# Shadow Layers for Participating Media

François Desrichard, David Vanderhaeghe, Mathias Paulin

*Volume 40 (2021), Number 0 pp. 1–10* COMPUTER GRAPHICS *forum*

# Shadow Layers for Participating Media

François Desrichard, David Vanderhaeghe, Mathias Paulin

IRIT, Université de Toulouse, CNRS, INPT, UPS, UT1, UT2J, France

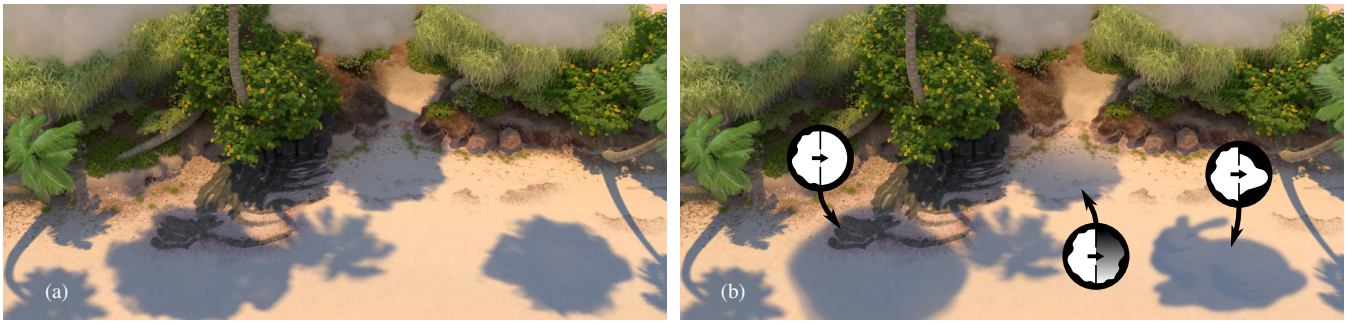



**Figure 1:** *We render separate layers containing shadows in scenes including participating media. (a) For this scene, our global illumination rendering algorithm exports the main image and shadow layers for the three clouds above the beach, all at once. (b) Using standard image space editing tools on these layers, shadows can be simplified, faded, or even entirely repainted.*

**Abstract**

*In the movie industry pipeline, rendering programs output the main image along with a collection of Arbitrary Output Variable layers (AOVs) that retain specific information on light transport and scene properties in image space. Compositing artists use AOVs to improve the quality and appearance of the rendered picture during post-processing, according to the artistic goal of the shot. In particular, cast shadows are manipulated to support narration and storytelling, as the human perception tolerates non-physical edits. Conventional path tracing renderers often propose a shadow matte AOV containing radiance lost when shadow rays are occluded. Previous work has shown that they incorrectly estimate shadow and miss occluded radiance from indirect light sources, and that shadow layers must be used to correctly recover radiance from single, solid occluders. In this paper, we generalise shadow layers to an arbitrary number of occluders, and add support for participating media. We begin by quantifying the radiance loss between the radiative transfer equation and the rendering equation, and translate it into a path integral formulation for efficient Monte Carlo integration. We propose a prototype implementation that renders the main image and shadow layers in a single pass with an affordable computational overhead.*

**CCS Concepts**

• ***Computing methodologies*** → ***Ray tracing; Visibility;*** *Non-photorealistic rendering;*

## 1. Introduction

Lights and shadows play an important role in movies as they convey meaningful information to the viewer at a glance, and often support storytelling [Bir13]. In animated motion picture, artists use computer graphics techniques to carefully setup lighting and shadows and reach the narrative goal expected from the shot. Humans perceive cast shadows as a feature of the object rather than a feature of lighting [SCC18], leaving artists creative freedom for local, non-physical edits. One can for instance strengthen the shadow of important characters or fine-tune the result of global illumination without unsettling the viewer.

In a typical pipeline, light editing takes place both before and after high-quality rendering of a frame: before, artists set up light rigs and materials interactively using low-quality previews; after, they use *Arbitrary Output Variables (AOVs)* – additional layers defined in image space and exported during rendering to assist the transformation of the resulting image, the *main image*. AOVs can contain any useful information such as surface normal, specular lighting, and shadows. During the post-processing stage, they are

assembled by compositing artists to produce the final frame while benefiting from fast previews. Movie production images often involve participating media such as fog and clouds for scenery, and fire and smoke as special effects. While they have a noticeable impact on light transport and shadow formation, few tools exist along the pipeline to artistically direct their appearance. Professional renderers largely rely on path tracing [FHH*19], and offer a *shadow matte* AOV containing the radiance lost among all shadow rays of a path. Because shadow rays are responsible for gathering direct light at an interaction, the shadow matte is inherently limited to direct shadows. Indirect shadows from secondary light sources, such as a photography reflector, are not picked up.

Desrichard *et al.* [DVP19] propose a definition of *shadow layers*, an AOV containing the direct and indirect light loss generated by a given object on the rest of the scene. Such layers are useful inputs for the post-processing stage, as they effectively isolate the shadow component in the image. While providing an effective shadow editing tool, their method only considers solid objects, which prevents its application in many scenes. We propose to lift this restriction by defining an extended framework supporting arbitrary unions of solid and volumetric objects. In a typical usage scenario, a lighting artist selects the objects of the scene whose shadows are to be edited, and our algorithm renders both the main image and all generalised shadow layers in a single pass. The layers are then freely edited, as illustrated on the BEACH scene (Figure 1).

To that end, we propose the following contributions:

- The quantification of shadow created by a participating medium.
- A generalised path integral formulation to measure the shadow layer of finite unions of solid and volumetric objects.

In Section 3, we present an intuitive method to obtain a shadow layer from the image space difference of two specific renders of the scene. Due to its intrinsic rendering overhead, this definition is intractable in practice. We thus present a more flexible, single pass solution. After reminding the basic properties of participating media and their interaction with light, we compare light transport with and without volumetric interactions in Section 4. This analysis allows us to quantify the radiance lost when a ray traverses a participating medium. It also shows that no practical rendering equation can be derived for shadows, and we thus present in Section 5 a path integral formulation alleviating this shortcoming.

Our rendering algorithm, described in Section 6, is derived from this formulation. Based on volumetric path tracing, it is amenable to any rendering framework supporting participating media. While the algorithm reduces the number of zero radiance paths generated, it incurs a performance overhead and distributes the original sampling budget over separate layers. We analyse the resulting images in Section 7, and assess the implementation performance in terms of rendering time and convergence.

## 2. Related Work

**Light transport editing** Several techniques providing artistic control over light transport inside a scene have been proposed. The user can perform non-physical edits on light paths by applying linear transformations [SNM*13], or teleporting light from one location to another using portals [SMVP17]. Another interaction consists in scaling and offsetting light throughput in a 6D structure describing transport between two points [OKP*08]. Related to participating media, the method of Nowrouzezahrai *et al.* [NJS*11] turns the physical properties of emissive volumes into artistically directable quantities. While these approaches require re-rendering the scene after each edit, our method generates AOVs that are exported once and enable real-time compositing of shadows.

**Shadow manipulation** Shadows are an indirect phenomenon, caused by objects occluding light rays in the scene. Indirect manipulation of shadows is enabled by applying transforms to their shape [PTG02], or editing their boundary points [MIW13]. Assuming direct environment lighting, the user can alter the visibility of objects to produce non-physical results [OPP10], or rework shadows by drawing strokes over the render [Pel10]. These methods apply to real-time scenarios where the relation between the light source and cast shadow can be inverted. Conversely, our method is designed for offline global illumination rendering, and does not depend on the type of lighting primitive. In the work of DeCoro *et al.* [DCFR07], shadows are transformed in image space using predefined filters; our contribution shares their motivation but does not constrain the range of edits, deferring it to compositing.

**Differentiable rendering** Shadow has been characterised as the derivative of radiance with respect to object visibility [RMB07]. The general topic of differentiable global illumination has since received a lot of attention, as exporting image derivatives with respect to scene parameters enables gradient-based optimisations such as inverse rendering and backpropagation. Differentiable rendering algorithms must keep track of many parameters using a limited memory budget; reverse mode differentiation effectively reduces memory usage, but the computation time evolves quadratically with path length [NDSRJ20]. Vicini *et al.* have achieved constant memory usage with linear complexity, but drop support for heterogeneous media [VSJ21]. The visibility of objects introduces highly discontinuous integrands when differentiating light transport equations, and requires dedicated schemes when sampling area [LHJ19, BLD20] or boundary edges [LADL18, ZWZ*19], even with path-space approaches [ZMY*20, ZYZ21]. Despite this extensive literature, many questions remain open regarding the mathematics of differentiable rendering [ZSGJ21]. Because it is tailored to the specifics of shadow rendering, our solution is simpler and does not constrain the properties of participating media.

**Shadow in photographs** Separating shadowing effects from the appearance of surfaces in photographs is an ill-posed problem. Existing methods that transform a picture into its shadowless version typically involve a model with many parameters [FHLD06], explicitly rely on user hints [AHO11], or assume recurring structures among surfaces [GDH13]. Alternatively, neural networks incorporate strong priors that help the disambiguation [WLY18, LS19]. Philip *et al.* have used them in image-based rendering to plausibly reproduce shadows under new lighting conditions [PGZ*19, PMGD21], or after an extraneous object is added into the scenery [NPD20]. While these methods could be applied to extract approximate shadows in renders, we propose to reuse information from the lighting simulation for an exact result.

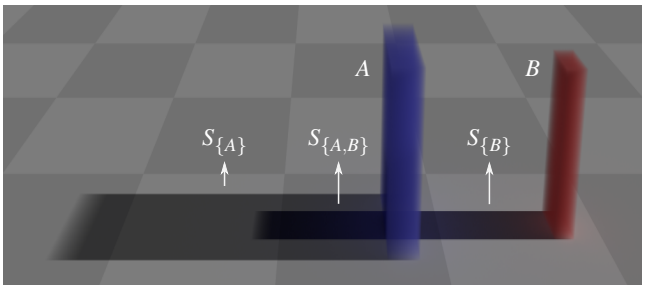



**Figure 2:** *In addition to the shadow layers of media A and B, our method picks up their mutual shadow in a separate layer $S_{\{A,B\}}$.*

**Shadow layers** Global Illumination Shadow Layers [DVP19] contain the radiance lost in the scene due to occlusions, estimated by a path tracing algorithm that stems from a path integral formulation of shadow. This approach has two main limitations. First, a shadow layer is associated with exactly one object, meaning that a mutual shadow created by one or more objects is not picked up (Figure 2); the total amount of shadow on surfaces of the scene is thus underestimated. Second, only the shadow of solid objects is estimated. In this work, we alleviate these two shortcomings by generalising the definition of shadow layers to participating media, and enabling the estimation of shadows created by multiple objects.

## 3. Shadow Layers from Image Subtraction

We begin this section by formally introducing the concept of shadow. We then present an intuitive approach to obtain shadow layers from image subtraction, and discuss its computational cost.

Global illumination algorithms estimate radiance inside scenes where both solid and volumetric objects interact with light. Locally, these interactions have an impact on the light field that can be split into two opposite terms:

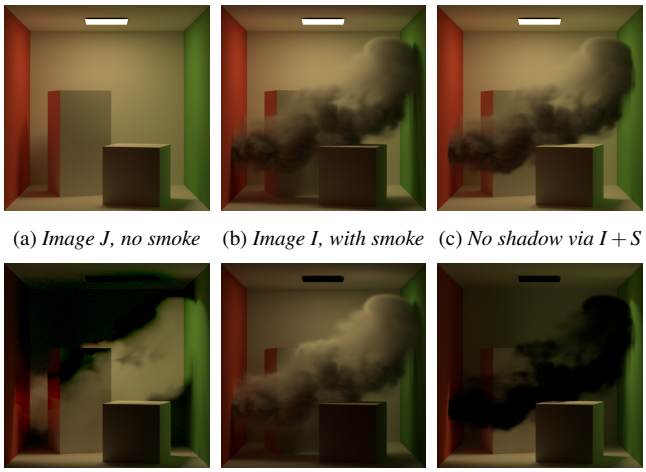
- A local gain of energy when an object emits or scatters light towards the outgoing the direction.
- A shadow, *i.e.* a local loss in the opposite of the incoming direction when an object scatters or absorbs light.

When light interacts with participating media, this duality appears in the radiative transfer equation [Cha60] and explains the variation of radiance at a position $x$ along direction $\omega$:

$$\omega \cdot \nabla L(x,\omega) = Q(x,\omega) - \sigma(x)L(x,\omega)\ , \qquad (1)$$

where $Q$ corresponds to gains from emission and in-scattering, and $-\sigma L$ encompasses losses due to absorption and out-scattering. The attenuation coefficient of the medium $\sigma = \sigma_a + \sigma_s$ is the sum of the absorption and scattering coefficients respectively.

We illustrate this duality on the CORNELL scene, the image $I$ of Figure 3b. In Figure 3a, we display a render $J$ where we prevented any interaction with the smoke simulation by removing it from the scene. The image resulting from the subtraction $J - I$ is seen in Figure 3d. It contains both gains and losses in radiance that interactions with the smoke create over the image. Because of the chosen convention, shadows bring positive radiance to the image while gains



(a) *Image J, no smoke* (b) *Image I, with smoke* (c) *No shadow via $I+S$*

(d) *Subtraction $J-I$* (e) *Scattered layer* (f) *Shadow layer S*

**Figure 3:** *Image (d) is the result of the subtraction $J - I$ between (a) a render $J$ without smoke and (b) the version $I$ with smoke. It contains radiance both gained and lost because of the medium. Our shadow layer $S$ depicted in (f) isolates the losses, and can be added with $I$ to remove shadows, as shown in (c).*

have a negative contribution. We generate Figure 3e and Figure 3f following Desrichard *et al.* [DVP19]: we compute a third render where the object of interest is made completely absorbing, which allows to further split the subtraction $J - I$ into gains (scattered layer) and losses (shadow layer) respectively. The shadow layer $S$ can then be added to the main image $I$ to remove any shadow created by the object, as seen in Figure 3c.

The scattered layer is also described using an extension of Heckbert notation [Hec90] called *light path expressions* [Gri16]. For instance in the light path expression `.*<[RT].'object'>.*` the wildcard `.*` matches anything, and the disjunction `[RT]` ensures that matched paths contain at least one reflection or transmission on `object`. An example of scattered layer is shown in Figure 3e, and corresponds to light paths interacting with the smoke.

As scattered layers are already well supported in off-the-shelf renderers using light path expressions, we focus on efficiently computing shadow layers. While intuitive, the image space subtraction requires two additional renders per object, leading to a total of $2N+1$ renders for $N$ separate objects – without considering mutual interactions that occur when several objects occlude light to create shadow. This inherent computational overhead prevents the use of the image space approach in a production context. In the following, we reuse the intuition of comparing light transport with and without the impact of a given object to devise a more efficient solution. However, we carry out our analysis directly on light rays instead of working in image space.

## 4. Radiance Loss when Traversing a Participating Medium

We start with the case of a single participating medium, called *shadow caster* and denoted $\mathbf{c}$. Its radiance loss is computed as follows. First, the radiance reaching a point $x$ from direction $\omega$ is given

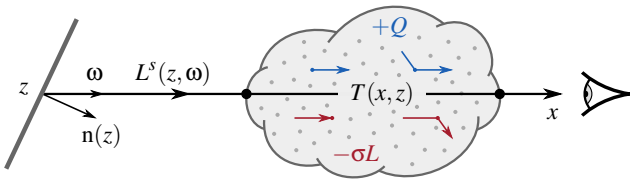



**Figure 4:** *The volume rendering equation summarises radiance exchanges between a medium and a light ray. Locally, positive contributions $+Q$ come from emission and in-scattering, while losses $-\sigma L$ are due to absorption and out-scattering.*

by the volume rendering equation [Arv93], which is the integrated version of Equation (1):

$$L(x,\omega) = \int_x^z T(x,y)Q(y,\omega)\,\mathrm{d}y + T(x,z)L^s(z,\omega)\ , \tag{2}$$

where $T$ is the transmittance, the factor quantifying the absorption and out-scattering of radiance between two points.

As illustrated in Figure 4, the point $z$ is located on the nearest visible surface from $x$ in direction $-\omega$, and $L^s(z,\omega)$ is the radiance leaving the surface from $z$ towards $x$. In turn, $L^s$ is comprised of surface emission and incoming light that is reflected or transmitted according to the bidirectional scattering distribution function (BSDF) $\rho$, integrated over the sphere of directions:

$$L^s(z,\omega) = L_e(z,\omega) + \int_{\mathcal{S}^2} \rho(z,\omega',\omega)\,|\mathrm{n}(z)\cdot\omega'|\,L(z,\omega')\,\mathrm{d}\omega'\ .$$

To quantify the amount of energy lost along the ray when traversing the medium, we consider the light field when the shadow caster is removed from the scene, *i.e.* no volumetric interaction happens between $\mathbf{c}$ and the ray. In this case, radiance is given by the rendering equation [Kaj86] and expressed concisely:

$$L(x,\omega) = L^s(z,\omega)\ . \tag{3}$$

The energy missing due to medium interactions is found by comparing Equations (2) and (3). In Equation (2), the integral along the ray is a purely positive contribution from the source term $Q$. The energy loss is located in the second term: only a fraction $T(x,z)$ of the radiance outgoing from the surface at $z$ reaches $x$ after absorption and out-scattering have been taken into account. The fraction of radiance lost along the ray is thus equal to $1-T(x,z)$.

However, generalising this observation to an entire light path is non-obvious as $1-T$ is not multiplicative: after two medium traversals, $1-T(x,z) \neq (1-T(x,y))(1-T(y,z))$ and so $1-T$ does not expand naturally into products. Even though the previous analysis would be sufficient for a single convex participating medium rendered with direct lighting, the derivation of a recursive shadow rendering equation in the general case is more tedious.

## 5. A Path Integral Formulation for Shadow

As explained above, shadow cannot be measured at the sensor by recursively accumulating the radiance lost over successive rays. In this section, we thus perform our derivation directly on light paths.

### 5.1. Classic Expansion with Participating Media

The path integral formulation [Vea97] is a practical framework where a measure $I$ at the sensor position $j$ is expressed as an integral over the set of all light paths $\Omega$:

$$I_j = \int_\Omega f_j(\bar{\mathrm{x}})\,\mathrm{d}\mu(\bar{\mathrm{x}})\ . \tag{4}$$

This path integral applies to scenes containing participating media and we remind the details of its expansion, adapted from Pauly *et al.* [PKK00]. The differential measure $\mathrm{d}\mu$ of a light path $\bar{\mathrm{x}} = \mathrm{x}_0 \ldots \mathrm{x}_k$ develops into the product of the usual surface or volumetric measures, depending on where each vertex $\mathrm{x}_i$ is located. With paths going from the source to the sensor, the measurement contribution function $f$ is defined as:

$$\begin{aligned} f_j(\bar{x}) = {} & L_e(\mathrm{x}_0,\mathrm{x}_1)W_e^j(\mathrm{x}_{k-1},\mathrm{x}_k)G(\mathrm{x}_0,\mathrm{x}_1)VT(\mathrm{x}_0,\mathrm{x}_1) \\ & \cdot \prod_{i=1}^{k-1} F(\mathrm{x}_{i-1},\mathrm{x}_i,\mathrm{x}_{i+1})G(\mathrm{x}_i,\mathrm{x}_{i+1})VT(\mathrm{x}_i,\mathrm{x}_{i+1})\ , \end{aligned} \tag{5}$$

where $L_e(\mathrm{x}_0,\mathrm{x}_1)$ is the radiance emitted from $\mathrm{x}_0$ towards $\mathrm{x}_1$ and $W_e^j(\mathrm{x}_{k-1},\mathrm{x}_k)$ is the importance leaving the sensor from $\mathrm{x}_k$ towards $\mathrm{x}_{k-1}$. The transmittance $T$ is multiplied with the visibility term $V$, which equals 1 if the two points are mutually visible, and 0 otherwise. The other terms depend on the vertices location:

$$G(x,y) = \frac{D(x,y)D(y,x)}{\|x-y\|^2} \quad \text{with}$$

$$D(x,y) = \begin{cases} |\mathrm{n}(x)\cdot\omega_{x\to y}| & \text{if } x \text{ is on a surface} \\ 1 & \text{if } x \text{ is in a medium}\ , \end{cases}$$

and the term $F(x,y,z)$ corresponds to either the BSDF on a surface, or the phase function inside a medium.

### 5.2. Generalisation to the Measure of Shadow

We denote $\mathbf{O}$ the set of all objects in the scene. We only assume that $\mathbf{O}$ is finite and well-defined, *i.e.* its elements can be uniquely identified. Among them, we wish to measure the shadow layer of the objects subset $\mathbf{C} \subset \mathbf{O}$. As illustrated in Figure 2, accounting for sets of objects is necessary to pick up radiance lost from occlusion by several objects. Starting from the classic formulation, we need to generalise a few observations presented in Section 4 to express the shadow of $\mathbf{C}$ as a path integral.

**Domain of integration** To quantify the radiance loss along a ray, we did not consider the integral term of Equation (2) as it corresponds to an energy gain from the source term $Q$. In the path integral framework, this gain is measured by light paths containing at least one volumetric interaction with the medium. More generally, the set of light paths encountering an object $\mathbf{o}$ measures the radiance gains brought by $\mathbf{o}$ to the sensor. This idea is subsumed by the definition of the scattered layer (Section 3). In order to measure shadow, we need to ignore radiance gains and the light paths carrying them. We thus change the domain of integration compared to Equation (4): we define $\Omega_{\mathbf{C}}$ the set of all paths with at least one interaction on a caster in $\mathbf{C}$, and integrate over $\Omega \setminus \Omega_{\mathbf{C}}$.

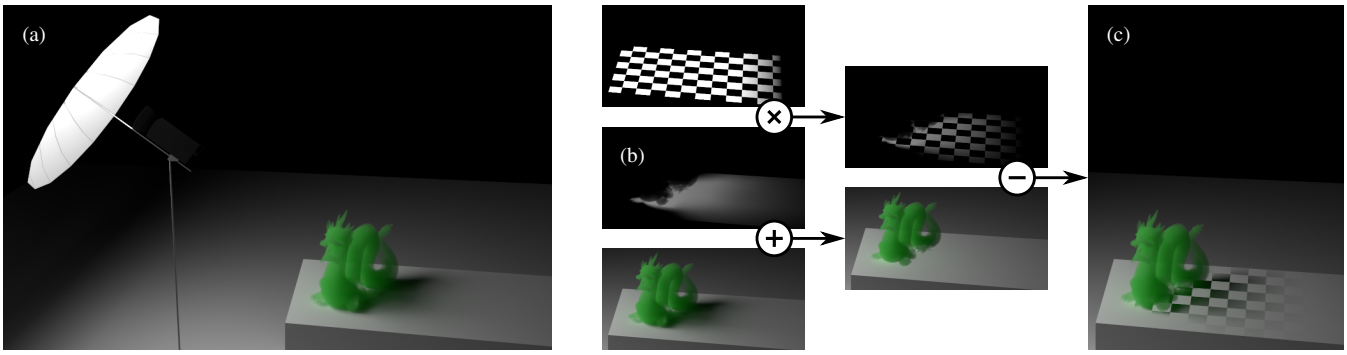



**Figure 5:** *(a) In this scene, the dragon is filled with an homogeneous medium and receives purely indirect illumination from the reflector. (b) The shadow layer of the dragon correctly picks up indirect shadow all over the pedestal. Using a simple compositing graph, the shadow of the subject is first removed from the image, transformed separately, and then (c) brought back with a checkerboard pattern inlaid.*

**Complementary of visibility** Between two successive vertices of a light path, media may decrease the transmittance $T$ while solid objects may cancel the visibility term $V$. We quantify the radiance loss due to a single solid occluder between two vertices by following the analysis presented in Section 4, and taking the complementary of visibility $1-V$. As for transmittance, the complementary of visibility is not a multiplicative quantity and we cannot directly accumulate products of $1-V$ over a light path. Instead, we group all the *loss factors* $VT$, and take the complementary of their product.

Knowing that objects in $\mathbf{O}$ can be uniquely identified, we begin by isolating the radiance loss due to each object $\mathbf{o} \in \mathbf{O}$ in Equation (5) using the subscript notation $V_\mathbf{o} T_\mathbf{o}$:

$$\prod_{i=0}^{k-1} VT(\mathrm{x}_i, \mathrm{x}_{i+1}) = \prod_{\mathbf{o}\in\mathbf{O}} \prod_{i=0}^{k-1} V_\mathbf{o} T_\mathbf{o}(\mathrm{x}_i, \mathrm{x}_{i+1}) = \prod_{\mathbf{o}\in\mathbf{O}} V_\mathbf{o} T_\mathbf{o}(\bar{\mathrm{x}}) \,,$$

where we denote $V_\mathbf{o} T_\mathbf{o}(\bar{\mathrm{x}})$ the loss factor of object $\mathbf{o}$ over the path $\bar{\mathrm{x}}$. This reordering allows us to circumvent the non-multiplicativity of $1-VT$ and to take the complementary of each loss factor to obtain the radiance loss over the whole path. This defines the measurement contribution function for the shadow layer of the object set $\mathbf{C}$:

$$\begin{aligned} f_{\mathbf{C},j}(\bar{\mathrm{x}}) = {} & L_e(\mathrm{x}_0, \mathrm{x}_1) W_e^j(\mathrm{x}_{k-1}, \mathrm{x}_k) G(\mathrm{x}_0, \mathrm{x}_1) \\ & \cdot \prod_{i=1}^{k-1} F(\mathrm{x}_{i-1}, \mathrm{x}_i, \mathrm{x}_{i+1}) G(\mathrm{x}_i, \mathrm{x}_{i+1}) \\ & \cdot \prod_{\mathbf{o}\in\mathbf{O}\setminus\mathbf{C}} V_\mathbf{o} T_\mathbf{o}(\bar{\mathrm{x}}) \cdot \prod_{\mathbf{c}\in\mathbf{C}} (1 - V_\mathbf{c} T_\mathbf{c}(\bar{\mathrm{x}})) \,. \end{aligned} \tag{6}$$

Reinjecting Equation (6) into the path integral formulation and restricting the domain of integration to $\Omega \setminus \Omega_\mathbf{C}$ yields the shadow layer measurement at sensor position $j$:

$$S_{\mathbf{C},j} = \int_{\Omega\setminus\Omega_\mathbf{C}} f_{\mathbf{C},j}(\bar{\mathrm{x}}) \, \mathrm{d}\mu(\bar{\mathrm{x}}) \,. \tag{7}$$

The set of shadow casters $\mathbf{C}$ can be empty, in which case the path integral is simply that of the main image $I$. As detailed in Appendix A, when $\mathbf{C} = \{\mathbf{c}\}$ with $\mathbf{c}$ a solid object, this formulation is equivalent to that of a Global Illumination Shadow Layer [DVP19].

This definition naturally handles global illumination effects such as indirect shadows created by the occlusion of reflected light sources, as shown in the DRAGON scene (Figure 5). Overlapping objects are also accounted for, thanks to the separation of radiance loss factors per-object. Our path integral is general, and makes no assumptions on the physical properties of participating media. In practice, it enables efficient rendering of shadow layers using Monte Carlo integration, as detailed next.

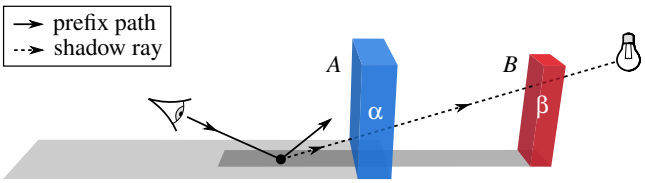
## 6. Integration in a Path Tracing Framework

We present a rendering algorithm based on volumetric path tracing that computes the main image and any number of shadow layers in a single pass. It only requires as parameters the shadow-casting objects $\mathbf{c}_1, \ldots, \mathbf{c}_N$ for which shadow layers are requested by the user. We do not consider the internal representation of media (density grid, analytic expression, *etc.*) and only require that both solid and volumetric objects are uniquely identified in the scene.

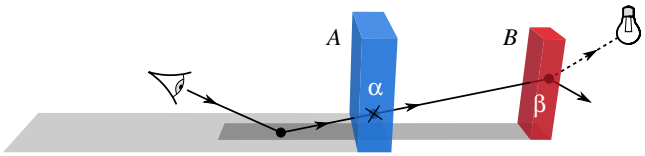
The algorithm samples random light paths $\bar{\mathrm{x}}$ and measures their contribution according to $f_\mathbf{C}$ for each of the $2^N$ possible unions of casters $\mathbf{C}$ that can be formed from the collection $\mathbf{c}_1, \ldots, \mathbf{c}_N$, as long as $\bar{\mathrm{x}} \in \Omega \setminus \Omega_\mathbf{C}$. In order to compute complementaries $1 - V_{\mathbf{c}_i} T_{\mathbf{c}_i}$ in each shadow layer $S_\mathbf{C}$ containing $\mathbf{c}_i$, we need to accumulate $N$ loss factors during rendering. This is achieved through the modification of two key steps: the construction of the prefix path starting from the camera, and the gathering of direct light at each interaction.

**Path construction** Indirect shadows are created when the light emitted from a source is scattered at least once, and then occluded before it reaches the camera. Because the path tracer follows the inverse direction of light, changing the way the prefix path is built is necessary to account for indirect shadows. Our path tracer may ignore the first interaction with a shadow caster, and search for the amount of indirect shadow it creates. Specifically, there are two possible outcomes for the first interaction with each caster $\mathbf{c}_i$:

- The event is discarded, and the path does not interact with the caster anymore: $\mathbf{c}_i$ is traversed without scattering. Its loss factor $V_{\mathbf{c}_i} T_{\mathbf{c}_i}$ is set to 0, as an interaction with the object would have normally prevented light from going through along a straight line. As the path belongs to $\Omega \setminus \Omega_{\{\mathbf{c}_i\}}$, it contributes to shadow layers according to Equation (6).




*The shadow ray contributes to the original image $I$ with factor $\alpha\beta$, and to the mutual shadow layer $S_{\{A,B\}}$ of the two media with factor $1-\alpha\beta$.*




*A volumetric interaction sampled in A is ignored, and the path continues until another one is sampled in B. The next shadow ray does not contribute radiance to $I$, but indirect shadow to $S_{\{A\}}$ with factor $(1-\alpha)\beta = \beta$.*

**Figure 6:** *Our path tracing algorithm executed in the setup of Figure 2. It accounts for mutual (top) and indirect (bottom) shadows.*

- The event occurs normally, and thus the impact of $\mathbf{c}_i$ on propagation is acknowledged. If $\mathbf{c}_i$ is encountered again, the algorithm proceeds with the event. The path now belongs to $\Omega_{\{\mathbf{c}_i\}}$ and it will not contribute to shadow layers $S_{\mathbf{C}}$ for which $\mathbf{c}_i \in \mathbf{C}$.

These outcomes have probabilities $p$ and $1-p$, offering a trade-off between indirect shadow sampling in shadow layers, and indirect radiance sampling in the main image. By default, we set $p = 1/2$.

**Direct light gathering** At each vertex of the prefix path, the algorithm sends shadow rays towards light sources. Changing the behaviour of these shadow rays allows us to pick up direct shadow at the same time as radiance. As illustrated in Figure 6, a modified shadow ray keeps track of separate loss factors $V_{\mathbf{c}_i} T_{\mathbf{c}_i}$ for each traversed caster $\mathbf{c}_i$ that was never encountered during propagation. This implies that solid occluders may be ignored when testing visibility with the light if they are marked as casters. The radiance emitted by the source then contributes to each shadow layer according to Equation (6), where we combine the loss factors stored in the shadow ray with those of the prefix path.

These modifications of the path tracing algorithm are sufficient to measure shadow at the same time as radiance. We now describe two additions that provide greater user control for editing.

**Shadow catchers** When rendered using the previous algorithm, shadow layers display the background wherever a caster is directly visible, as in Figure 3d. This is because shadow is measured over the sensor: if the camera's sensor has a rectangular shape inside the scene, the shadow layer contains the amount of radiance missing from this rectangle. For proper artistic editing, shadow must instead be measured over objects in the scene. We call those receiving objects the *shadow catchers*, according to the prevalent terminology. The user specifies which objects are considered as shadow catchers; if none is given, any object of the scene becomes a valid catcher. In the rendering loop, the measure of shadow can only begin after the path has interacted with a shadow catcher.

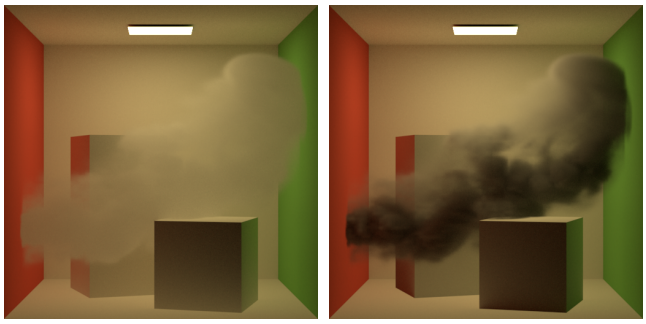


**Figure 7:** *Left: shadow removal for the smoke and boxes, including self-shadowing. Right: the same setup, ignoring self-shadowing.*

**Self-shadowing** When an object is identified as both a shadow caster and catcher, it may exhibit self-shadowing *i.e.* display a local loss of radiance due to itself. As shown in Figure 7, the result of self-shadowing is much more pronounced on participating media compared to solid objects because it also includes the absorbed radiance. Taking into account self-shadowing for a medium can impede artistic editing, which is why we allow the user to turn off their extraction for any object $\mathbf{o}$. This is done by preventing contributions to shadow layers involving $\mathbf{o}$ whenever the first shadow catcher encountered during propagation is $\mathbf{o}$ itself. In all presented figures (excepted Figure 7 left), self-shadowing is ignored.

## 7. Results and Performance

Our prototype shadow integrator is based on the volumetric path tracer of pbrt-v3 [PJH16]. We ran our measurements on an Intel Xeon E5-2630 v4 processor with 20 threads at 2.20 GHz and 64 GB RAM, and display the results in Table 1. $N$ is the number of shadow casters in the scene. When $N = 0$, the standard volumetric path tracer is used to render only the main image and when $N > 0$, our algorithm renders the main image and $2^N - 1$ shadow layers. We include in Table 1 the rendering times, Root-Mean-Square Error (RMSE) between main images, and Zero Radiance Paths (ZRP) percentage for the different figures of this paper.

Turning on the export of shadow layers incurs a performance overhead in rendering time and memory usage, attributable to several factors. First, while managing additional images has a predictable memory footprint, it involves running numerous 2-dimensional loops to apply reconstruction filters around the samples. Second, and most observable in the measurements, our algorithm accumulates the product of $N$ loss factors to form $2^N$ factors over every sampled path. Indeed when incoming radiance $L$ is picked up, each of the $2^N$ layers $i$ receives a contribution $\alpha_i L$ where $\alpha_i$ is the final loss factor over the light path. Factor $\alpha_i$ is the accumulated product of all $V_{\mathbf{o}} T_{\mathbf{o}}$ or their complementary $1 - V_{\mathbf{o}} T_{\mathbf{o}}$, according to Equation (6). The overhead remains consistently under 15% in our experiments except for the CHESS scene, which is designed for paths to encounter many casters. Performance for this setup is compared separately in Figure 8 by increasing the number of shadow casters from $N = 0$ to 10 and measuring rendering times. The resulting data exhibits the $2^N$ complexity expected when accounting for all possible interactions between $N$ casters.

| Scene | N | SPP | Time | RMSE·$10^3$ | ZRP |
|---|---|---|---|---|---|
| BEACH (Fig. 1) | 0 | 256 | 67'44" | 1.6333 | 71% |
| | 3 | 256 | 68'51" | 1.6350 | 43% |
| CORNELL (Fig. 3) | 0 | 1024 | 8' 08" | 1.2833 | 56% |
| | 1 | 1024 | 8' 23" | 1.3454 | 28% |
| | 3 | 1024 | 9' 13" | 1.7062 | 21% |
| DRAGON (Fig. 5) | 0 | 4096 | 17'23" | 7.2070 | 92% |
| | 1 | 4096 | 17'58" | 7.2790 | 91% |
| MANHOLE (Fig. 9) | 0 | 512 | 9'57" | 2.5707 | 30% |
| | 1 | 512 | 10'12" | 2.5732 | 18% |

**Table 1:** *Performance comparison for N shadow casters between a standard path tracer ($N = 0$) and our shadow integrator ($N > 0$). Rendering times increase with an overhead consistently under 15%; the RMSE is computed for the main images, compared to a converged reference. The systematic decrease in Zero Radiance Paths (ZRP) with more layers supports our single pass approach.*

Compared to a standard path tracer, convergence at each pixel is also affected. The same number of samples is now distributed among a number of different layers, meaning that the main image does not receive as many contributions as usual. This is measured by the RMSE, which we computed against a converged reference image that uses at least 16 times the sampling budget.

The BEACH scene is a rendering of the Moana Island Scene featuring around 50 million unique triangles and 20 million parametric curves (see Figure 1). We disposed three distinct instances of Walt Disney Animation Studios' Cloud Data Set at half resolution above the beach, for a total of 9.2 GB uncompressed density data. Measurements show that our algorithm has a negligible impact on this production-grade scene, where the overall complexity of surfaces and volumes prevails: the overhead in rendering time is under 2%, while the convergence of the main image is barely affected.

In the CORNELL scene, we begin by rendering a single shadow layer for the smoke, and follow with a total of 8 layers by also considering the two boxes. While the number of zero radiance paths systematically decreases, we notice a deterioration of performance in rendering time and convergence. While rendering time is mainly affected by the number of layers, the worse convergence is due to the light source reflecting on walls, which creates strong indirect shadows spanning a large area of the image. In turn, many generated paths end up measuring indirect shadows, and do not contribute to the main image anymore.

Our approach is also compatible with animated scenes, as demonstrated by the MANHOLE sequence of Figure 9. In this example we display the *shadow ratio*, defined as $I/(I+S)$. This quantity represents the ratio of occluded radiance at a pixel, and its intuitive signification makes it useful for image space manipulation. We refer to the supplemental material for an editing session using third-party post-processing software.

*The CHESS scene involves occlusions by different casters along light paths.*

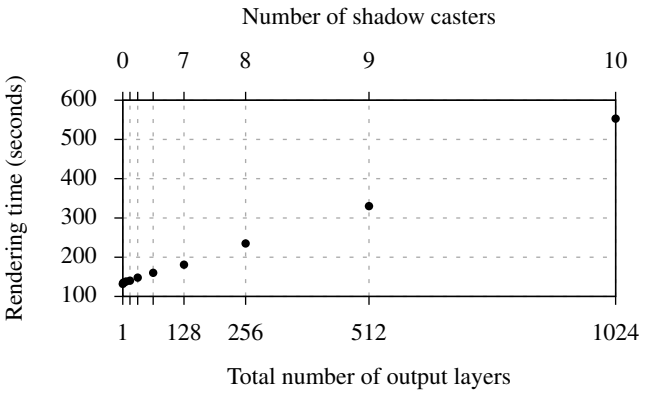



**Figure 8:** *Evolution of the rendering time with the number of output layers in the CHESS scene (top) at 1024 samples per pixel. The number of layers increases exponentially with the number of shadow casters, but the rendering time per layer remains constant.*

While our single pass approach requires more computations, it is legitimated by the systematic decrease in zero radiance paths. Building paths is costly as it involves numerous intersection tests to simulate the propagation of rays. By applying a different measurement contribution functions for each rendered layer to every single path, we fully leverage the cost of intersections. In many instances, we also reuse otherwise lost information: even when it is exponential, transmittance can be null in practice when delta tracking is used for its estimation. In this case, our algorithm picks up shadow where a standard path tracer would return no radiance.

## 8. Conclusion and Future Work

We have introduced a new definition of shadow layers in the context of global illumination, that is compatible with participating media. By comparing light transport with and without the impact of a participating medium, we have shown that the radiance loss along a ray is quantified by the complementary of transmittance. However, whereas transmittance is most often multiplicative and expands naturally into products, its complementary does not, which prevents the derivation of a recursive rendering equation. We thus resorted to the path integral formulation to properly define the shadow layer in the presence of multiple occlusions, leading us to a general formulation that is amenable to Monte Carlo integration.

We presented our path tracing algorithm, which renders both the main image and any number of shadow layers in a single pass. It incurs an overhead in rendering time of around 15%, but efficiency is improved as the number of paths carrying no radiance systematically decreases, validating our single pass approach. While our

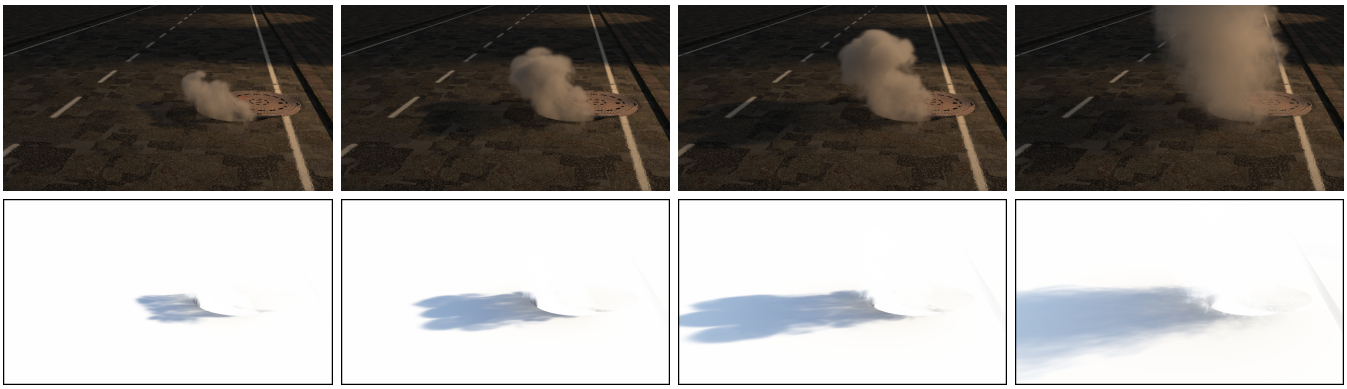


**Figure 9:** *A selection of frames from an animated sequence. The top row shows main images I, the bottom row shows shadow ratios. The shadow ratio is a useful quantity for editing, defined as $I/(I+S)$. It evolves consistently with the expansion of smoke, showing that our method applies to animated scenes. Please refer to the supplemental material for the complete footage and an example compositing session.*

renderer distributes samples over all layers, we noticed a minor increase in RMSE when comparing the main image with the result of a standard path tracer. Our method makes no particular assumption on the properties of the medium.

### Limitations

The main limitation of our algorithm appears when a large number $N$ of shadow casters is taken as input. Because accounting for all possible unions yields $2^N$ different shadow layers, rendering complexity increases exponentially. The current implementation removes shadow layers with negligible energy after rendering, but it does compute them all. As a workaround, the user can identify multiple objects as one to reduce the total number of layers, at the cost of ignoring their mutual interactions.

### Future work

We would like to investigate how to ignore some shadow layers during rendering to save computational power. This could be achieved manually by the user, but all possible interactions between many objects are not obvious to figure out; alternatively, two automated solutions are possible. The first is to limit the maximum size of the casters sets $|\mathbf{C}|$ to a fixed number, and thus consider only low-order interactions between objects. The second is to add an initial bootstrap pass that estimates the contribution of each layer, and discards those containing too little energy.

When the distribution of volumetric scatterers is correlated, the transmittance can take various non-exponential profiles [BRM*18]; we believe that our formulation extends in this case. The purpose of our path integral is to make up for the radiance lost due to volumetric interactions along full paths, which is possible as long as the loss is quantified by a factor $T(\bar{\mathrm{x}}) \leq 1$.

Compositing artists often resort to *deep images* to properly rework renders containing translucent surfaces or participating media. Instead of storing a single colour value at a pixel, deep images contain a collection of slabs defined by a front depth, a back depth, a colour, and an opacity. Our prototype does not export deep images; their addition is orthogonal to our work, but possible.

We reckon that shadow layers can also be useful outside of the compositing pipeline, and apply to machine learning approaches where networks are trained to infer or remove shadow in photographs [WLY18, LS19], or perform image-based relighting with plausible shadows [PGZ*19, PMGD21, NPD20]. Compared to binary masks, our representation is linear and should better perform in linear operations such as convolution. Whereas shadow layers do not directly provide a segmentation as masks do, we believe that they could generate more accurate models for shadow removal and relighting, where no segmentation is required.

## 9. Acknowledgements

We thank the reviewers for their valuable feedback and suggestions. This work benefited from the support of project CaLiTrOp ANR-16-CE33-0026 of the French National Research Agency (ANR). The BEACH scene uses the Moana Island Scene assets and the Cloud Data Set, both distributed by Walt Disney Animation Studios. CORNELL was adapted to pbrt-v3 by Benedikt Bitterli. DRAGON uses a model from the Stanford 3D Scanning Repository.

## References

[AHO11] ARBEL E., HEL-OR H.: Shadow Removal Using Intensity Surfaces and Texture Anchor Points. *IEEE Transactions on Pattern Analysis and Machine Intelligence 33*, 6 (2011), 1202–1216. doi:10.1109/TPAMI.2010.157. 2

[Arv93] ARVO J.: Transfer Equations in Global Illumination. In *Global Illumination, SIGGRAPH '93 Course Notes* (1993). 4

[Bir13] BIRN J.: *Digital Lighting and Rendering: Third Edition*. Pearson Education, 2013. 1

[BLD20] BANGARU S. P., LI T.-M., DURAND F.: Unbiased Warped-Area Sampling for Differentiable Rendering. *ACM Trans. Graph. 39*, 6 (Nov. 2020). doi:10.1145/3414685.3417833. 2

[BRM*18] BITTERLI B., RAVICHANDRAN S., MÜLLER T., WRENNINGE M., NOVÁK J., MARSCHNER S., JAROSZ W.: A Radiative Transfer Framework for Non-Exponential Media. *ACM Trans. Graph. 37*, 6 (Dec. 2018). doi:10.1145/3272127.3275103. 8

[Cha60] CHANDRASEKHAR S.: *Radiative Transfer*. Dover Books on Intermediate and Advanced Mathematics. Dover Publications, 1960. 3

[DCFR07] DeCoro C., Cole F., Finkelstein A., Rusinkiewicz S.: Stylized Shadows. In *Proc. 5th International Symposium on Non-photorealistic Animation and Rendering* (New York, NY, USA, 2007), NPAR '07, ACM, pp. 77–83. doi:10.1145/1274871.1274884. 2

[DVP19] Desrichard F., Vanderhaeghe D., Paulin M.: Global Illumination Shadow Layers. *Computer Graphics Forum* (July 2019). doi:10.1111/cgf.13781. 2, 3, 5, 10

[FHH*19] Fascione L., Hanika J., Heckenberg D., Kulla C., Droske M., Schwarzhaupt J.: Path Tracing in Production: Part 1: Modern Path Tracing. In *ACM SIGGRAPH 2019 Courses* (New York, NY, USA, 2019), SIGGRAPH '19, ACM. doi:10.1145/3305366.3328079. 2

[FHLD06] Finlayson G., Hordley S., Lu C., Drew M.: On the Removal of Shadows From Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 1 (2006), 59–68. doi:10.1109/TPAMI.2006.18. 2

[GDH13] Guo R., Dai Q., Hoiem D.: Paired Regions for Shadow Detection and Removal. *IEEE Transactions on Pattern Analysis and Machine Intelligence 35*, 12 (2013), 2956–2967. doi:10.1109/TPAMI.2012.214. 2

[Gri16] Gritz L.: OSL Light Path Expressions. https://github.com/imageworks/OpenShadingLanguage/wiki/OSL-Light-Path-Expressions, 2016. Accessed 2021-09-01. 3

[Hec90] Heckbert P. S.: Adaptive Radiosity Textures for Bidirectional Ray Tracing. In *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1990), SIGGRAPH '90, ACM, pp. 145–154. doi:10.1145/97879.97895. 3

[Kaj86] Kajiya J. T.: The Rendering Equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1986), SIGGRAPH '86, ACM, pp. 143–150. doi:10.1145/15922.15902. 4

[LADL18] Li T.-M., Aittala M., Durand F., Lehtinen J.: Differentiable Monte Carlo Ray Tracing Through Edge Sampling. *ACM Trans. Graph. 37*, 6 (Dec. 2018), 222:1–222:11. doi:10.1145/3272127.3275109. 2

[LHJ19] Loubet G., Holzschuch N., Jakob W.: Reparameterizing Discontinuous Integrands for Differentiable Rendering. *Transactions on Graphics (Proceedings of SIGGRAPH Asia) 38*, 6 (Dec. 2019). doi:10.1145/3355089.3356510. 2

[LS19] Le H., Samaras D.: Shadow Removal via Shadow Image Decomposition. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), pp. 8577–8586. doi:10.1109/ICCV.2019.00867. 2, 8

[MIW13] Mattausch O., Igarashi T., Wimmer M.: Freeform Shadow Boundary Editing. *Computer Graphics Forum 32* (May 2013), 175–184. doi:10.1111/cgf.12037. 2

[NDSRJ20] Nimier-David M., Speierer S., Ruiz B., Jakob W.: Radiative Backpropagation: An Adjoint Method for Lightning-Fast Differentiable Rendering. *Transactions on Graphics (Proceedings of SIGGRAPH) 39*, 4 (July 2020). doi:10.1145/3386569.3392406. 2

[NJS*11] Nowrouzezahrai D., Johnson J., Selle A., Lacewell D., Kaschalk M., Jarosz W.: A Programmable System for Artistic Volumetric Lighting. *ACM Trans. Graph. 30*, 4 (July 2011), 29:1–29:8. doi:10.1145/2010324.1964924. 2

[NPD20] Nicolet B., Philip J., Drettakis G.: Repurposing a Relighting Network for Realistic Compositions of Captured Scenes. In *Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2020), I3D '20, ACM. doi:10.1145/3384382.3384523. 2, 8

[OKP*08] Obert J., Křivánek J., Pellacini F., Sýkora D., Pattanaik S.: iCheat: A Representation for Artistic Control of Indirect Cinematic Lighting. *Computer Graphics Forum 27*, 4 (2008), 1217–1223. doi:10.1111/j.1467-8659.2008.01260.x. 2

[OPP10] Obert J., Pellacini F., Pattanaik S.: Visibility Editing for All-Frequency Shadow Design. *Computer Graphics Forum 29*, 4 (2010), 1441–1449. doi:10.1111/j.1467-8659.2010.01741.x. 2

[Pel10] Pellacini F.: envyLight: An Interface for Editing Natural Illumination. In *ACM SIGGRAPH 2010 Papers* (New York, NY, USA, 2010), SIGGRAPH '10, ACM. doi:10.1145/1833349.1778771. 2

[PGZ*19] Philip J., Gharbi M., Zhou T., Efros A. A., Drettakis G.: Multi-View Relighting Using a Geometry-Aware Network. *ACM Trans. Graph. 38*, 4 (July 2019). doi:10.1145/3306346.3323013. 2, 8

[PJH16] Pharr M., Jakob W., Humphreys G.: *Physically Based Rendering: From Theory to Implementation*, 3rd ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2016. 6

[PKK00] Pauly M., Kollig T., Keller A.: Metropolis Light Transport for Participating Media. In *Rendering Techniques 2000* (Vienna, 2000), Péroche B., Rushmeier H., (Eds.), Springer Vienna, pp. 11–22. doi:10.1007/978-3-7091-6303-0_2. 4

[PMGD21] Philip J., Morgenthaler S., Gharbi M., Drettakis G.: Free-Viewpoint Indoor Neural Relighting from Multi-View Stereo. *ACM Trans. Graph. 40*, 5 (Sept. 2021). doi:10.1145/3469842. 2, 8

[PTG02] Pellacini F., Tole P., Greenberg D. P.: A User Interface for Interactive Cinematic Shadow Design. *ACM Trans. Graph. 21*, 3 (July 2002), 563–566. doi:10.1145/566654.566617. 2

[RMB07] Ramamoorthi R., Mahajan D., Belhumeur P.: A First-Order Analysis of Lighting, Shading, and Shadows. *ACM Trans. Graph. 26*, 1 (Jan. 2007), 2–23. doi:10.1145/1189762.1189764. 2

[SCC18] Santos P. E., Casati R., Cavanagh P.: Perception, Cognition and Reasoning about Shadows. *Spatial Cognition & Computation 18*, 2 (2018), 78–85. doi:10.1080/13875868.2017.1377204. 1

[SMVP17] Subileau T., Mellado N., Vanderhaeghe D., Paulin M.: RayPortals: A Light Transport Editing Framework. *Vis. Comput. 33*, 2 (Feb. 2017), 129–138. doi:10.1007/s00371-015-1163-2. 2

[SNM*13] Schmidt T.-W., Novák J., Meng J., Kaplanyan A. S., Reiner T., Nowrouzezahrai D., Dachsbacher C.: Path-space Manipulation of Physically-based Light Transport. *ACM Trans. Graph. 32*, 4 (July 2013), 129:1–129:11. doi:10.1145/2461912.2461980. 2

[Vea97] Veach E.: *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, Stanford, CA, USA, 1997. 4

[VSJ21] Vicini D., Speierer S., Jakob W.: Path Replay Backpropagation: Differentiating Light Paths using Constant Memory and Linear Time. *ACM Trans. Graph. 40*, 4 (Aug. 2021), 108:1–108:14. doi:10.1145/3450626.3459804. 2

[WLY18] Wang J., Li X., Yang J.: Stacked Conditional Generative Adversarial Networks for Jointly Learning Shadow Detection and Shadow Removal. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 1788–1797. doi:10.1109/CVPR.2018.00192. 2, 8

[ZMY*20] Zhang C., Miller B., Yan K., Gkioulekas I., Zhao S.: Path-Space Differentiable Rendering. *ACM Trans. Graph. 39*, 4 (July 2020), 143:1–143:19. doi:10.1145/3386569.3392383. 2

[ZSGJ21] Zeltner T., Speierer S., Georgiev I., Jakob W.: Monte Carlo Estimators for Differential Light Transport. *Transactions on Graphics (Proceedings of SIGGRAPH) 40*, 4 (Aug. 2021). doi:10.1145/3450626.3459807. 2

[ZWZ*19] Zhang C., Wu L., Zheng C., Gkioulekas I., Ramamoorthi R., Zhao S.: A Differential Theory of Radiative Transfer. *ACM Trans. Graph. 38*, 6 (Nov. 2019). doi:10.1145/3355089.3356522. 2

[ZYZ21] Zhang C., Yu Z., Zhao S.: Path-Space Differentiable Rendering of Participating Media. *ACM Trans. Graph. 40*, 4 (July 2021), 76:1–76:15. doi:10.1145/3450626.3459782. 2

**Appendix A:** Equivalent formulation for a single solid object

We focus on a special case of the path integral formulation for shadow (Section 5) where the scene contains solid surfaces, and the set of casters is a single object: $\mathbf{C} = \{\mathbf{c}\}$. This setup is identical to Desrichard *et al.* [DVP19], and we demonstrate the equivalence with our work. Omitting the dependence on the sensor position $j$ and light path $\bar{\mathrm{x}}$, their path integral for the shadow layer of $\mathbf{C}$ is

$$S_{\mathbf{c}} = \int_{\Omega_{\mathbf{C}}} f_{T,\mathbf{c}} \, \mathrm{d}\mu \tag{8}$$

with $f_{T,\mathbf{c}}$ the measurement contribution function where the BSDF of $\mathbf{c}$ is replaced by $F_{\mathbf{c}}$, a straight transmission. $F_{\mathbf{c}}(\mathrm{x}_{i-1}, \mathrm{x}_i, \mathrm{x}_{i+1})$ is non-null only when $\mathrm{x}_{i-1}$, $\mathrm{x}_i$ and $\mathrm{x}_{i+1}$ are collinear and arranged in this order. Parameterised by directions, it is written

$$F_{\mathbf{c}}(x, \omega_i, \omega_o) = \frac{\delta(\omega_i + \omega_o)}{|\mathrm{n}(x)\omega_i|} \, .$$

The path integrals in Equations (4) and (8) differ by their integration domains; to connect the two, we define $\varphi : \Omega_{\mathbf{C}} \to \Omega \setminus \Omega_{\mathbf{C}}$ that removes all intersections between a path and $\mathcal{M}_{\mathbf{c}}$ the surface of $\mathbf{c}$. The mapping $\varphi$ is highly surjective, as many paths in $\Omega_{\mathbf{C}}$ end up having the same image in $\Omega \setminus \Omega_{\mathbf{C}}$. We thus reconstruct $\Omega_{\mathbf{C}}$ from preimages $\varphi^{-1}(B)$ as follows:

$$\Omega_{\mathbf{C}} = \bigsqcup_{k=1}^{+\infty} \varphi^{-1}(\Omega \setminus \Omega_{\mathbf{C}} \cap \Omega_k) \, .$$

From this observation, Equation (8) becomes

$$\int_{\Omega_{\mathbf{C}}} f_{T,\mathbf{c}} \, \mathrm{d}\mu = \sum_{k=1}^{+\infty} \int_{\varphi^{-1}(\Omega \setminus \Omega_{\mathbf{C}} \cap \Omega_k)} f_{T,\mathbf{c}} \, \mathrm{d}\mu \, ,$$

and we also know that Equation (4) decomposes into

$$\int_{\Omega \setminus \Omega_{\mathbf{C}}} f_{\mathbf{C}} \, \mathrm{d}\mu = \sum_{k=1}^{+\infty} \int_{\Omega \setminus \Omega_{\mathbf{C}} \cap \Omega_k} f_{\mathbf{C}} \, \mathrm{d}\mu \, .$$

The equivalence will follow if we prove that for $k > 1$,

$$\int_{\varphi^{-1}(\Omega \setminus \Omega_{\mathbf{C}} \cap \Omega_k)} f_{T,\mathbf{c}} \, \mathrm{d}\mu = \int_{\Omega \setminus \Omega_{\mathbf{C}} \cap \Omega_k} f_{\mathbf{C}} \, \mathrm{d}\mu \, . \tag{9}$$

We assume that emitted radiance $L_e$ and importance $W_e$ are null over $\mathcal{M}_{\mathbf{c}}$. Let us focus on $k = 1$ and consider paths of length 2 in $\varphi^{-1}(\Omega \setminus \Omega_{\mathbf{C}} \cap \Omega_1)$ that encounter $\mathcal{M}_{\mathbf{c}}$ on their second vertex. They have the form $\bar{\mathrm{x}} = \mathrm{x}_0 \mathrm{x}_1 \mathrm{x}_2$, where $\mathrm{x}_1 \in \mathcal{M}_{\mathbf{c}}$. The BSDF $F_{\mathbf{c}}$ constrains the shape of paths that bring a non-null contribution to the integral. When we write the measurement contribution function

$$f_{T,\mathbf{c}}(\bar{\mathrm{x}}) = L_e(\mathrm{x}_0, \mathrm{x}_1) GV(\mathrm{x}_0, \mathrm{x}_1) F_{\mathbf{c}}(\mathrm{x}_0, \mathrm{x}_1, \mathrm{x}_2) GV(\mathrm{x}_1, \mathrm{x}_2) W_e(\mathrm{x}_1, \mathrm{x}_2)$$

where $GV(x,y) = G(x,y)V(x,y)$, the collinearity of $\mathrm{x}_0$, $\mathrm{x}_1$, and $\mathrm{x}_2$ implies that the measurement does not change if we replace

$$L_e(\mathrm{x}_0, \mathrm{x}_1) = L_e(\mathrm{x}_0, \mathrm{x}_2) \quad \text{and} \quad W_e(\mathrm{x}_1, \mathrm{x}_2) = W(\mathrm{x}_0, \mathrm{x}_2) \, . \tag{10}$$

Also, Figure 10 illustrates the behaviour of the geometric factor $GV(\mathrm{x}_i, \mathrm{x}_{i+1}) = \mathrm{d}\omega^{\perp}_{i+1} / \mathrm{d}A(\mathrm{x}_i)$ in the presence of surface $\mathcal{M}_{\mathbf{c}}$. The change in differential projected solid angle relative to differential area is not affected by the occlusion because of $F_{\mathbf{c}}$, and only the visibility term $V$ is cancelled.

When we focus on the middle part of the path integral where $\mathrm{x}_1$ covers $\mathcal{M}_{\mathbf{c}}$, simplifications occur:

$$\begin{aligned} &\int_{\mathcal{M}_{\mathbf{c}}} GV(\mathrm{x}_0, \mathrm{x}_1) F_{\mathbf{c}}(\mathrm{x}_0, \mathrm{x}_1, \mathrm{x}_2) GV(\mathrm{x}_1, \mathrm{x}_2) \, \mathrm{d}A(\mathrm{x}_1) \\ &= \int_{\mathcal{M}_{\mathbf{c}}} \frac{\mathrm{d}\omega_1^{\perp}}{\mathrm{d}A(\mathrm{x}_0)} \frac{\delta(\omega_1 - \omega_2)}{\mathrm{d}\omega_1^{\perp} / \mathrm{d}\omega_1} \frac{\mathrm{d}\omega_2^{\perp}}{\mathrm{d}A(\mathrm{x}_1)} \, \mathrm{d}A(\mathrm{x}_1) \\ &= \int_{\mathcal{S}^2} \frac{\mathrm{d}\omega_2^{\perp}}{\mathrm{d}A(\mathrm{x}_0)} \delta(\omega_1 - \omega_2) \, \mathrm{d}\omega_1 = \; G(\mathrm{x}_0, \mathrm{x}_2) \overline{V}^1_{\mathbf{c}}(\mathrm{x}_0, \mathrm{x}_2) \, . \end{aligned} \tag{11}$$

Whereas the usual geometric factor $GV(x,y)$ rules the change of variables $(\mathrm{d}A, \mathcal{M}) \leftrightarrow (\mathrm{d}\omega^{\perp}, \mathcal{S}^2)$, the factor $G(x,y)\overline{V}^m_{\mathbf{c}}(x,y)$ results from the change of variables between $(\mathcal{M}_{\mathbf{c}})^m$ and $(\mathcal{S}^2)^m$ under the constraint of $F_{\mathbf{c}}$, with $m = 1$ here. The full definition of $\overline{V}^m_{\mathbf{c}}$ is

$$\overline{V}^m_{\mathbf{c}}(x,y) = \begin{cases} 1 & \text{if there are exactly } m \text{ occlusions between } x \\ & \text{and } y, \text{ and these occlusions are due to } \mathcal{M}_{\mathbf{c}}; \\ 0 & \text{otherwise.} \end{cases}$$

In summary, Equations (10) and (11) transform a path integral over vertices $\mathrm{x}_0, \mathrm{x}_1, \mathrm{x}_2$ where $\mathrm{x}_1 \in \mathcal{M}_{\mathbf{c}}$ and $\mathrm{x}_0, \mathrm{x}_2 \notin \mathcal{M}_{\mathbf{c}}$ into a path integral over vertices $\mathrm{x}_0, \mathrm{x}_2$. When considering paths of longer length in $\varphi^{-1}(\Omega \setminus \Omega_{\mathbf{C}} \cap \Omega_1)$, or when $k > 1$, the simplifications described in these equations apply in chain: a measurement over $\varphi^{-1}(\Omega \setminus \Omega_{\mathbf{C}} \cap \Omega_k)$ always transforms into a measurement over $\Omega \setminus \Omega_{\mathbf{C}} \cap \Omega_k$. We now need to show that the measurement contribution function changes from $f_{T,\mathbf{c}}$ to $f_{\mathbf{C}}$ between the integrals of Equation (9). Indeed, we can rework the integration domain of the left integral without changing its value, from $\varphi^{-1}(\Omega \setminus \Omega_{\mathbf{C}} \cap \Omega_k)$ to

$$\bigsqcup_{(m_1, \ldots, m_k) \in (\mathbb{N}^k)^*} (\mathcal{M} \setminus \mathcal{M}_{\mathbf{c}}) \times (\mathcal{M}_{\mathbf{c}})^{m_1} \times \ldots \times (\mathcal{M}_{\mathbf{c}})^{m_k} \times (\mathcal{M} \setminus \mathcal{M}_{\mathbf{c}})$$

where $m_i$ denotes the number of intersections that are removed between vertices $\mathrm{x}_{i-1}$ and $\mathrm{x}_i$ when $\varphi$ is applied. The integral over $\varphi^{-1}(\Omega \setminus \Omega_{\mathbf{C}} \cap \Omega_k)$ becomes a sum of integrals that simplify using the rules of Equations (10) and (11), and their domain turns into $\Omega \setminus \Omega_{\mathbf{C}} \cap \Omega_k$. As we add the measurement contribution functions under the integral, the visibility term factors out of the sum:

$$\sum_{(m_1, \ldots, m_k) \in (\mathbb{N}^k)^*} \prod_{i=1}^{k} \overline{V}^{m_i}_{\mathbf{c}}(\mathrm{x}_{i-1}, \mathrm{x}_i) \; = (1 - V_{\mathbf{c}} T_{\mathbf{c}}(\bar{\mathrm{x}})) \prod_{\mathbf{o} \neq \mathbf{c}} V_{\mathbf{o}} T_{\mathbf{o}}(\bar{\mathrm{x}}) \, .$$

Indeed, despite the summation over $(m_1, \ldots, m_k)$, only one tuple may yield a non-null product of $\overline{V}^{m_i}_{\mathbf{c}}$ depending on the occlusion of the path. This last equality between visibility factors boils down to rewording the definition of $\overline{V}^m_{\mathbf{c}}$.

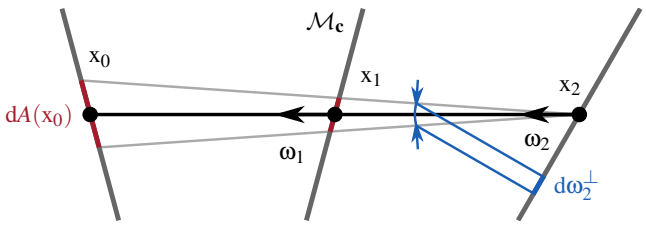



**Figure 10:** *Inside the geometry factor* $GV(\mathrm{x}_0, \mathrm{x}_2) = \mathrm{d}\omega_2^{\perp} / \mathrm{d}A(\mathrm{x}_0)$, *only the visibility* $V$ *is affected by surface* $\mathcal{M}_{\mathbf{c}}$.