mathbf{z}) L(\mathbf{z}, \omega). \quad (3)$$

Given a closest surface point \mathbf{z} , our volumetric rendering equation holds true up until this point, giving us the boundary condition for truncating the bounds of Equation (??). (As we have to calculate the propagation of the light ray in the media up until this point.) The radiance at this point \mathbf{z} is given by Equation (??) in direction ω . $L(\mathbf{z}, \omega)$ represents the exitant radiance from the surface given by Equation (??), and the fraction of this that actually reaches our point \mathbf{x} is given by the transmittance coefficient $T(\mathbf{x}, \mathbf{y})$.

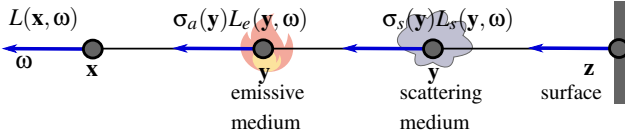


Figure 3: The Volume Rendering Equation (VRE) visualized.

3.3. Monte Carlo integration

Practically all modern high-quality physically-based rendering engines use Monte Carlo integration to solve the aforementioned equations.

A *primary* Monte Carlo estimator $\langle F \rangle = f(x)/p(x)$ is used to approximate $F(x)$'s integrand $f(x)$, where the probability density function (PDF) $p(x)$ is used to sample points x . Averaging N independent realizations of a primary estimator, one can obtain a *secondary* (i.e. multi-sample) estimator.

We can use a Monte Carlo estimator to solve the Volume Rendering Equation (??), estimating the amount of radiance arriving at point \mathbf{x} from direction ω :

$$\langle L(\mathbf{x}, \omega) \rangle = \frac{T(\mathbf{x}, \mathbf{y})}{p(y)} [\sigma_a(\mathbf{y}) L_e(\mathbf{y}, \omega) + \sigma_s(\mathbf{y}) L_s(\mathbf{y}, \omega)] + T(\mathbf{x}, \mathbf{z}) L(\mathbf{z}, \omega), \quad (4)$$

where $p(y)$ is the PDF of sampling point \mathbf{y} , which is y units away from \mathbf{x} . This estimator requires two main routines: one for sampling distances along the ray, and one for estimating the transmittance $T(\mathbf{x}, \mathbf{y})$ between two given points. We will discuss these in Section ?? and ?? respectively.

Estimator ?? gives us a somewhat *localized* view on the light-transport simulation, as we consider only one path segment at a time. For a more *global* view on the transport problem, one can also utilize the path integral framework [?], enabling the sampling of entire sequence of vertices at once, in contrast to (??) sampling only one vertex at once. Examples of such global techniques are joint importance sampling [?] and Metropolis sampling [?]. As we will not discuss these techniques here, the interested reader might want to refer to the cited sources.

Whatever the sampling technique might be, two fundamental building blocks should be considered. Firstly, sampling a distance in a given direction, which is covered in Section ???. This is used, as we construct our ray, propagating from the sensor (or virtual camera) into our virtual scene (and most notably, into the participating media). On the other hand, some techniques do not perform analog walks by sampling distances, but instead rely on estimating the transmittance between two points. We will cover transmittance estimation in Section ??.

4. Distance sampling

In this and the next section, we discuss techniques for sampling distances and estimating transmittance along a ray. To classify distance-sampling methods, we share the terminology of *analog* and *non-analog* estimators of [?], which they borrowed from the field of neutron transport. They categorize the algorithms according to whether they strictly adhere to the physical process of light propagation (analog), or not (non-analog).

Analog methods sample the distance to the next light-medium collision along the line of flight analogously to how photons interact with materials in the real world. The sampling procedure is in such cases commonly referred to as *free-path sampling* or *free-flight-distance sampling*. The distance distribution strictly adheres to the Beer-Lambert law: it has a PDF proportional to the transmittance along the given ray. Sampling can be explicit, via inverting the corresponding cumulative distribution function (CDF), or implicit, through probabilistic reasoning as in null-collision algorithms (discussed in ??).

Non-analog methods have been developed to improve sampling efficiency over analog methods. They deviate from the true distribution of free paths, which are then "corrected" by appropriately weighting the samples. They usually lift some restriction present in analog methods. As we will not concentrate on these non-analog techniques, the interested reader may refer to [?].

Distance sampling is essential in the common case where a path

is constructed incrementally by successively extending it from the sensor to the lights.

Methods share the common theme of sampling distances according to a certain probability density function (PDF). In the following, we first review analytic and semi-analytic analog methods for media that allows free-path sampling in closed form or through a simple iterative process. Next, we discuss rejection-based analog estimators that rely on so-called *null collisions*.

4.1. Closed-form tracking in homogeneous volume

For sampling purposes, we can define a PDF by normalizing the transmittance function $T(t) = e^{-\int_{s=0}^t \sigma_t(\mathbf{x}_s) ds}$.

If the corresponding Cumulative Distribution Function (CDF) is analytically invertible, then the free-path distance t' can be sampled analytically using a single random number ζ .

For the homogeneous volume, where σ_t is not spatially varying, the transmittance becomes

$$T(t) = e^{-\sigma_t t},$$

which is the Beer-Lambert law: exponential extinction of radiance. We would like a free-flight estimator, that produces a free-path distance t' , the distance a photon will travel in the volume. The PDF to do so is defined by normalizing the integral of the exponential function of the transmittance: $p(t) = \sigma_t e^{-\sigma_t t}$.

We can perfectly importance sample the PDF (producing a weight of 1) with the analytic formula for sampling free paths using uniform random numbers $\zeta \in [0, 1)$, [?]: $t' = -\ln(1 - \zeta)/\sigma_t$, with PDF

$$p(t) = \sigma_t e^{-\sigma_t t}. \quad (5)$$

4.2. Regular tracking

In piecewise constant volumes, we can apply closed-form tracking to each of the piecewise constant domains, simply traversing into the next domain if we do not scatter. We can interpret the surface radiance $L(\mathbf{z}, \omega)$ as coming not from a hit surface, but rather a different volume. Finding boundaries separating homogeneous areas can introduce a substantial computing overhead. In order to be efficient, this approach needs large parts of the volume to be homogeneous.

4.2.1. Ray marching

We can reduce the cost of regular tracking, by ignoring the boundaries and marching along the ray with fixed-size steps. This significantly simplifies the implementation at the cost of introducing bias. The algorithm queries the local medium extinction and then moves forward by a fixed distance. We assume either constant or linear optical thickness between the sampling points, deviating from the true free-path distribution. Sampling more frequently is always an option, although this adds to the computational costs. One way to reduce the cost can be to locally adapt the step size, or introduce other more sophisticated methods, such as higher order ray marching schemes [?].

4.3. Delta tracking

Also known as Woodcock tracking, the null-collision algorithm, or pseudo scattering, the main idea behind delta tracking dates back to the rejection sampling technique introduced by John von Neumann in 1951 [?].

The key idea to sampling free-path distances is to homogenize the collision density in heterogeneous volumes by introducing a fictitious collision type. By making the total collision density constant, the volume can be now considered homogeneous. In this new type of collision, called *null-collision*, the volume scatters in the same direction as the incoming direction, having no effect on the light transport itself. We express this collision type with the null-collision coefficient $\sigma_n(\mathbf{x})$ which acts in the same way as the other physical coefficients introduced earlier. The physical collision coefficients are now spatially variant, as is the null-collision coefficient $\sigma_n(\mathbf{x})$. To homogenize the overall volume, we choose $\sigma_n(\mathbf{x})$ in such a way that the sum of all coefficients, the *free-path coefficient* $\bar{\sigma}$, becomes constant:

$$\bar{\sigma} = \sigma_a(\mathbf{x}) + \sigma_s(\mathbf{x}) + \sigma_n(\mathbf{x}) = \sigma_t(\mathbf{x}) + \sigma_n(\mathbf{x}). \quad (6)$$

A consequence of this is that $\bar{\sigma}$ is always equal or greater to the maximum of $\sigma_t(\mathbf{x})$, which is often formulated as being a *majorant* of $\sigma_t(\mathbf{x})$: $\bar{\sigma} \geq \sigma_t(\mathbf{x})$.

We can easily calculate $\sigma_n(\mathbf{x})$:

$$\sigma_n(\mathbf{x}) = \bar{\sigma} - \sigma_t(\mathbf{x}). \quad (7)$$

As $\bar{\sigma}$ is constant, it can take on the role of the constant extinction σ_t used in the closed-form technique ?? and we can draw a distance sample in the same way as in the closed-form tracking:

$$p_n(t) = \bar{\sigma} e^{-\bar{\sigma} t}. \quad (8)$$

By introducing null-collisions, we now have three collision types instead of two, resulting in the definition of three probabilities to consider:

$$P_a(\mathbf{x}) = \frac{\sigma_a(\mathbf{x})}{\bar{\sigma}}, \quad P_s(\mathbf{x}) = \frac{\sigma_s(\mathbf{x})}{\bar{\sigma}}, \quad P_n(\mathbf{x}) = \frac{\sigma_n(\mathbf{x})}{\bar{\sigma}},$$

where $P_a(\mathbf{x}) + P_s(\mathbf{x}) + P_n(\mathbf{x}) = 1$. Applying the additional null-collision probability to the VRE gives the recursive form

$$L(\mathbf{x}_j, \omega_j) = \int_{t=0}^{\infty} p_n(t) \left[P_a(\mathbf{x}) L_e(\mathbf{x}_{j+1}, \omega_j) + P_s(\mathbf{x}) L_s(\mathbf{x}_{j+1}, \omega_j) + P_n(\mathbf{x}) L(\mathbf{x}_{j+1}, \omega_j) \right] dt. \quad (9)$$

To work efficiently, we need $\bar{\sigma}$ to be as close to the maximum of σ_t as possible. While it is valid, to just simply choose a very large $\bar{\sigma}$, that would result in having mostly null-collisions, stopping only to do nothing, and continue further.

5. Transmittance estimation

Transmittance $T(\mathbf{x}, \mathbf{y})$ gives us how likely it is for a photon to pass through the volume between points \mathbf{x} and \mathbf{y} without undergoing absorption or out-scattering. This is also called *free-flight estimation*. As mentioned before, the transmittance function can be simplified to taking in a single real value t , denoting the distance between \mathbf{x} and \mathbf{y} .

Transmittance can be calculated by essentially marching along the ray (instead of using the Monte Carlo technique) accumulating the loss of light (e.g. transmittance). This however introduces bias, and also might result in rendering artifacts if the volume is undersampled. In production, a ray marching transmittance with very small step sizes can be used as a reference to validate more sophisticated implementations [?].

5.1. Delta tracking transmittance estimator

We can use Equation (??) as a transmittance estimator in the PDF approach, testing if a collision occurred before or after distance t . This estimator produces a binary estimator: $T(t) = 1$ or $T(t) = 0$. This technique is *unbiased*. [?] The mean of the samples is the transmittance, although the variance is large. This binary estimate might be considered to be too inaccurate to be used efficiently in the PDF approach as it either totally discards or lets through. [?] However, the concept of adding null collisions opens up the possibility for other methods, that we describe in this section.

5.2. Ratio tracking transmittance estimator

Introduced to computer graphics by [?], the goal of ratio tracking methods is the same: estimating the percentage of photons that make it beyond distance t . The main idea is to remove the binary fashion of the estimation mentioned above, and instead weighting the samples by the probability of continuing the walk.

In contrast to the aforementioned binary estimation, ratio tracking allows every single distance sample to reach t , scoring a fractional weight. The tracking *never* terminates before t , and weight the accumulated transmittance at potential collision position \mathbf{c} by $\sigma_n(\mathbf{c})/\bar{\sigma}(\mathbf{c}) = 1 - \sigma_t(\mathbf{c})/\bar{\sigma}(\mathbf{c})$, the "probability" of continuing forward. This can be formulated in the final transmittance estimator

$$T(d) = \prod_{i=1}^K \left(1 - \frac{\sigma_t(\mathbf{x}_i)}{\bar{\sigma}}\right), \quad (10)$$

where K are all of the collisions created before reaching the end of integration d . Like in delta tracking, the free path coefficient $\bar{\sigma}$ needs to be constant and a majorant of $\sigma_t(\mathbf{x}_i)$.

An incremental improvement, called residual ratio tracking is also introduced by [?]. Refer to ?? for more details.

6. Acceleration Data Structures / Optimization

Acceleration data structures are important, as *data access* usually dominates the overall render time.

While ray tracing, an improvement can be achieved by *avoiding empty spaces* before the first interaction with the volumetric effect.

In the case of delta tracking, having a *localized $\bar{\sigma}$ majorant* for spatially diverse transmittance also yields performance gains, as we reduce the number of null collisions, which means we "stop, and continue without doing anything" fewer times. Space partitioning data structures, such as *k-d trees* [?] or *octrees* [?] might be used to achieve this.

7. Remaining challenges and open problems

7.1. Machine Learning

As is the case with many others fields of computer graphics (and beyond), enhancing volumetric rendering with Artificial Intelligence (AI) methods, such as machine learning has huge potentials. The vast cost involved when accessing voxelized data and tracing paths in high-albedo volumes involving lots of scattering (e.g. clouds) make it challenging for Monte Carlo techniques to deliver results at tractable costs. The high expense of importance sampling such high-dimensional spaces can be mitigated by incorporating various aggregators [?] [?] [?], diffusion approximations [?] [?] [?] [?] or deep learning [?], but this always comes at the cost of introducing some kind of bias. AI techniques can also be applied at the level of distance and directional samples for surface rendering [?] [?] [?] [?], or other forms of path guiding could potentially provide significant benefits.

7.2. Joint handling of surfaces and volumes

A final render usually necessitates including both volumes and surfaces in the same scene. Different techniques have been developed to tackle these different tasks, which could even mean the usage of different renderers for surfaces and volumes in the same scenes, potentially leading to problems when combined. A unified scene representation might lend itself better for many use cases.

8. Conclusion

In this seminar paper, we introduced the problem statement of rendering participating media. We formalized the problem in a way that is applicable to usual ray tracing methods utilized in rendering algorithms. We also looked at subtasks needed, namely distance sampling and transmittance estimation. Without going into more advanced techniques, our aim was to lay the foundations for further reading into the topic, and gave pointers to notable research papers.

Appendix A: Notation

Some papers denote the absorption, scattering and extinction coefficients with μ_a , μ_s and μ_e respectively, devising them from the cross-sectional areas $[m^2]$ σ_a , σ_s and σ_t for absorbing, scattering and extinguishing (both absorbing *and* scattering) particles and also the density per unit volume ρ $[\frac{1}{m^3}]$.

They denote the above mentioned collision coefficients with $\mu_{a,s,t} = \sigma_{a,s,t} \cdot \rho$, multiplying the cross-sectional areas by the density per unit volume.

$\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$	positions in 3D space
$s \in \mathbb{R}$	scalar value
$\boldsymbol{\omega} \in \mathbb{R}^3$	direction vector
$\mathbf{y} = \mathbf{x} - s\boldsymbol{\omega}$	\mathbf{y} is s away from \mathbf{x} in direction $-\boldsymbol{\omega}$
$\mathbf{x}_s := \mathbf{x} - s\boldsymbol{\omega}$	optional shorthands for the above
$\mathbf{x}_t := \mathbf{x} - t\boldsymbol{\omega}$	
$y = \mathbf{x} - y\boldsymbol{\omega}$	y denotes the distance of \mathbf{y} from \mathbf{x} .

Table 1: A quick overview of notation used

For the purposes of this seminar paper, we use the notations introduced in Section ??, denoting the absorption, scattering and extinction coefficients with σ_a , σ_s and σ_t respectively.

Appendix B: Radiative Transfer Equation

The Radiative Transfer Equation (RTE) gives the change in radiance traveling along direction $\boldsymbol{\omega}$ through a differential volume element at point \mathbf{x} .

As a photon travels through a medium, it interacts with material particles, which results in a loss of radiance due to absorption, and scattering in different directions (*specified by the phase function f_p*), while also gaining radiance due to the material's emission of energy, and collecting the scattered light from all directions.

We call the radiance loss due to the light scattering in all directions *out-scattering*, and the radiance gain due to collecting the light scattered elsewhere *in-scattering*.

We lose a portion of radiance $L(\mathbf{x}, \boldsymbol{\omega})$ due to absorption and out-scattering: $\sigma_a(\mathbf{x})L(\mathbf{x}, \boldsymbol{\omega})$ and $\sigma_s(\mathbf{x})L(\mathbf{x}, \boldsymbol{\omega})$, respectively. For brevity, we usually combine these into a single extinction term:

$$\sigma_t(\mathbf{x})L(\mathbf{x}, \boldsymbol{\omega}).$$

For calculating the *radiance gains due to in-scattering*, we collect all incident radiance $L_i(\mathbf{x}, \boldsymbol{\omega})$ from all directions on the unit sphere S^2 , and call it *in-scattered radiance*:

$$L_s(\mathbf{x}, \boldsymbol{\omega}) = \int_{S^2} f_p(\boldsymbol{\omega}, \boldsymbol{\omega}') L_i(\mathbf{x}, \boldsymbol{\omega}') d\boldsymbol{\omega}'.$$

After calculating the in-scattered radiance, we multiply it with the σ_s scattering coefficient, as light gets scattered in the direction of the viewer only if there are scattering particles there.

We can also gain radiance due to the volume emitting energy. The volume's emitted radiance is expressed as a radiance field $L_e(\mathbf{x}, \boldsymbol{\omega})$, whose output gets absorbed and scattered just like any other "regular", non-volume light source. If a volume does not emit, $L_e(\mathbf{x}, \boldsymbol{\omega})$ is simply zero.

Adding together these four terms, we arrive at the Radiative Transfer Equation (RTE):

$$(\boldsymbol{\omega} \nabla) L(\mathbf{x}, \boldsymbol{\omega}) = -\sigma_t(\mathbf{x})L(\mathbf{x}, \boldsymbol{\omega}) + \sigma_s(\mathbf{x})L_s(\mathbf{x}, \boldsymbol{\omega}) + \sigma_a(\mathbf{x})L_e(\mathbf{x}, \boldsymbol{\omega}). \quad (11)$$

Note: $(\boldsymbol{\omega} \nabla)$ is a notation for looking at the change in radiance L in direction $\boldsymbol{\omega}$.

Equation (??) gives us the change in radiance in a forward-transport fashion using the gradient $\boldsymbol{\omega} \nabla$, defining what happens to a radiance beam as it travels forward in direction $\boldsymbol{\omega}$, as can be seen on Figure ??. This is useful for many applications, but not in a path tracing setting. Integrating Equation ?? along the direction $\boldsymbol{\omega}$, changing the gradient $(\boldsymbol{\omega} \nabla)$ on the left side into an integral on the right side, we arrive at the integral form of the Radiative Transfer Equation, giving us the explicit function

$$L(\mathbf{x}, \boldsymbol{\omega}) = \int_0^\infty e^{-\int_0^s \sigma_t(\mathbf{x}-s\boldsymbol{\omega}) ds} \left[\sigma_s(\mathbf{y})L_s(\mathbf{y}, \boldsymbol{\omega}) + \sigma_a(\mathbf{y})L_e(\mathbf{y}, \boldsymbol{\omega}) \right] d\mathbf{y}. \quad (12)$$

Appendix C: Residual ratio tracking

Next to ratio tracking, [?] also introduced a more sophisticated technique called residual ratio tracking. It is an incremental improvement, introducing a control extinction coefficient σ_c , which is constant over the volume, and ideally is very similar to the actual extinction coefficient σ_t . With σ_c , the transmittance $T(t)$ is now broken into two parts: a control transmittance $T_c(t)$, which can be written in closed form, and a residual transmittance $T_r(t)$, which must be estimated:

$$\begin{aligned} T_c(t) &= e^{-\sigma_c t} \\ T_r(t) &= e^{-\int_{s=0}^t \sigma_t(\mathbf{x}_s) - \sigma_c ds} \\ T(t) &= T_c(t)T_r(t) \end{aligned} \quad (13)$$

The key improvement of residual ratio tracking over ratio tracking is the use of $\bar{\sigma}_r$, the majorant over the residual extinction coefficient $\sigma_r = \sigma_t - \sigma_c$.