Self-normalized estimators and joint path construction for efficient multi-view rendering

Basile Fraboni

▶ To cite this version:

Basile Fraboni. Self-normalized estimators and joint path construction for efficient multi-view rendering. Graphics [cs.GR]. Université de Lyon, 2021. English. NNT : 2021LYSEI103 . tel-03675200

HAL Id: tel-03675200
https://theses.hal.science/tel-03675200v1

Submitted on 23 May 2022

N°d'ordre NNT : 2021LYSEI103

THESE de DOCTORAT DE L'UNIVERSITE DE LYON
opérée au sein de
l'Institut National des Sciences Apliquées de Lyon

Ecole Doctorale EDA512
Informatique et Mathématiques de Lyon

Spécialité de doctorat : Informatique

Soutenue publiquement le 15/12/2021, par:

Basile Fraboni

Self-normalized estimators and joint path construction for efficient multi-view rendering

---

Devant le jury composé de:

George Drettakis	directeur de recherche	INRIA Sophia-Antipolis	Rapporteur
Nicolas Holzschuch	directeur de recherche	INRIA Grenoble	Rapporteur
Tamy Boubekeur	directeur de recherche	Adobe	Examinateur
Mathias Paulin	professeur	Université Paul Sabatier de Toulouse	Examinateur
Véronique Eglin	professeure	INSA de Lyon	Directrice de thèse
Jean-Claude Iehl	maître de conférences	Université Claude Bernard Lyon 1	Co-encadrant de thèse

Département FEDORA – INSA Lyon - Ecoles Doctorales

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	**CHIMIE DE LYON** **https://www.edchimie-lyon.fr** Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage secretariat@edchimie-lyon.fr	**M. Stéphane DANIELE** C2P2-CPE LYON-UMR 5265 Bâtiment F308, BP 2077 43 Boulevard du 11 novembre 1918 69616 Villeurbanne **directeur@edchimie-lyon.fr**
E.E.A.	**ÉLECTRONIQUE, ÉLECTROTECHNIQUE, AUTOMATIQUE** **https://edeea.universite-lyon.fr** Sec. : Stéphanie CAUVIN Bâtiment Direction INSA Lyon Tél : 04.72.43.71.70 secretariat.edeea@insa-lyon.fr	**M. Philippe DELACHARTRE** INSA LYON Laboratoire CREATIS Bâtiment Blaise Pascal, 7 avenue Jean Capelle 69621 Villeurbanne CEDEX Tél : 04.72.43.88.63 **philippe.delachartre@insa-lyon.fr**
E2M2	**ÉVOLUTION, ÉCOSYSTÈME, MICROBIOLOGIE, MODÉLISATION** **http://e2m2.universite-lyon.fr** Sec. : Sylvie ROBERJOT Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 secretariat.e2m2@univ-lyon1.fr	**M. Philippe NORMAND** Université Claude Bernard Lyon 1 UMR 5557 Lab. d'Ecologie Microbienne Bâtiment Mendel 43, boulevard du 11 Novembre 1918 69 622 Villeurbanne CEDEX **philippe.normand@univ-lyon1.fr**
EDISS	**INTERDISCIPLINAIRE SCIENCES-SANTÉ** **http://ediss.universite-lyon.fr** Sec. : Sylvie ROBERJOT Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 secretariat.ediss@univ-lyon1.fr	**Mme Sylvie RICARD-BLUM** Institut de Chimie et Biochimie Moléculaires et Supramoléculaires (ICBMS) - UMR 5246 CNRS - Université Lyon 1 Bâtiment Raulin - 2ème étage Nord 43 Boulevard du 11 novembre 1918 69622 Villeurbanne Cedex Tél : +33(0)4 72 44 82 32 **sylvie.ricard-blum@univ-lyon1.fr**
INFOMATHS	**INFORMATIQUE ET MATHÉMATIQUES** **http://edinfomaths.universite-lyon.fr** Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage Tél : 04.72.43.80.46 infomaths@univ-lyon1.fr	**M. Hamamache KHEDDOUCI** Université Claude Bernard Lyon 1 Bât. Nautibus 43, Boulevard du 11 novembre 1918 69 622 Villeurbanne Cedex France Tél : 04.72.44.83.69 **hamamache.kheddouci@univ-lyon1.fr**
Matériaux	**MATÉRIAUX DE LYON** **http://ed34.universite-lyon.fr** Sec. : Yann DE ORDENANA Tél : 04.72.18.62.44 yann.de-ordenana@ec-lyon.fr	**M. Stéphane BENAYOUN** Ecole Centrale de Lyon Laboratoire LTDS 36 avenue Guy de Collongue 69134 Ecully CEDEX Tél : 04.72.18.64.37 **stephane.benayoun@ec-lyon.fr**
MEGA	**MÉCANIQUE, ÉNERGÉTIQUE, GÉNIE CIVIL, ACOUSTIQUE** **http://edmega.universite-lyon.fr** Sec. : Stéphanie CAUVIN Tél : 04.72.43.71.70 Bâtiment Direction INSA Lyon mega@insa-lyon.fr	**M. Jocelyn BONJOUR** INSA Lyon Laboratoire CETHIL Bâtiment Sadi-Carnot 9, rue de la Physique 69621 Villeurbanne CEDEX **jocelyn.bonjour@insa-lyon.fr**
ScSo	**ScSo*** **https://edsciencessociales.universite-lyon.fr** Sec. : Mélina FAVETON INSA : J.Y. TOUSSAINT Tél : 04.78.69.77.79 melina.faveton@univ-lyon2.fr	**M. Christian MONTES** Université Lumière Lyon 2 86 Rue Pasteur 69365 Lyon CEDEX 07 **christian.montes@univ-lyon2.fr**

*ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

"Blessed are the cracked, for they shall let in the light."

Groucho Marx

Abstract

Physically based light transport simulation has progressively become the standard approach in the image production industry over the past few years. The Path Tracing algorithm and its variations are used to render photo-realistic images both for their theoretical simplicity and their accuracy to simulate complex lightning phenomenons.

However those simulations require exploring the whole set of light paths connecting a light source to the camera sensor and averaging their contributions in the associated pixels. Constructing such light paths is a complex sequential process since light can bounce around the scene several times before reaching the camera. Due to the stochastic nature of Monte Carlo path tracing we often need a large number of paths per pixel to reach acceptable levels of noise in final images. This is even more problematic when the integration domain gets higher in dimension, which is the case when adding production effects such as motion blur, depth of field and volumetric rendering.

Another important aspect is that we often have to generate more than one image of a scene, for instance when rendering stereo pairs to simulate binocular vision; lenticular images, light field images and holographic stereograms to simulate depth and parallax using multiple points of views; animated camera paths for virtual walkthroughs or classical animation rendering with motion blur.

In this dissertation, our goal is to accelerate the rendering of multiple viewpoint during a single simulation by exploiting redundancy across views. This poses a challenge as existing methods for sharing light paths across views introduce significant variance and–or bias, and are not well suited to render all production effects, materials, surfaces and participating media that a scene may include.

We start by introducing the path integral formulation of light transport, the associated Monte Carlo estimators, and the building blocks that forms the basis of path reusing applications. We then introduce a theoretical framework for multi-view rendering that rewrite each pixel integrals as a multi-strategy integrals with transformed samples. We show that to accurately reuse path samples across different pixel integrals we can not rely on classical estimators, that suffer from high variance, and that we need correctly defined path transformations, that currently can not handle volumetric light transport.

We hence introduce a new Monte Carlo estimator, which is biased but consistent, that combines multiple sampling strategies and automatically handles normalization issues which results in lower variance than classical estimators. Our new estimator allow us to accurately evaluate each pixel integral using a simple weighted average of sample contributions with specific weighting functions. We further extends the family of path transformations to allow path reusing in presence of participating media. Finally we develop a new unidirectional algorithm that readily extends state of the art volumetric path tracing to jointly render several frames at once. We introduce new methods to transform and reuse paths from one camera to the others and to generate subpaths that best contribute to a subset of views.

We demonstrate the effectiveness of our method at reducing noise at equal rendering time budget on several scenes with complex geometry and materials, participating media and production effects

Résumé

Au cours des dernières années, la simulations du transport de lumière physiquement réaliste est progressivement devenue l'approche standard dans l'industrie de la production d'images. L'algorithme de *Path Tracing* et ses variantes sont utilisés pour rendre des images photoréalistes, tant pour leur simplicité théorique et d'implémentation que pour leur capacité à simuler des phénomènes d'éclairage complexe.

Cependant, ces simulations nécessitent d'explorer l'ensemble des chemins reliant une source lumineuse au capteur de la caméra et de calculer la moyenne de leurs contributions dans les pixels associés. La construction de ces chemins est un processus séquentiel complexe puisque la lumière peut rebondir plusieurs fois dans la scène avant d'atteindre la caméra. En raison de la nature stochastique des estimateurs de Monte Carlo utilisés en *Path Tracing*, il nécéssite souvent de construire et évaluer un très grand nombre de chemins par pixel pour atteindre des niveaux de bruit acceptables dans les images. Ceci est encore plus problématique lorsque les dimensions du domaine d'intégration augmentent. Ce qui est le cas lorsque l'on ajoute des effets de production tels que le flou de mouvement, la profondeur de champ et le rendu volumétrique.

Un autre aspect important est que nous devons souvent calculer plus d'une image d'une scène, par exemple lors du rendu de paires stéréo pour simuler la vision binoculaire ; d'images lenticulaires, d'images de champs lumineux (*lightfield*) et de stéréogrammes holographiques pour simuler et visualiser la profondeur et la parallaxe en utilisant plusieurs points de vue ; de trajectoires de caméra animées pour les visites virtuelles ou le rendu d'animation classique avec flou de mouvement.

Dans cette thèse, notre objectif est d'accélérer le rendu de plusieurs points de vue lors d'une même simulation en exploitant la cohérence entre les caméras. Ceci représente un défi car les méthodes existantes pour réutiliser les chemins entre plusieurs vues introduisent du biais visible dans les images, et ne sont pas adaptées pour tous les effets de production, les matériaux, les surfaces et les volumes qu'une scène peut inclure.

Nous commençons par présenter la formulation intégrale du transport de lumière, les estimateurs de Monte Carlo associés, et les blocs de construction qui constituent la base des applications de réutilisation de chemin. Nous introduisons ensuite un cadre théorique pour le rendu multi-vues qui exprime l'intégrale d'un pixel comme une intégrale multi-stratégie avec des transformation d'échantillons. Nous montrons que pour réutiliser correctement les chemins échantillonnés dans différents pixels, nous ne pouvons pas nous reposer sur les estimateurs classiques, qui souffrent d'une variance élevée, et que nous avons besoin de transformations de chemin correctement définies, qui ne peuvent actuellement pas gérer le transport de lumière dans les milieux participants.

Nous introduisons donc un nouvel estimateur de Monte Carlo, biaisé mais cohérent, qui combine plusieurs stratégies d'échantillonnage et traite automatiquement les problèmes de normalisation, ce qui se traduit par une variance inférieure à celle des estimateurs classiques. Notre nouvel estimateur nous permet d'évaluer avec précision et sans biais visible l'intégrale de chaque pixel en utilisant une simple moyenne pondérée des contributions avec des fonctions de pondération spécifiques. Enfin, nous développons un nouvel algorithme unidirectionnel qui étend directement l'état de l'art en *Path Tracing* volumétrique pour rendre conjointement plusieurs images.

Nous introduisons de nouvelles méthodes pour transformer et réutiliser le chemin d'une caméra à l'autre en présence de milieux participants et pour générer des sous-chemins qui contribuent le mieux à un sous-ensemble d'observateur.

Nous démontrons sur plusieurs scènes comprenant de la géométrie complexe, des matériaux complexes, des milieux participants et des effets de production que cette méthode réduit efficacement le bruit par rapport aux calculs image par image à temps de calcul équivalent.

Contents

Chapter 1

Introduction

1.1 The multi-view problem

Physically based light transport simulations have progressively become the standard approaches in the image production industry over the past few years. The *Path Tracing* algorithm and its variations are used to render photo-realistic images both for their theoretical and practical simplicity and their accuracy to simulate complex lightning phenomenons such as depth of field, soft shadows, motion blur, specular caustics, participating media, spectral effects like diffraction, and much more.

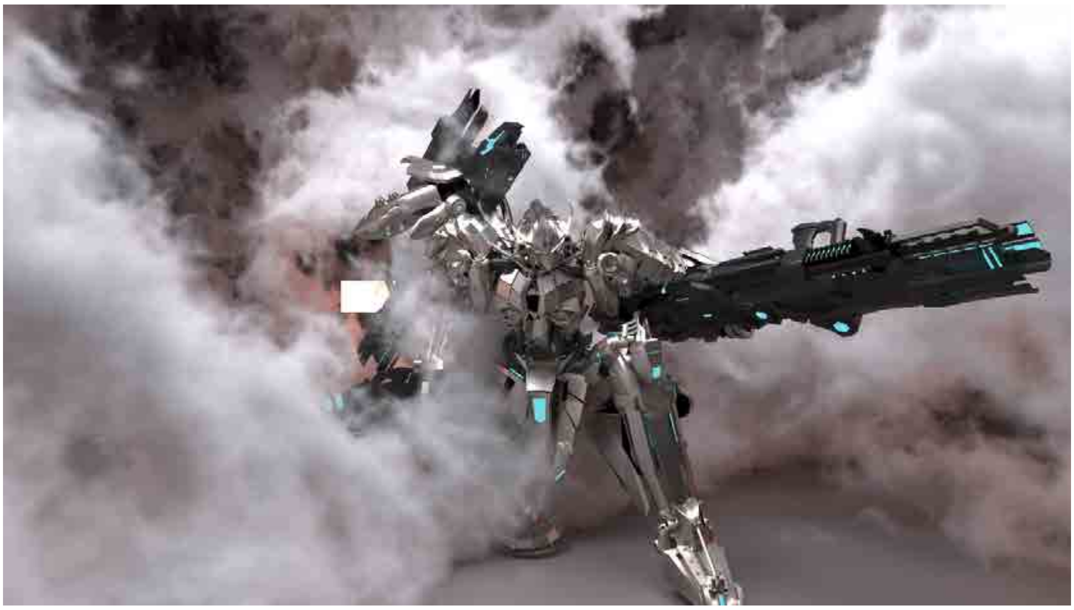


FIGURE 1.1 – Example of a complex scene composed of a dense procedural heterogeneous medium, a 166k triangles model [1] and 3600 light sources rendered in 88 hours. Note how the fine details in the clouds enhance the realism of the scene at the cost of an overlong rendering time.

However those simulations require exploring the whole set of light paths connecting a light source to the camera sensor and averaging their contributions in the associated pixels. Constructing such light paths is a complex sequential process since light can bounce around the scene several times before reaching the camera. To remain tractable the path construction involves Monte Carlo methods and random sampling

---

[1] Original model courtesy of Zeroswat made available under the CC-BY license.

to accurately estimate the integral of the radiance reaching every pixels with a finite number of samples. Due to the stochastic nature of Monte Carlo path tracing we often need a large number of paths per pixel to reach acceptable levels of noise in final images. This is even more problematic when the integration domain gets higher in dimension, which is the case when adding production effects such as motion blur, depth of field and volumetric rendering. For these reasons, rendering photo realistic images is a time consuming operation as illustrated in Figure 1.2.

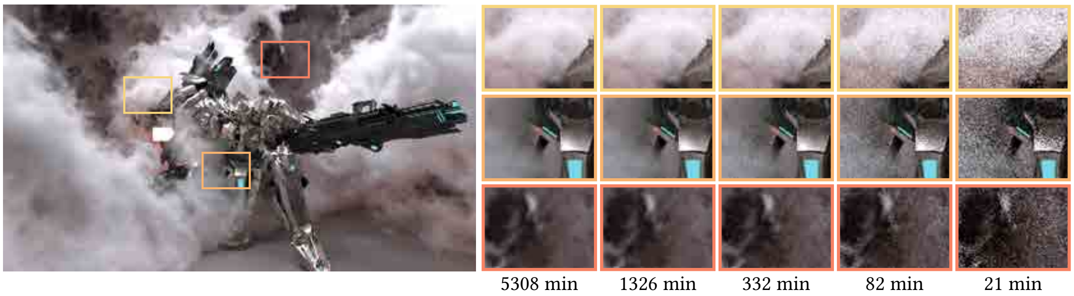


FIGURE 1.2 – Close up details of the scene from Figure 1.1 at different rendering times. A desirable quality cannot always be reached in reasonable time budget.

Another important aspect is that we often have to generate more than one image of a scene, for instance when rendering stereo pairs to simulate binocular vision; lenticular images, light field images and holographic stereograms to simulate depth and parallax using multiple points of views; animated camera paths for virtual walkthroughs or classical animation rendering with motion blur. These setups can all benefit of path reuse since all cameras are close in the spatio-temporal domain, thus are likely to share some parts of the light path space. Although there is a strong coherency between observed regions, the method used so far is to render frame after frame for practical reasons: lack of robust multi-view estimators, ease of distributed computing, and ease of implementation.

In this dissertation, our goal is to accelerate the rendering of multiple viewpoint during a single simulation by exploiting redundancy across views. This poses a challenge as existing methods for reusing light paths across views introduce significant variance and/or bias. Furthermore they are not well suited to render all production effects, materials, surfaces and participating media that a scene may include.

1.2 Overview of thesis contributions

Our contributions cover new statistical estimators that are beneficial to reduce variance, new path sampling strategies and a new application pipeline for multiple view rendering (published as [Fraboni et al. 2019, 2022]).

1.2.1 Previous work

We start this dissertation by introducing the path integral framework and its recent extensions for volume rendering in chapter 2, which is essential to allow path reuse in presence of participating media. We then present the classical unbiased Monte Carlo estimators used in rendering applications in chapter 3, which have limitations in some situations as we shall see in chapter 6. Next, we review the common path construction methods used in unidirectional path tracing in chapter 4 that generate interactions in path space. Finally, we present the common path transformations methods in chapter 5 that modify some path interactions to allow path reuse in rendering applications. This concludes the first part of the thesis covering previous approaches. We invite the readers that are familiar with some of these topics to selectively skip these chapters.

1.2.2 The multi view problem

The second part of this dissertation starts by introducing a theoretical framework for multi-view rendering and exposing the main challenges in chapter 6. We define an extended path space and decompose each pixel integral to several subdomains integrals with multiple sampling techniques on each subdomain. Our formulation is derived from the path integral formulation of light transport introduced in chapter 2.

Effectively reducing the variance by combining several sampling techniques requires robust multi-strategy estimators such as Multiple Importance Sampling (MIS) [Veach and Guibas 1995] or its derived versions. Some former multiple view rendering methods neglected such weighting combinations [Henrich et al. 2011, Havran et al. 2003, Adelson and Hodges 1995], on the contrary others methods demonstrated that MIS is efficient to optimally combine each sample in the multi view context [Méndez Feliu et al. 2006, Schwarzhaupt 2019]. However, as we shall see MIS and derived techniques perform poorly when the number of contributing strategies is unknown a priori. This results in a problematic additional variance that cannot be addressed with classical unbiased estimators, motivating the introduction of new estimators that overcome these limitations in chapter 7.

Additionally, we show that to accurately reuse samples from one pixel to another some bijective path transformations have to be used, similar to the ones presented in chapter 5. However, path transformations have never been extended to handle heterogeneous participating media, motivating the introduction of new transformations in chapter 8.

1.2.3 A self normalized multi-strategy estimator

Reusing path samples from different pixels poses certain difficulties in Monte Carlo integration. The amount of paths built from every pixels is consequent but only few can be transformed to reach a given pixel.

We demonstrate that unbiased multi strategy Monte Carlo estimators (MIS – CMIS) [Veach and Guibas 1995, West et al. 2020] exhibit high variance when the number of contributing strategies to a pixel is not known a priori. In fact, the more we add paths that gather zero contributions to the estimator, the more the variance increases. MIS and CMIS require knowing the total number of samples drawn for each pixel to correctly normalize each pixel estimate.

Furthermore in the context of multi view rendering the number of samples per pixel is variable depending on how many other pixels share some visible areas – volumes in the scene. Using the naive assumption that every pixel could contribute to every other pixels lead to a huge amount of variance since the renormalization takes into account all samples drawn on every pixels. Explicitly estimating the contributing pixels informations for each pixel allows to compute a better estimate of the normalization but results in a biased but consistent estimator.

We hence introduce in chapter 7 a new Monte Carlo estimator, which is biased but consistent, that combines multiple sampling strategies and automatically handles this normalization issue [Bekaert et al. 2000, Owen 2013] and leads to a considerable variance reduction. Our final estimator reduces to a weighted sum that is simple to compute and does not require additional storage than a 4-channel image for each camera. In addition, we show that available bias-correction techniques [Talbot et al. 2005, McLeish 2011, Booth 2007, Bitterli et al. 2020] are inappropriate for the multi-view problem.

1.2.4 Path transformations for heterogeneous volumes

Next, we introduce new methods to transform and reuse paths that traverses arbitrary media in chapter 8. Our new mappings build upon the theory of null scattering [Miller et al. 2019] and can be evaluated in closed form. And their Jacobian determinants required to take into account the change in density are simple to compute. Our mappings allow to extend any path reuse application to handle arbitrary participating media.

1.2.5 Practical multi view rendering

Finally, we develop a new unidirectional algorithm that readily extends state of the art volumetric path tracing to jointly render several frames at once in chapter 9. Our multi-view path construction build upon our new transformations and the contributions are accumulated using our new estimator. Additionally we propose a new similarity heuristic to compare on-the-fly arbitrary scattering distributions in order to generate subpaths that best contribute to a subset of pixels.

We implement our variant of multiple view rendering in a custom renderer, which is also the first path reusing method that handles heterogeneous participating media. We demonstrate the efficiency of our method on several test scenes with complex geometry and materials, participating media and production effects in chapter 10. Our results exhibit less noise at equal rendering time budget than frame by frame path tracing.

Chapter 2

The path integral framework

We start this dissertation with an introduction to the classical formulations of light transport simulations. We review light integrals in presence of surfaces and media in their path space form that defines the theoretical basis of unidirectional rendering algorithms. We do not cover the operator formulations as well as adjoint and bidirectional methods since they are out of the scope of this dissertation [Veach 1998].

Some recommended references for an overview are the excellent Eric Veach thesis that formalizes surfaces integrals [Veach 1998], the recent state-of-the-art report on volumetric light transport simulations [Novák et al. 2018], and finally the extended null-scattering framework introduced in [Miller et al. 2019]. Additionally recent courses on path tracing provide excellent materials on novel techniques [Fascione et al. 2019, Jakob et al. 2019, Hill et al. 2020]. For an introductory book about radiometry and light transport from the physical point of view see [McCluney 2014]. Last but not least, the book *Physically based rendering: From theory to implementation* by Pharr et al. is a good introduction from both technical and theoretical points of view.

2.1 Extended path integral formulation: Surface, Volumes, Null scattering

The path integral formulation of light transport [Kajiya 1986, Veach 1998], its extension to participating media [Pauly et al. 2000] and null scattering [Miller et al. 2019] formulate the estimate of a pixel j as the integral of all the light contributions reaching the pixel:

$$I_j = \int_{\Omega} f_j(\bar{x}) \mathrm{d}\mu(\bar{x}) \tag{2.1}$$

where a path $\bar{x}$ is a finite sequence of vertices in space $\{x_0, \ldots, x_k\}$ starting from the camera aperture (or lens in simple models) and reaching a light source, the integration domain Ω is the union of all paths of finite lengths ($\Omega = \bigcup_{i=1}^{\infty} \Omega_i$), the path measure $\mathrm{d}\mu$ is the product of the differential measures at each path vertex, and $f_j(x)$ is the *measurement contribution function* of the path.

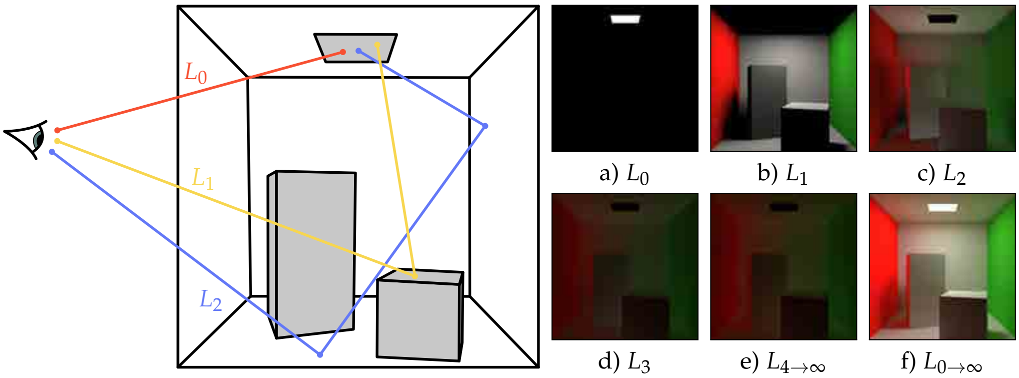
We review the terms of the pixel integral and the measurement contribution function in the following sections. In addition, we illustrate several configurations in Figure 2.5 to show the evolution of the path integral framework. We do not cover all aspects in depth but rather propose a synthesized view of the current state of the rendering framework supported by a rich literature.

2.1.1 Path length

A path of length k is comprised of $k+1$ vertices where $k-1$ of them are surface or volume interactions in the scene, the first is a lens interaction and the last is a light source interaction. The sensor point – i.e. on the film plane – is usually not accounted in the path length since it is a pure virtual point and the lens is considered the first real surface interaction in the scene. Camera sample paths can therefore be classified by length, further illustrated in Figure 2.1:

- a path of length $k=1$ connects the camera exit point to a light source directly and its contribution is denoted L_0,
- a path of length $k=2$ connects the camera exit point, to a scene interaction and then to a light source and its contribution is denoted L_1 or *direct lighting*,
- a path of length $k=3$ will be composed of a lens interaction, two interactions in the scene and a light interaction and its contribution is denoted L_2 or *indirect lighting*,
- and correspondingly for longer paths, which are also accounted in the *indirect lighting*.

The path integral hence gathers contributions from paths of every length to accurately simulate the light propagation in the real world.



a) L_0 b) L_1 c) L_2

d) L_3 e) $L_{4\to\infty}$ f) $L_{0\to\infty}$

FIGURE 2.1 – Several contribution layers of a diffuse Cornell Box lit by a light source: from left to right the contribution layer of paths of length a) $k=1$ – visible light source, b) $k=2$ – direct lighting, c) $k=3$ – indirect 2 bounces, d) $k=4$ – indirect 3 bounces, e) $k=\{5,\ldots,\infty\}$ – indirect 4 bounces and more. The complete global illumination f) is the sum of all layers.

2.1.2 Measurement contribution function

The measurement contribution function evaluates the light contribution for a given path $\bar{x}$. This function is a multidimensionnal integrand – since a path is multidimensionnal – which decompose in a product of terms modeling light scattering:

$$\begin{aligned} f_j(\bar{x}) =& W_j(x_0,x_1)G(x_0,x_1)T(x_0,x_1)L_e(x_{k-1},x_k)\cdot \\ & \left[\prod_{i=1}^{k-1} f_s(x_{i-1},x_i,x_{i+1})G(x_i,x_{i+1})T(x_i,x_{i+1})\right] \end{aligned} \tag{2.2}$$

where W_j is the camera importance function w.r.t pixel j, L_e is the emitted radiance of the light vertex x_k, T is the transmittance and G is the geometric terms on every path segment, and f_s is the *bidirectional scattering distribution function* (BSDF) at each non terminal vertex. We then present each term of the integrand in the following sections.

2.1.3 Camera importance function

A camera is a virtual object that is defined by a set of parameters such as the position, orientation, exposure interval, lens system, focal distance, field of view, trajectory, etc. The camera *sensor* or *film* is a virtual pixel matrix positioned in space. A light ray that traverses the lens system of a camera reaches the sensor with a certain density which is accounted by the camera importance function.

Therefore the camera importance function W_j hides the sensor response of the camera and a filter function over the exposure window (temporal) and over the image plane w.r.t pixel j (spatial).

The sensor response is usually baked [1] in a smart way to cancel out terms that arise after sampling primary rays that finds the first hit point on the scene and geometric terms [Kolb et al. 1995, Hanika and Dachsbacher 2014]. This is also necessary for adjoint and bidirectional methods to correctly compare the path differential measures of different sampling techniques [Veach 1998].

2.1.4 Participating media

A participating medium is a volume filled with microscopic particles which absorb and scatter light particles. We model these volumes using statistical descriptors which represent the probability densities of photons to absorb or scatter per unit distance traversed through the volume:

- the absorption coefficient μ_a,
- the scattering coefficient μ_s,
- the extinction coefficient μ_t, defined as $\mu_t = \mu_a + \mu_s$.

A medium is said to be *homogeneous* if these coefficients are spatially constant, and *heterogeneous* if they are spatially varying. In addition, these coefficients may vary depending on the wavelength to simulate wavelet dependent phenomenons, for example to simulate the color bleed from blue to red in earth atmospheric scattering or the colors in rainbows [Sadeghi et al. 2012].

Transmittance

The light propagation in participating media is described by the *radiative transfer equation* [Chandrasekhar 1960], which takes into account four processes: absorption, out-scattering, in-scattering and emission due to the particles encountered along a straight line in the volume. The two former processes induce a loss of radiance along the ray, whereas the latter two induce a gain of energy. The transmittance measures the light attenuation along a straight line through media due to the absorption and

---

[1] The camera importance function derivation given in [Pharr et al. 2016, section 16.1.1 equation 16.4] seems to be using the wrong cosine term. For a corrected version please see Appendix A.

out scattering processes [Lambert 1760], and writes:

$$T(x,y) = e^{-\int_{t_x}^{t_y} \mu_t(x_t)\mathrm{d}t} = e^{-\tau(x,y)} \qquad \text{where} \qquad x_t = x + t \cdot \omega_{x\to y} \tag{2.3}$$

This models the light attenuation considering uncorrelated particles in the medium, hence producing an exponentially decreasing distribution through the medium. However it is possible to handle non-exponential models [Bitterli et al. 2018]. The integral of the extinction coefficient along the segment is called the *optical thickness* denoted τ. Although the optical thickness does not have a closed-form for spatially varying extinctions, the transmittance of homogeneous volumes can be solved as follows:

$$T(x,y) = e^{-\mu_t \cdot \left(t_y - t_x\right)}. \tag{2.4}$$

To evaluate transmittance in heterogeneous media we rely on estimators of the transmittance that can be biased (e.g. ray marching) or unbiased (e.g. delta tracking and derivated methods). We invite the reader to refer to [Novák et al. 2018] for a detailed survey of radiative transfer equations and existing transmittance estimators. In addition more recent works propose several new transmittance estimators based on new integral formulations of the radiative transfer equation that are more accurate and converge faster in some situations [Georgiev et al. 2019, Kettunen et al. 2021].

We illustrate various exponential transmittance profiles with constant and varying extinction coefficients in Figure 2.2.

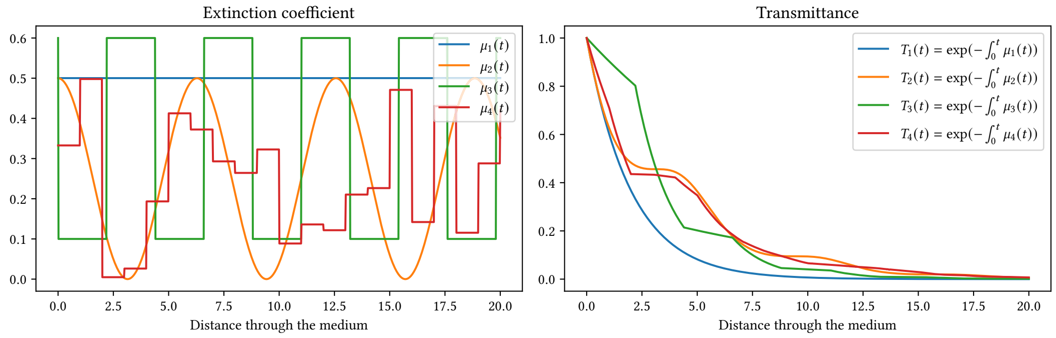


FIGURE 2.2 – We plot several extinction coefficients (left) and their associated transmittance (right) in 1D. Note that constant extinction (i.e. homogeneous medium – blue curve) has a perfect decreasing exponential transmittance. On the contrary spatially varying extinctions (i.e. heterogeneous media – green, orange and red curves) have non trivial transmittance profiles.

Null-scattering framework

Until recently, no path formulation provided the closed-form expressions associated to the transmittance of ordinary heterogeneous participating media. Miller et al. defined a *Null scattering framework* that extends the general path integral formulation to explicitly take into account fictitious (null) particles in media. This technique differs from previous unbiased methods by explicitly accounting and sampling null particles along a straight line between two real scattering interactions (see [Novák et al. 2018, sect. 4.2] for a survey of preceding unbiased methods). Two new statistical quantities are added to the model:

- the combined extinction coefficient $\bar{\mu}$, or majorant of the medium, which bounds the extinction coefficient μ_t,
- the null scattering extinction coefficient μ_n, defined as $\mu_n = \bar{\mu} - \mu_t$.

The transmittance in the real heterogeneous medium, in general impossible to express in closed-form, is replaced by the *combined transmittance* of an homogenized medium which has a closed-form expression:

$$\bar{T}(x,y) = e^{-\bar{\mu}\cdot(t_y - t_x)}. \tag{2.5}$$

The path sampling process then includes explicit scattering event sampling (e.g. null, absorption or scattering interaction) at each step in the medium and enable analytical evaluation of the combined transmittance $\bar{T}$ and paths pdfs between paths vertices on a volume segment. A path of length k has then m real interactions (see in Figure 2.1), plus several null interaction on medium segments. Note that when $\mu_n = 0$ this formulation reduces to the classical exponential form. For example in presence of homogeneous media by setting $\bar{\mu} = \mu_t$, it falls back to the closed-form sampling and evaluation process (i.e. with only real interactions). Additionnaly the null scattering framework naturally allows combining different sampling techniques in spatially and spectrally varying media.

2.1.5 Bidirectional scattering distribution function

The bidirectional scattering distribution function (BSDF) describes the directional scattering model at each non terminal interaction of the path. The BSDF quantifies the fraction of light propagated at a path vertex as a function of the incident direction ω_i – from where comes the light – and the outgoing direction – towards which the light is scattered, see Figure 2.3 for a picture of several BSDF models.

At each interaction of a path, the scattering model is expressed with the generalized BSDF formulation including surface, volume and null scattering, which writes:

$$f_s(x,y,z) = \begin{cases} f_r(\omega_{y\to x}, \omega_{y\to z}) & \text{if } y \text{ is on a surface,} \\ \mu_s(y) f_p(\omega_{y\to x}, \omega_{y\to z}) & \text{if } y \text{ is a scattering medium interaction,} \\ \mu_n(y)\delta(\omega_{y\to x}, \omega_{y\to z}) & \text{if } y \text{ is a null medium interaction.} \end{cases} \tag{2.6}$$

Properties of a BSDF

A correctly defined BSDF should respect three main properties:

1. Positivity: $f_s(\omega_i, \omega_o) \geq 0$.
2. Energy conservation: $\int_\Omega f_s(\omega_i, \omega_o) \cos\theta_i \mathrm{d}\omega_i \leq 1$. Note that for an ideal energy conserving model the total scattered energy should be equal to the total received energy. However several BSDF models only consider a single scattering interaction – i.e. does not take into account inter-reflections at the micro scale of the matter or particle – which may result in a loss of energy. Since losing energy is tolerable in practice, a BSDF model adding energy is a counter physical behavior and must be avoided since emissions are taken into account separately (see subsection 2.1.6). The problem of energy loss is addressed with multiple scattering models [Heitz et al. 2016] or precomputed energy compensation for single scattering models [Turquin 2017, Conty and Kulla 2017].

3. Reciprocity: $f_s(\omega_i, \omega_o) = f_s(\omega_o, \omega_i)$. Reciprocity is a physical property of scattering models and is required for BSDF models to be usable in adjoint and bidirectional methods [Veach 1998, chap. 3.6 and chap 5.]. Although most BSDF models are reciprocal by construction, this condition can be reasonably violated in unidirectional rendering algorithms since reciprocity is not a strong requirement. For example the multiple scattering model proposed in [Turquin 2017] is voluntarily not reciprocal, since the author focused its work on the shape of the multiple scattering lobe rather than on its reciprocity.

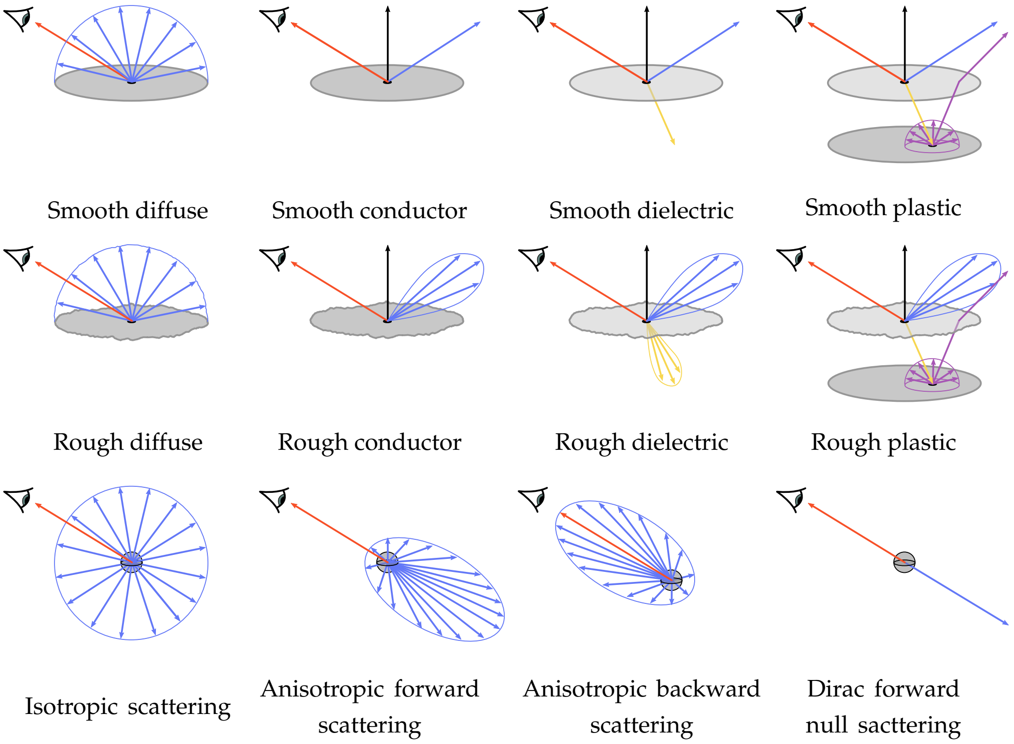

Smooth diffuse
Smooth conductor
Smooth dielectric
Smooth plastic
Rough diffuse
Rough conductor
Rough dielectric
Rough plastic
Isotropic scattering
Anisotropic forward scattering
Anisotropic backward scattering
Dirac forward null sacttering

FIGURE 2.3 – Examples of common BSDF models used in computer graphics depicted from top to bottom: smooth surface models (top), rough surface models (center), and medium phase functions (bottom).

Surface BSDF

On surfaces the scattering function describes the reflection and – or transmission model of the surface material. This BSDF is defined over the unit hemisphere centered around the normal of the path interaction. A very wide family of models have been developed over the years (analytical, numerical or measured), for a non-exhaustive list with comparisons see [Guarnera et al. 2016, table 1] and [Montes and Ureña 2012, table 1, figure 2], and for a taxonomy of BSDF see [McGuire et al. 2020].

The most common models include: diffuse Lambertian models, Dirac distributions for perfectly specular surfaces (i.e. smooth conductors and dielectrics), rough diffuse models [Oren and Nayar 1994, Heitz and Dupuy 2015, d'Eon 2021], glossy models for rough surface through the micro-facet theory [Cook and Torrance 1982] such as the GGX distribution [Walter et al. 2007, Heitz 2014, Heitz and d'Eon 2014, Heitz 2018], models accounting for multiple scattering at the micro scale [Heitz et al. 2016, Turquin 2017, Conty and Kulla 2017], models for multi layered materials for richer appearances [Jakob et al. 2014, Guo et al. 2018, Xia et al. 2020, Gamboa et al. 2020, Belcour 2018] – e.g. one or several coatings, one substrate, and eventually media in between.

Phase functions

Inside media the scattering function, called the *phase function*, is defined over the unit sphere around the path interaction. As for surfaces, it models the directional distribution of scattered rays ω_i leaving a real medium particle when the particle is observed from a given direction ω_o.

Several scattering models exists, such as: the isotropic phase function which is modeled by a uniform directional distribution over the unit sphere, the Rayleigh phase function which describes atmospheric scattering in presence of particles that have a small radius w.r.t to the wavelengths of the light [Rayleigh 1871], the Lorenz-Mie phase function for realistic cloud simulation in presence of spherical droplets with radiis of the same order than the wavelength [Lorenz 1890, Mie 1908], the simpler Henyey-Greenstein scattering model [Henyey and Greenstein 1941] which has a simple closed-form parameterized by the mean cosine parameter $g \in [-1, 1]$. The latter parameter defines the shape of the directional distribution: $g = 0$, $g > 0$ and $g < 0$ corresponds respectively to isotropic scattering, forward scattering and backward scattering (see bottom row in Figure 2.3). Mixture of simple phase functions – e.g. Henyey-Greenstein – can be used to model richer volume appearances [Gkioulekas et al. 2013, Sharma 2015]. We illustrate several phase function profiles in Figure 2.4. Several other models exist and some of them are for example developed in astrophysics to describes scattering phenomenons in the universe, simpler ones used in computer graphics for simulation or visualization, see [Pegoraro 2016] and [d'Eon 2016] for additional materials regarding phase functions. Note that in media, the phase function is weighted by the scattering coefficient of the medium μ_s.

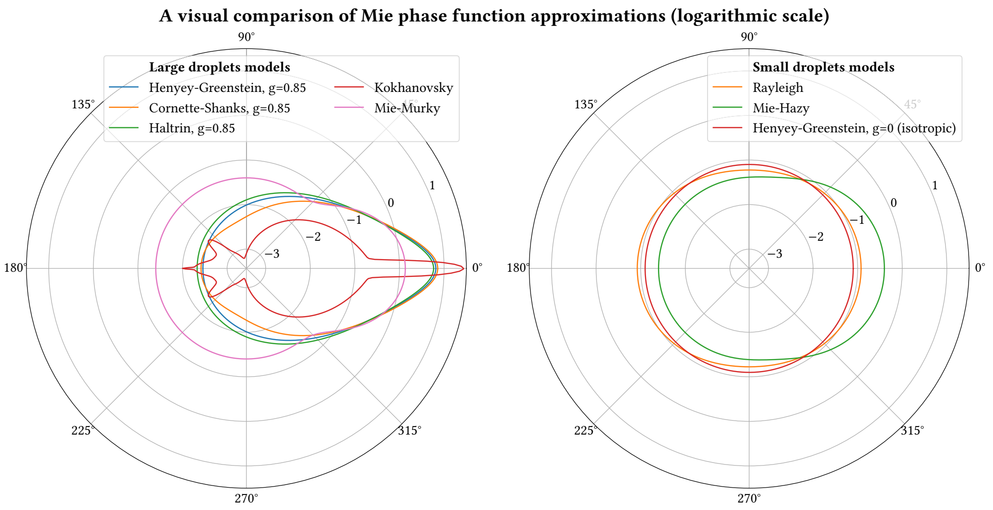


FIGURE 2.4 – Several phase functions profiles approximated from the Mie theory to simulate cloud-sized spherical droplets (left plots) and haze-sized spherical droplets (right plots).

Null scattering singularity

A last form due to the null-scattering framework describes the scattering model at null interactions in the medium. Since null events model fictitious particles – i.e. where no scattering happens – which are constrained on a straight line between two real scattering vertices, the light propagation direction remains unchanged. Therefore the resulting scattering function is a Dirac delta distribution[2] in the forward direction (see bottom right in Figure 2.3) weighted by the null scattering extinction coefficient μ_n.

Discussion

The BSDF is at the core of rendering applications to accurately simulate light scattering in a scene. As mentioned before, a BSDF is parameterized by the observing direction, hence is often a *view-dependent* distribution. As we shall see in chapter 9, comparing BSDF with identical parameters but different outgoing directions is a key component in the context of multi-view rendering. In fact reusing a section of a path that has been sampled w.r.t to an outgoing direction $\omega_{o,1}$, does not necessarily contributes to another observing direction $\omega_{o,2}$ if the distribution is strongly view-dependent.

2.1.6 Emitted radiance

Light sources are essentials in a scene in order to illuminate visible objects. The emitted radiance L_e from a path vertex y towards another vertex x is expressed differently depending if the emitting vertex is on a surface or in a medium. In the latter case, the attenuation due to the volume absorption μ_a is taken into account.

$$L_e(x,y) = \begin{cases} L_e(x,y) & \text{if } y \text{ is on a surface,} \\ \mu_a(y)L_e(x,y) & \text{if } y \text{ is an absorbing medium interaction.} \end{cases} \tag{2.7}$$

Several models of light sources have been developed to model different light behaviors such as: area lights, point lights, spot light, infinite lights (i.e. sun), image based lights, measured lights (i.e. environment lighting) and blackbody emitters (i.e. stars). For further details see [Pharr et al. 2016, chap. 12].

2.1.7 Geometric term

The geometric term on each path segment takes into account the visibility V between segment endpoints, the light attenuation due to the traveled distance (the inverse squared falloff) and the orientation with respect to the geometric normal of surfaces (the cosine between the normal and the segment normalized direction).

---

[2] In the original article Miller et al. use a heavyside function to enforce the ordering of null vertices along the null scattering chain. However since the chain is constrained between two real scattering events, the only possible scattering direction is the forward direction. For that reason we prefer encoding the null vertices ordering inside the path measure and use the Dirac to denote the only possible scattering direction at a null interaction.

$$\begin{aligned} G(x,y) &= V(x,y)\cdot\frac{D(x,y)D(y,x)}{\|x-y\|^2} \quad \text{where} \\ D(x,y) &= \begin{cases} |n(x)\cdot\omega_{x\to y}| & \text{if } x \text{ is on a surface with } n(x) \text{ its normal,} \\ 1 & \text{if } x \text{ is in a medium.} \end{cases} \\ V(x,y) &= \begin{cases} 1 & \text{if } x \text{ and } y \text{ are mutually visible,} \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{2.8}$$

The light attenuation comes from the change of measure between differential solid angle to differential area measure [Veach 1998, eq. 8.2] [Pharr et al. 2016, eq. 5.6].

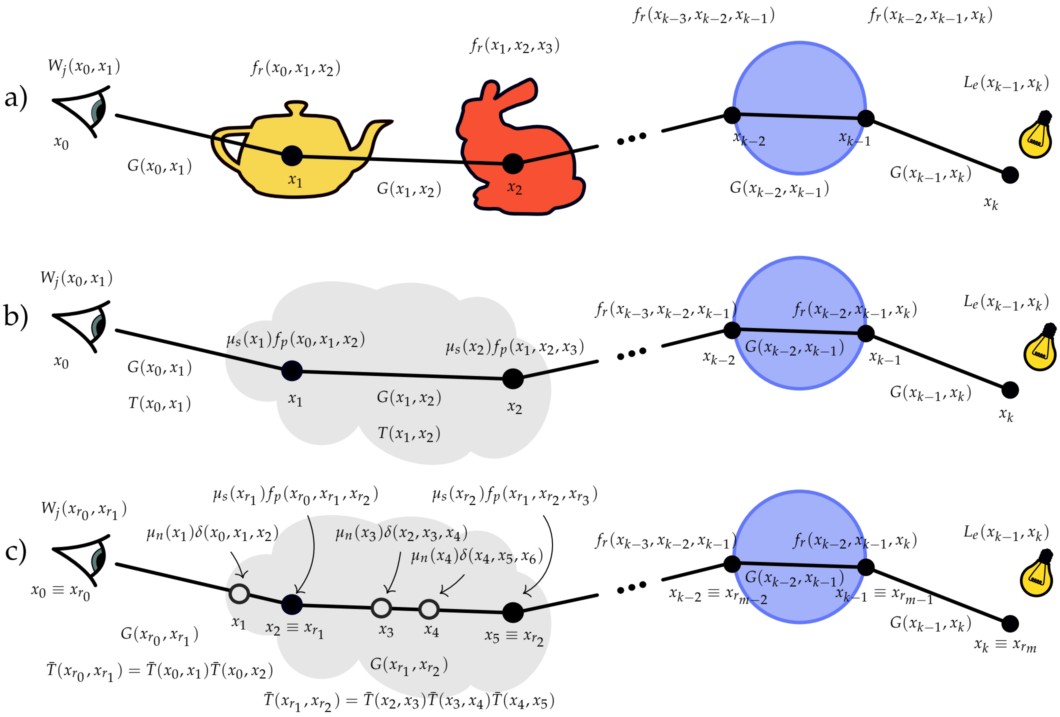


FIGURE 2.5 – Path space terms of the path integral formulation within a) the surface framework [Veach 1998], b) the surface and real medium framework [Pauly et al. 2000], and c) the null scattering framework [Miller et al. 2019]. It is worth noting that null scattering interactions (x_1, x_3, x_4) in figure c) are taken into account explicitly in the path. Thus a path of k interactions has only m real interactions denoted by the subscript r.

2.1.8 The path space measure

The extended path integral framework includes three different types of interactions along the path:

- Surface interactions in the surface space denoted A,
- Volume interactions in the volume space denoted V,
- Null interactions in the null scattering volume space denoted V_n.

Note that V_n is simply a copy of V since null scattering interactions may happen anywhere some volume is defined. We can thus rewrite the path space definition as:

$$\begin{aligned}\Omega &= \bigcup_{i=1}^{\infty} \Omega_i \\ &= \bigcup_{i=1}^{\infty} (A \cup V \cup V_n)^{i+1}\end{aligned} \tag{2.9}$$

with i denoting the path length (number of segments on a path). The path differential measure is the product of the differential measures at each vertex that constitutes the path. Therefore the differential measure of a path $\bar{x} = \{x_0, \ldots, x_k\}$ writes:

$$\mathrm{d}\mu(\bar{x}) = \prod_{i=0}^{k} \begin{cases} \mathrm{d}A(x_i) & \text{if } x_i \text{ is on a surface,} \\ \mathrm{d}V(x_i) & \text{if } x_i \text{ is a real medium interaction,} \\ \mathrm{d}V_n(x_i) & \text{if } x_i \text{ is a null medium interaction.} \end{cases} \tag{2.10}$$

It is worth noting that an interaction or a chain of interactions of a path may be generated by different construction techniques resulting in the same associated measure but in different probability densities. The fact that several construction strategies are available for the same samples lets us select or weight them in a way such that they best contribute to our integration problem.

2.2 Conclusion

In this section, we detailed the framework for evaluating the contribution of a light path in a scene (illustrated in Figure 2.5). There are still two missing pieces to estimate a pixel value: generating light paths and integrating their contributions.

Integral estimation. In fact, rendering an image involves estimating this integral for each pixel of the camera. Monte Carlo estimators are convenient to get accurate estimates of high dimensional integrals since their convergence does not depend on the dimensionality of the integrand. As we shall see in chapter 3, these methods rely on probability theory and random sampling of the integration domain.

Path construction. Random construction of light paths is often a sequential approach. We review in detail the path sampling methods used in a path tracer in chapter 4.

Path transformations. Constructing paths is expensive in practice and sometimes fails at finding some light contributions. Several rendering methods uses deterministic transformations of paths to improves the local exploration of the path space. Furthermore these techniques can be used to mutualize the exploration between several pixels enabling path reusing and multiple view rendering methods. These transformations and existing applications, in particular path reusing, are presented in chapter 5.

Chapter 3

Unbiased Monte Carlo estimators

The measurement contribution function of a path detailed in the previous chapter is a multidimensional integrand comprised of several variables, each vertex of the path. Integrating in more than 1-dimension is a complex problem, and classical 1-dimensional methods such as piecewise polynomial interpolation or finite elements can hardly be employed due to the complexity of the integrand and a convergence rate that becomes exponentially worse w.r.t to the dimension.

Nevertheless several early *radiosity* techniques [Goral et al. 1984, Cohen and Greenberg 1985, Cohen et al. 1988, 1993, Sillion and Peuch 1994] exploited finite element methods to compute light transport integrals, but were constrained by scene complexity – i.e. limited to a moderate amount of geometry, simple materials and a finite number of bounces – and required long precomputation and large storage. The complexity of these methods depends on the number of space subdivisions or patches and on the number of indirect bounces to integrate – i.e. the dimension of the light paths.

For that reason rendering algorithms moved to use Monte Carlo (MC) methods [Cook 1986, Kajiya 1986] for which the convergence rate does not depend on the random variable dimensionality. The idea behind Monte Carlo is that we only need to evaluate the integrand at some stochastically chosen points in the domain to get an estimate of its integral. Hence we require the ability to sample light paths and to evaluate their contribution. Monte Carlo methods are more generic and can elegantly handle arbitrary light scattering models, at the expense of variance that results in high-frequency noise in the images. Fortunately this noise vanishes as the number of samples averaged per pixel increases.

In this chapter we start by detailing common existing models for evaluating integrals with Monte Carlo and review their main properties and drawbacks.

Monte Carlo methods are well studied from the neutron transport research [Metropolis and Ulam 1949, Metropolis et al. 1987] and several books propose a great introduction on the topic such as *Monte Carlo statistical methods* by Robert and Casella. An rendering oriented introduction on MC can be found in [Pharr et al. 2016, section 13] and further details can be found in Eric Veach thesis [Veach 1998]. For advanced importance sampling methods and richer examples we recommend the book *Monte Carlo theory, methods and examples* by Art B. Owen.

3.1 Prerequisites

Monte Carlo methods are statistical methods based on the probability theory. We recall some basic definitions to better introduce Monte Carlo estimators.

3.1.1 Random variable

A random variable X is a function defined on a probability space which takes values that depend on a stochastic process. Furthermore a function f of a random variable X is a random variable, we write $Y = f(X)$. This is the basic object we manipulate in rendering application: we generate random light paths in a 3D scene. A sampled path X is thus a random variable, and the contribution $f(X)$ of a sampled path is also a random variable.

3.1.2 Distribution and density

A continuous random variable is described by a continuous cumulative distribution function (CDF). The CDF is the probability of a random variable to take a value less or equal than a threshold.

$$\mathrm{cdf}(x) = P(X \leq x) \tag{3.1}$$

The probability density function (PDF) of a random variable is the derivative of the CDF.

$$\mathrm{pdf}(x) = \frac{\mathrm{d}\,\mathrm{cdf}(x)}{\mathrm{d}x} \tag{3.2}$$

In practical applications (sampling), the PDF describes the effective repartition of the probabilities over the domain: regions where the PDF takes high value are more likely to be sampled, on the contrary regions where the PDF is close to zero have fewer samples distributed as illustrated by the histogram in Figure 3.1. For the sake of simplicity we will denote $p(x)$ as a PDF in the rest of the dissertation. Since a CDF is monotonically increasing and its maximal value is 1, the associated PDF is strictly positive and integrate to 1 over the continuous domain of events.

$$\int_{\Omega} p(x)\mathrm{d}x = 1 \tag{3.3}$$

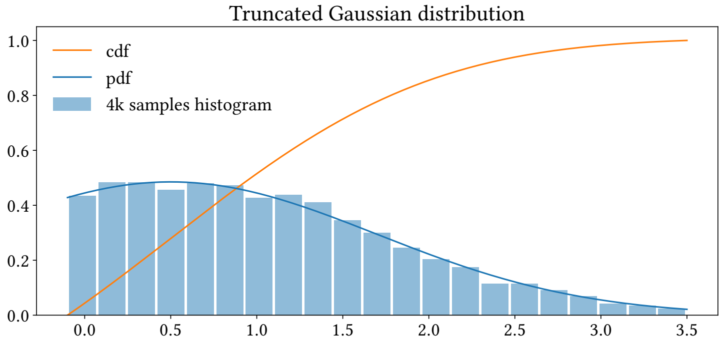


FIGURE 3.1 – Example of Gaussian distribution of mean 0.5 and standard deviation 1.2 truncated to the interval $[-0.5, 2.5]$. The effective number of samples distributed (histogram) is proportional to the value of the PDF (blue curve).

3.1.3 Discrete case

A random variable may also be discrete when the possible set of event is finite countable (e.g. the outcomes of a dice). It is then described by a discrete CDF which is piecewise constant. The equivalent of the PDF in the discrete case is the probability mass function (PMF) (sometimes called discrete PDF) which is normalized over the sum of all events probabilities, $\sum_{\Omega} p(x) = 1$. Discrete random variables are illustrated in Figure 3.2. They are widely used in computer graphics applications for discrete stochastic selection, e.g. light sampling, 2D texture sampling, mipmap hierarchy sampling (trees), etc [Haines and Akenine-Möller 2019, chap. 16][Marrs et al. 2021, chap. 21].

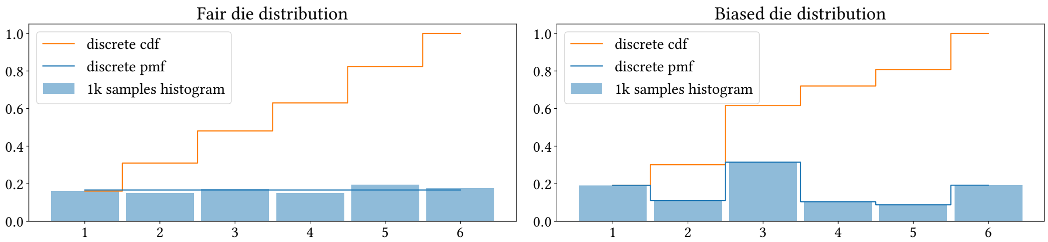


FIGURE 3.2 – Example of 6-sided dice distributions and associated discrete CDF and PMF. The left plot illustrates the fair case where all outcomes have the same probabilities (uniform discrete distribution). On contrary the right plot depicts a rigged die for which the probabilities has been biased. The effective number of samples distributed (histogram) is proportional to the value of the PMF for each possible outcome (blue curve).

3.1.4 Expected value

The *expected value* or *first moment* of a random variable $Y = f(X)$ over a measurable space Ω with associated measure μ is:

$$\mathbb{E}[Y] = \int_{\Omega} f(x)p(x)\mathrm{d}\mu(x) \tag{3.4}$$

The expected value has the following useful properties which serve as a basis for numerous derivations:

$$\forall \alpha \in \mathbb{R}, \quad \mathbb{E}[\alpha X] = \alpha \mathbb{E}[X] \qquad \text{and} \qquad \mathbb{E}\left[\sum_i X_i\right] = \sum_i \mathbb{E}[X_i] \tag{3.5}$$

3.1.5 Variance

The *variance* or *second central moment*, of a random variable Y is defined as:

$$\mathbb{V}[Y] = \mathbb{E}\left[(Y - \mathbb{E}[Y])^2\right] \tag{3.6}$$

and measures the dispersion of the random variable w.r.t its expectation. This is an important measure to determine whether chosen density is a good choice for a given integrand. The base variance formula from Equation 3.6 can be simplified as follows:

$$\begin{aligned}\mathbb{V}[Y] &= \mathbb{E}\left[(Y-\mathbb{E}[Y])^2\right] \\ &= \mathbb{E}\left[Y^2 - 2Y\mathbb{E}[Y] + \mathbb{E}[Y]^2\right] \\ &= \mathbb{E}\left[Y^2\right] - 2\mathbb{E}[Y]\,\mathbb{E}[Y] + \mathbb{E}[Y]^2 \\ &= \mathbb{E}\left[Y^2\right] - \mathbb{E}[Y]^2\end{aligned} \tag{3.7}$$

The variance has the following properties:

$$\forall \alpha \in \mathbb{R}, \quad \mathbb{V}[\alpha X] = \alpha^2 \mathbb{V}[X] \qquad \text{and} \qquad \mathbb{V}\left[\sum_i X_i\right] = \sum_i \mathbb{V}[X_i] \tag{3.8}$$

which are useful to derive the variance of Monte Carlo estimators as we shall see in the next section.

3.1.6 Monte Carlo Estimator

Unlike Equation 3.4, in most applications our goal is to compute the following simpler integral:

$$I = \int_\Omega f(x)\mathrm{d}\mu(x) \tag{3.9}$$

It is interesting that the integral I equals the expected value of the ratio of the integrand and the PDF of a random variable X (which is a random variable):

$$\begin{aligned}\mathbb{E}\left[\frac{f(X)}{p(X)}\right] &= \int_\Omega \frac{f(x)}{p(x)} p(x)\mathrm{d}\mu(x) \\ &= \int_\Omega f(x)\mathrm{d}\mu(x) \\ &= I\end{aligned} \tag{3.10}$$

Hence for a given realization of X with an arbitrary density $p(X)$ over the domain, the ratio $\frac{f(X)}{p(X)}$ is an unbiased estimate of I as long as $p(x) > 0$ whenever $f(x) > 0$. A Monte Carlo estimator with one realization writes:

$$\hat{I}_1 = \frac{f(X)}{p(X)} \tag{3.11}$$

where the hat denotes the estimator and the subscript denotes the number of realizations. Using properties from Equation 3.5, it is thus possible to average over n realizations to get a better estimate:

$$\hat{I}_n = \frac{1}{n}\sum_{i=1}^{n} \frac{f(X_i)}{p(X_i)} \tag{3.12}$$

which is also a Monte Carlo estimator with n realizations. Note that this is an estimator of the true mean hence for different n-samples realizations the estimated value may change but the estimate stabilizes when n becomes very large.

3.1.7 Quality of the estimator

If the distribution of samples p is appropriate, the estimator converges to the expected value. But we know that there is a remaining error for a fixed sample budget n. To quantify this error, we can study the variance, the bias and the error of the estimator and their convergence rates.

Variance

We compute the variance of the n-samples Monte Carlo estimator $\hat{I}_n$ using the properties from Equation 3.8:

$$\begin{aligned}\mathbb{V}\left[\hat{I}_n\right] &= \mathbb{V}\left[\frac{1}{n}\sum_{i=1}^{n}\frac{f(X_i)}{p(X_i)}\right] \\ &= \frac{1}{n^2}\mathbb{V}\left[\sum_{i=1}^{n}\frac{f(X_i)}{p(X_i)}\right] \\ &= \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{V}\left[\frac{f(X_i)}{p(X_i)}\right] \\ &= \frac{1}{n}\mathbb{V}\left[\frac{f(X)}{p(X)}\right] \\ &= \frac{1}{n}\mathbb{V}\left[\hat{I}_1\right]\end{aligned} \tag{3.13}$$

This equation shows that the variance decreases linearly with the number of samples, reaching zero as n tends to infinity:

$$\lim_{n\to\infty}\mathbb{V}\left[\hat{I}_n\right] = 0 \tag{3.14}$$

Bias

The bias of an estimator is defined as the difference between the expected value of an estimator and the true integral value:

$$\mathbb{B}\left(\hat{I}\right) = \mathbb{E}\left[\hat{I}\right] - I \tag{3.15}$$

An estimator is said to be *unbiased* if its expected value equals the true integral value, on the contrary it is said to be *biased* if its expected value do not equal the true integral value, hence:

$$\begin{aligned}\mathbb{E}\left[\hat{I}\right] = I \quad &\Leftrightarrow \quad \hat{I} \text{ is unbiased} \\ \mathbb{E}\left[\hat{I}\right] \neq I \quad &\Leftrightarrow \quad \hat{I} \text{ is biased}\end{aligned} \tag{3.16}$$

The bias can either be constant for any fixed sample budget, $\forall n, \mathbb{B}\left(\hat{I}_n\right) = c$, or vanishing as n increases. In the latter situation, the estimator is said to be *consistent* and the bias converges to zero as n tends to infinity:

$$\forall n, \mathbb{B}\left(\hat{I}_n\right) = h(n) \quad \lim_{n\to\infty} h(n) = 0 \quad \Leftrightarrow \quad \hat{I} \text{ is biased but consistent} \tag{3.17}$$

Unbiased algorithms are preferred when convergence is a strong requirement, but biased alternatives can often improve the error and even performance particularly when it is also consistent. For additional details regarding the bias in Monte Carlo estimators, we refer the reader to [Hachisuka 2013]. Several bounds of convergence of the mean, variance and bias are summarized in Table 3.1.

	Mean	Variance	Bias
Unbiased	$\lim\limits_{n\to\infty} \hat{I}_n = I$	$\lim\limits_{n\to\infty} \mathbb{V}\left[\hat{I}_n\right] = 0$	$\forall n, \mathbb{B}\left(\hat{I}_n\right) = 0$
Biased	$\lim\limits_{n\to\infty} \hat{I}_n = I + c$	$\lim\limits_{n\to\infty} \mathbb{V}\left[\hat{I}_n\right] = 0$	$\forall n, \mathbb{B}\left(\hat{I}_n\right) = c$
Consistent	$\lim\limits_{n\to\infty} \hat{I}_n = I + h(n)$	$\lim\limits_{n\to\infty} \mathbb{V}\left[\hat{I}_n\right] = 0$	$\lim\limits_{n\to\infty} \mathbb{B}\left(\hat{I}_n\right) = 0$

TABLE 3.1 – Table of different limits of estimators properties w.r.t estimator type.

Error

If the expected value I is known or an accurate estimate is available, we can compute the *mean squared error* as the expected value of the squared error of an estimator w.r.t to the integral value:

$$\begin{aligned} \text{MSE}\left(\hat{I}\right) &= \mathbb{E}\left[\left(\hat{I} - I\right)^2\right] \\ &= \mathbb{V}\left[\hat{I}\right] + \mathbb{B}(\hat{I})^2 \end{aligned} \tag{3.18}$$

which quantify how close an estimator is from the expected value. Note that an alternative form of the MSE is the sum of the variance and the squared bias of an estimator. If the estimator is unbiased it is straightforward to show that the MSE equals the variance and converges to zero as well when n grows to infinity. If the estimator is biased, the MSE converges to a constant and never reaches the true integral value. If the estimator is consistent, the MSE tends to zero as a limit when n tends to infinity. The convergence rate of the MSE of an unbiased estimator is then similar to the convergence of the variance which decreases linearly w.r.t n. Hence, the convergence of the root mean squared error, $\text{RMSE} = \sqrt{\text{MSE}}$, is $O\left(n^{-0.5}\right)$.

Convergence

Depending on the integrand complexity a Monte Carlo estimator may require a large amount of samples to get an accurate estimate of the integral I. The measure of the accuracy of the estimator is determined by its convergence rate. It can be shown that the basic Monte Carlo estimator, using uniform random samples over the domain, converges in $O\left(n^{-0.5}\right)$. The latter property means that it requires four times more samples to divide the visible error by two.

The convergence rate of MC estimators can be improved by using *quasi Monte Carlo* methods based on *low discrepancy sequences* (e.g. Halton, Hammersley, Sobol, lattices) or *optimized pointsets* (e.g. Dart Throwing, BNOT, LDBN, SOT), see [Keller et al. 2019] for a recent course on the topic.

For example, the error of an estimator using the Sobol sequence decreases with rate $O(\frac{(\log n)^d}{n})$, which is close to $O(n^{-1})$ when the number of dimensions d is small and the number of sample n is large. Under the latter conditions, this means that with two times more samples the error is divided by a factor two.

3.1.8 Discussion

Even though the convergence rate may not be optimal, it is worth mentioning that the convergence rate of Monte Carlo estimators does not depend on the random variable dimensionality. This property makes these methods the only practical solution for high-dimensional integration problems. To get a better estimate of an integral we have to reduce the error of the estimator, consequently we have to reduce its variance (in the case of an unbiased estimator). The variance is closely related to the choice of the distribution of samples. As we shall see in the next section the best possible unbiased estimator is an estimator for which all realizations are constant.

Unbiased estimators are theoretically *perfect* however they often suffer from variance that is long to disappear, which directly affects the error. Nevertheless, carefully using biased estimators is often acceptable as long as the quality of the results is not degraded – because visible artifacts in images are not tolerable in practice – since they can dramatically reduce the variance or the computational cost of an estimator. Biased method are often used in real time application for performance reasons and as smoothing techniques since the presence of a high error is not desirable and would require extra filtering.

Furthermore biased but consistent estimators can be great compromise, with an infinite sample budget the expected value is reached, and at reasonable sample count the bias may barely be noticeable. Especially in the context of path reusing, where we massively reuse samples at a reduced cost, the increased number of paths further reduces the bias of consistent estimators. In chapter 7, we explore a family of such biased but consistent estimators that are practical in the context of multi view rendering.

3.2 Multidimensional integrals and Monte Carlo

Let us recall that the pixel integral takes the following form:

$$I = \int_{\Omega} f(\bar{x}) \mathrm{d}\mu(\bar{x}) \tag{3.19}$$

and by extension the multidimensional version can be written:

$$I = \int_{\Omega_0} \int_{\Omega_1} \dots \int_{\Omega_k} f(x_0, x_1, \dots, x_k) \prod_{i=0}^{k} \mathrm{d}x_i \tag{3.20}$$

A Monte Carlo estimator approximates the integral by random sampling of the integration domain and taking the average contribution over n sample realizations:

$$I \approx \hat{I}_n = \frac{1}{n} \sum_{i=1}^{n} \frac{f(\bar{X}_i)}{p(\bar{X}_i)} \tag{3.21}$$

Here, each sample $\bar{X}_i$ is a sequence of vertices in path space that has been sampled with PDF $p(\bar{X}_i)$, taking into account the repartition of samples in space. Since our random variable X_i is multidimensional its PDF here is a joint PDF that is the product of all independently sampled events PDF on the path. It can be shown that the expected value of the estimator $\hat{I}_n$ equals I with any correctly defined PDF:

$$\begin{aligned}\mathbb{E}\left[\hat{I}_n\right] &= \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\frac{f(\bar{X}_i)}{p(\bar{X}_i)}\right] \\ &= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\frac{f(\bar{X}_i)}{p(\bar{X}_i)}\right] \\ &= \frac{1}{n}\sum_{i=1}^{n}\int_{\Omega}\frac{f(x)}{p(x)}p(x)\mathrm{d}x \\ &= \frac{1}{n}\sum_{i=1}^{n}\int_{\Omega}f(x)\mathrm{d}x \\ &= \frac{1}{n}\sum_{i=1}^{n}I \\ &= I\end{aligned} \tag{3.22}$$

3.3 Uniform sampling

The simplest way to generate samples over a domain is by uniform sampling. Hence the PDF is constant over the domain Ω and equals the inverse of the domain volume $V_\mu(\Omega)$ w.r.t to the associated measure μ. Using uniform sampling the Monte Carlo estimator reduces to the following:

$$\begin{aligned}\hat{I}_n &= \frac{1}{n}\sum_{i=1}^{n}\frac{f(\bar{X}_i)}{p(\bar{X}_i)} \\ &= \frac{1}{n}\sum_{i=1}^{n}f(\bar{X}_i)V_\mu(\Omega) \quad \text{with} \quad p(\bar{X}_i) = \frac{1}{V_\mu(\Omega)} \\ &= \frac{V_\mu(\Omega)}{n}\sum_{i=1}^{n}f(\bar{X}_i)\end{aligned} \tag{3.23}$$

Uniform sampling can be done by sampling the d-dimensional hypercube and stretching each dimension to match the domain extent. An example integration problem using uniform sampling is presented in Figure 3.3. This Monte Carlo estimator is unbiased and converges to the expected value.

3.4 Importance sampling

When the integrand f is almost zero everywhere except in a small area of the domain, classic uniform sampling may converge slowly. Indeed, uniform sampling distributes an equal amount of samples everywhere in the domain. However, in this case most of them gather zero contribution, thus variance increases drastically. To address this problem *Importance Sampling* (IS) methods have been introduced [Kahn 1950a,b, Kahn and Marshall 1953]. The idea behind IS is to use non uniform distributions to sample more densely the regions of interest – i.e. biasing the samples to cover the regions where the integrand is non zero – and weighting the integrand by the PDF accounting for the modified distribution of samples. The IS estimator is unbiased as long as the modified PDF is positive whenever the integrand is. Although carefully chosen IS strategies can seriously reduce the variance of the estimator, choosing a wrong IS scheme can make the variance worse.

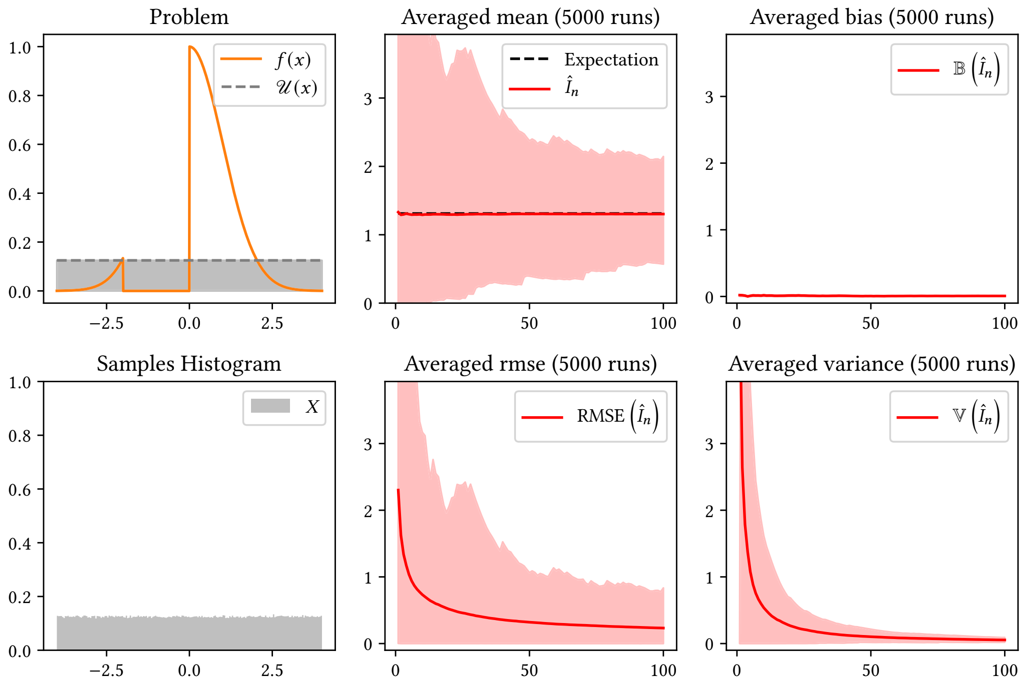


FIGURE 3.3 – Uniform Sampling in 1D. The integrand f in the top left plot (orange) is integrated on the domain $[-4, 4]$ using uniform samples with PDF in dashed gray. We plot the mean, rmse, variance and bias of the uniform Monte Carlo estimator averaged over 1000 realizations of 1 to 100 samples. The bias here is a residual due to numerical errors but is expected to be zero since the estimator is unbiased and the expected value is matched on average. The variance decreases with rate $O\left(n^{-1}\right)$ and the rmse with rate $O\left(n^{-0.5}\right)$.

3.4.1 Zero variance importance sampling

The source of error in Monte Carlo estimators comes from the variance of the samples w.r.t to the integrand. One property of IS is that by distributing samples exactly proportionally to the value of the integrand we can construct a *zero variance estimator*. Indeed, by making $p \propto f$ the ratio $\frac{f}{p}$ becomes constant and the source of variance can be cancelled:

$$\begin{aligned}
\mathbb{V}\left[\frac{f(\bar{X}_i)}{p(\bar{X}_i)}\right] &= \mathbb{E}\left[\left(\frac{f(\bar{X}_i)}{p(\bar{X}_i)}\right)^2\right] - \mathbb{E}\left[\frac{f(\bar{X}_i)}{p(\bar{X}_i)}\right]^2 \\
&= \mathbb{E}\left[c^2\right] - \mathbb{E}\left[c\right]^2 \\
&= c^2 - c^2 \\
&= 0
\end{aligned} \tag{3.24}$$

Each sample has the same contribution and the constant c can be deduced as follows:

$$\mathbb{E}\left[\frac{f(\bar{X}_i)}{p(\bar{X}_i)}\right] = \mathbb{E}\left[c\right] = I \iff c = \int_{\Omega} f(x)\mathrm{d}x \tag{3.25}$$

The constant c acts as the normalization factor of the PDF of zero variance samples, which writes:

$$p(x) = \frac{f(x)}{c} = \frac{f(x)}{\int_{\Omega} f(x)\mathrm{d}x} \tag{3.26}$$

Zero variance estimators thus require already knowing the integral value I, which in most cases is impossible to express in closed-form for complex integrands. We illustrate zero variance importance sampling in Figure 3.4.

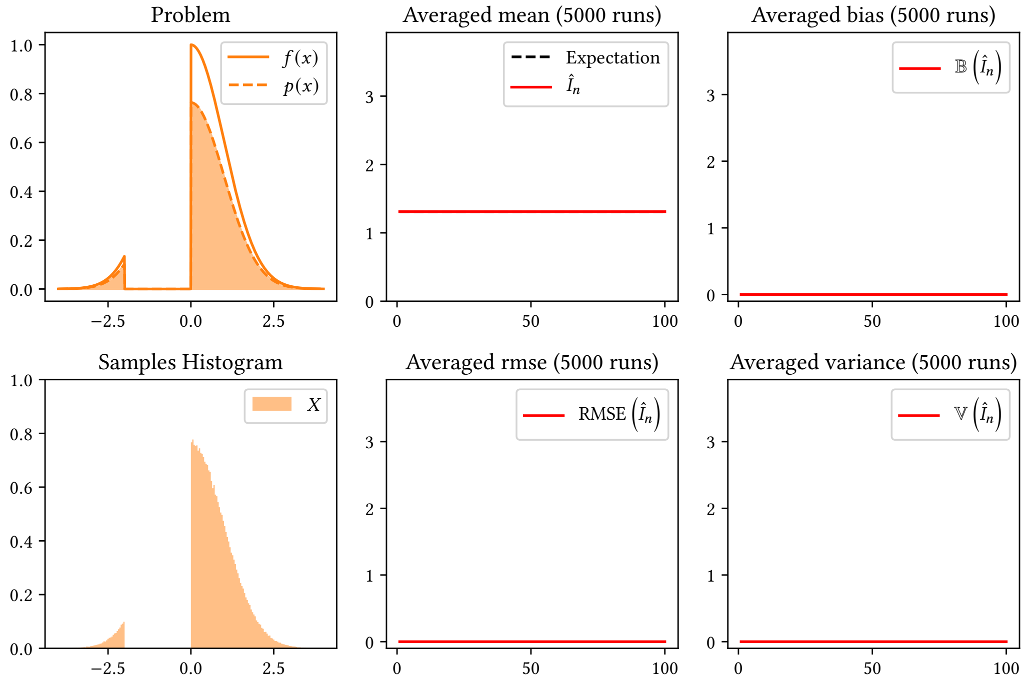


FIGURE 3.4 – Zero variance importance sampling in 1D. The integrand f in the top left plot (orange) is integrated on the domain $[-4, 4]$ using zero variance importance sampling with PDF $p \propto f$ in dashed orange. Due to the PDF construction the mean always equals the expected value, hence the rmse, the variance and the bias of the perfect importance sampling Monte Carlo estimator equal zero.

3.4.2 Design of non uniform distributions

Several methods exist to build non uniform distributions of samples, such as:

- the CDF inversion method which can be analytic [Devroye 1986] or numeric [Cline et al. 2009, Heitz 2020],
- the use of geometric transformations of samples as described in [Hart et al. 2020, sect. 3]: starting from a known distribution of samples and deterministically transforming samples, which can be analytic if the Jacobian of the mapping is known in closed-form or numeric [Hart et al. 2020],
- the rejection sampling method: by sampling a larger domain and rejecting samples that are outside the region of interest, however rejection methods can be very inefficient if the proportion of rejection is greater than the proportion of acceptance,
- generic models and machine learning: starting from a generic model (basis of PDF) we can learn a distribution using adaptive importance sampling [Rubinstein 1997, Lepage 1978, Cornuet et al. 2012], or directly encode the distribution in a latent space using neural importance sampling [Müller et al. 2019].

- stochastic resampling [Talbot et al. 2005, Bitterli et al. 2020]: starting from a set of samples drawn from distribution q, these methods randomly resample within the set of samples to fit a target distribution p, at the expense of additional variance.

3.4.3 Importance sampling of complex integrands

In the case of a complex – multi-term integrands, it is common to perform IS on specific terms of the function for which a zero variance IS scheme is known. This cancels some terms of the function and often leads to a smoother estimate with reduced variance. Another possibility is to find or learn coarser sample distributions we can sample from that resemble the integrand.

Taking back the example integrand from Figure 3.4, the integrand decouple in two terms, a Gaussian g and a discontinuity h, such that:

$$f(x) = g(x) \cdot h(x) \tag{3.27}$$

If the product $g \cdot h$ is complex to sample, a simpler PDF that fits only one of the terms, e.g. a Gaussian distribution proportional to g, can be sampled and still gives a large variance reduction in comparison to uniform sampling as depicted in Figure 3.5.

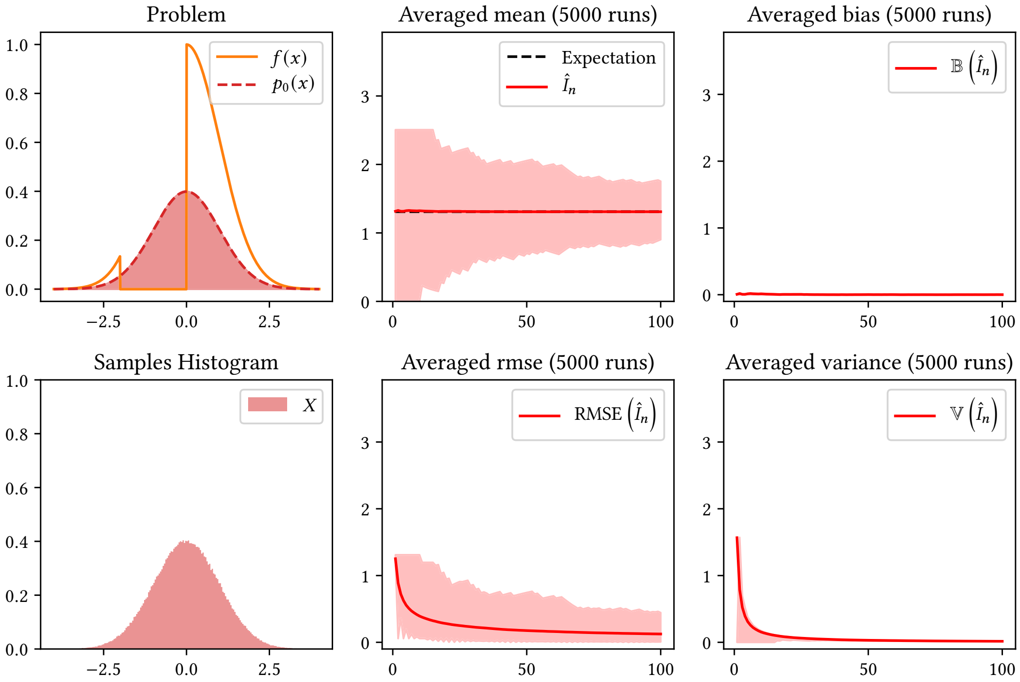


FIGURE 3.5 – Importance sampling in 1D. The integrand f in the top left plot (orange) is integrated on the domain $[-4, 4]$ using importance sampling with a PDF p that resemble f but does not match the discontinuity in dashed red. Since the shape of the PDF is similar to the shape of one term of the integrand the variance of the IS estimator is expectedly reduced.

On the contrary using distributions that poorly fit the integrand shape lead to deficient IS estimators. A deficient IS scheme may have higher variance than uniform sampling or worse, an infinite variance. Such sampling strategies lead to numerically instable estimators and should be avoided to get correct estimates of an integral as presented in Figure 3.6.

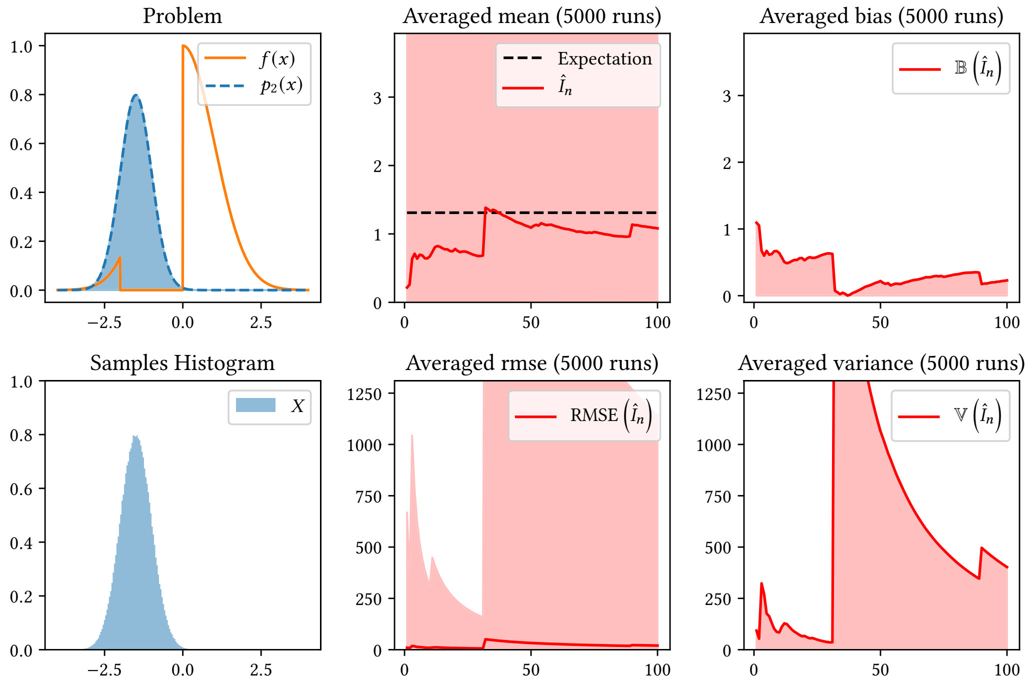


FIGURE 3.6 – Importance sampling in 1D. The integrand f in the top left plot (orange) is integrated on the domain $[-4, 4]$ using importance sampling with a PDF p that does not resemble f in dashed blue. The estimator is numerically unstable since the samples poorly cover the regions where f is positive. Hence the results is numerically biased even after 5000 realizations, and the variance is a thousand times higher than uniform sampling variance. Note that both RMSE and variance plots has been scaled in comparison to previous figures.

3.4.4 Sampling and integration domains

In importance sampling we are only interested in distributing samples where the integrand is positive. Since it can be difficult to achieve, several importance sampling techniques distribute samples in a larger domain than the positive domain of the integrand – for example rejection sampling. Consequently some samples yield zero contributions to the estimator, such that:

$$p(x) > 0 \qquad \text{and} \qquad f(x) = 0 \tag{3.28}$$

but are still taken into account in the sample count n hence in the samples mean. Thus increasing the variance of the estimator. In such situation we would rather count only non zero contributions in the estimator to reduce its variance, replacing n by $n_{\{x|f(x)>0\}}$. But in practice this leads to biased (and not consistent) estimates of the integral, as any sample drawn has to count for the estimator to be unbiased.

3.4.5 Discussion

Importance sampling is one of the best sampling tool in computer graphics application and lots of research has been done to find better IS schemes to sample light transport integrals. In a path tracer, IS is used everywhere to sample camera rays, materials, light sources, shapes, volumes, etc. Finding better IS strategies is still an active area of research.

Sometimes several IS strategies are available to sample the same event but accounting for different terms of the integrand. Different techniques are not guaranteed to yield similar contributions, with some working better than others. Thus, we can aim at selecting or preponderantly weighting the best strategy for each situation using a multi-strategy estimator.

Consequently some estimators has been proposed to optimally combine these strategies in an unbiased way as we shall see in the next section. Note that even though one IS strategy alone leads to a poor estimates of the integral (e.g. high variance or wrong sample distribution), it can still be efficient within a multi-strategy estimator.

3.5 Multiple importance sampling

Given multiple IS strategies, samples can be drawn from each strategy independently and averaged in the associated IS estimator. From these independent estimators, it is then possible to build a new estimator using a linear combination of them:

$$\hat{I}_{\text{naive-mis}} = \sum_{i=1}^{k} \alpha_i \hat{I}_i \qquad \text{with} \qquad \sum_{i=1}^{k} \alpha_i = 1 \tag{3.29}$$

for which we can easily show that it is an unbiased estimator of I:

$$\mathbb{E}\left[\hat{I}_{\text{naive-mis}}\right] = \mathbb{E}\left[\sum_{i=1}^{k} \alpha_i \hat{I}_i\right] = \sum_{i=1}^{k} \mathbb{E}\left[\alpha_i \hat{I}_i\right] = \sum_{i=1}^{k} \alpha_i \mathbb{E}\left[\hat{I}_i\right] = \sum_{i=1}^{k} \alpha_i I = I \tag{3.30}$$

But naively averaging the IS estimators associated to each strategy – or computing any linear combination of them – does not generally reduce its variance. The reason is that each estimator suffers from a different variance and the use of a linear combination simply blends these variance together without reducing the overall variance of the final estimator. The variance of the linear combination writes:

$$\mathbb{V}\left[\hat{I}_{\text{naive-mis}}\right] = \mathbb{V}\left[\sum_{i=1}^{k} \alpha_i \hat{I}_i\right] = \sum_{i=1}^{k} \mathbb{V}\left[\alpha_i \hat{I}_i\right] = \sum_{i=1}^{k} \alpha_i^2 \mathbb{V}\left[\hat{I}_i\right] \tag{3.31}$$

and if one of the strategies has high variance this estimator still assigns a non zero weight to the associated strategy. Hence its variance will be present in the resulting estimator. For that reason, Veach and Guibas [1995] introduced *Multiple Importance Sampling* (MIS) to optimally combine multiple sampling strategies.

3.5.1 Multiple Importance Sampling

Instead of weighting the estimators, MIS weights each sample separately in a combined estimator. It makes the use of a discrete set of weighting functions $w_i(x)$ (one per strategy) that locally compare the PDF assigned by each strategy to a sample. The MIS integral arises by splitting the integral into a sum of integrals, given that the sum of the weighting functions equals one, $\sum_i w_i(x) = 1$:

$$\begin{aligned} I_{\text{mis}} &= \int_{\Omega} f(x)\mathrm{d}x \\ &= \int_{\Omega} \sum_{i=1}^{k} w_i(x) f(x)\mathrm{d}x \\ &= \sum_{i=1}^{k} \int_{\Omega} w_i(x) f(x)\mathrm{d}x \\ &= \sum_{i=1}^{k} I_{i,\text{mis}} \end{aligned} \tag{3.32}$$

From this integral, two Monte Carlo estimators has been proposed:

- The *multi-sample model*, that distributes a fixed number of samples n_i per strategy among n total samples and writes:

$$\hat{I}_{\text{ms-mis}} = \sum_{i=1}^{k} \hat{I}_{i,\text{mis}} = \sum_{i=1}^{k} \frac{1}{n_i} \sum_{j=1}^{n_i} w_i(x_{ij}) \frac{f(x_{ij})}{p_i(x_{ij})} \tag{3.33}$$

- The *one-sample model*, that draws one sample x_i from strategy i, stochastically selected among k with a fixed probability c_i, with $\sum_i c_i = 1$:

$$\hat{I}_{\text{os-mis}} = \frac{w_i(x_i) f(x_i)}{c_i p_i(x_i)} \tag{3.34}$$

These two MIS estimators are unbiased a long as $\sum_i w_i(x) = 1$ whenever $f(x) > 0$ and $w_i(x) = 0$ whenever $p_i(x) = 0$, and are usually averaged over m realizations.

3.5.2 Weighting heuristics

The weighting functions $w_i(x)$ can be chosen arbitrarily while they respect the above conditions. Several weighting heuristics has been proposed, such as the *balance heuristic*:

$$w_i(x) = \frac{c_i p_i(x)}{\sum_{i=1}^{k} c_k p_k(x)} \tag{3.35}$$

, the *power heuristic* that raises the PDFs in Equation 3.35 at a power β, the *maximum heuristic* that only keeps the maximum PDF (it can be seen as the power heuristic with $\beta = \infty$), and the *cutoff heuristic* that only keeps the PDFs above a certain threshold value.

The probability c_i associated to each strategy – the relative number of samples per strategy – is a user defined parameter. In the multi sample model the probability assigned to each strategy is $c_i = \frac{n_i}{n}$, which lead to an alternative form of Equation 3.33:

$$\hat{I}_{\text{ms-mis}} = \frac{1}{n} \sum_{i=1}^{k} \sum_{j=1}^{n_i} w_i(x_{ij}) \frac{f(x_{ij})}{c_i p_i(x_{ij})} \tag{3.36}$$

The one sample model using the balance heuristic weights is similar to *Mixture Importance Sampling*. In fact the final sample distribution is a weighted mixture of strategies with a user defined probability per strategy. To sample from a mixture distribution we first randomly select strategy $p_i \propto c_i$, then we sample $x_i \propto p_i$, the PDF of such samples is then $p(x) = \sum_{i=1}^{k} c_i p_i(x)$ which is exactly what the one-sample model combined with the balance heuristic does (cf. Figure 3.7 and Figure 3.8).

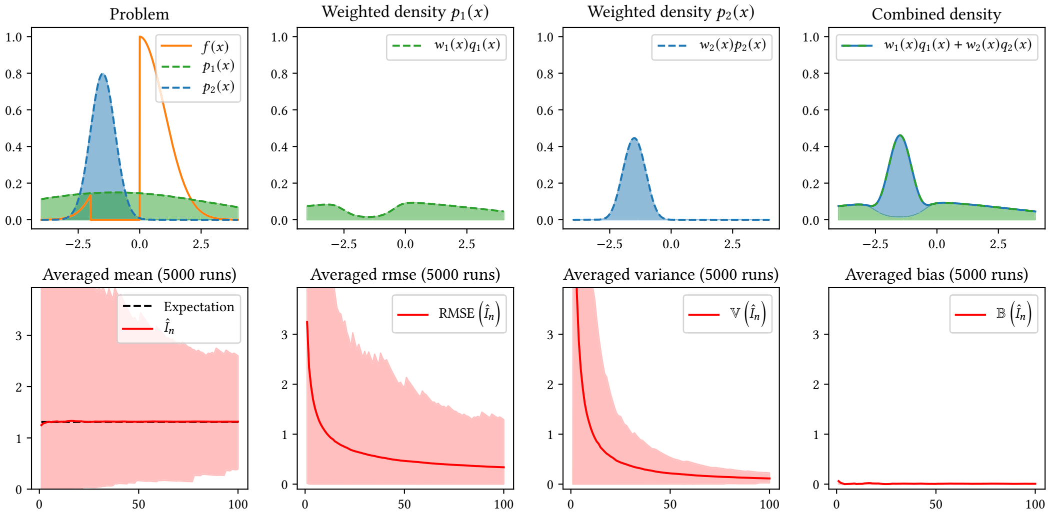


FIGURE 3.7 – Multiple importance sampling in 1D. The integrand f in the top left plot (orange) is integrated on the domain $[-4, 4]$ using two distributions p_1 and p_2. To combine these distribution we rely on the balance heuristic and the one-sample model with the same assigned probability c_i per strategy. Given these strategies the balance heuristic provably minimizes the variance of the combination. The estimator is expectedly unbiased, but still exhibits high variance, since any of the source strategies is a good importance sampler of the integrand.

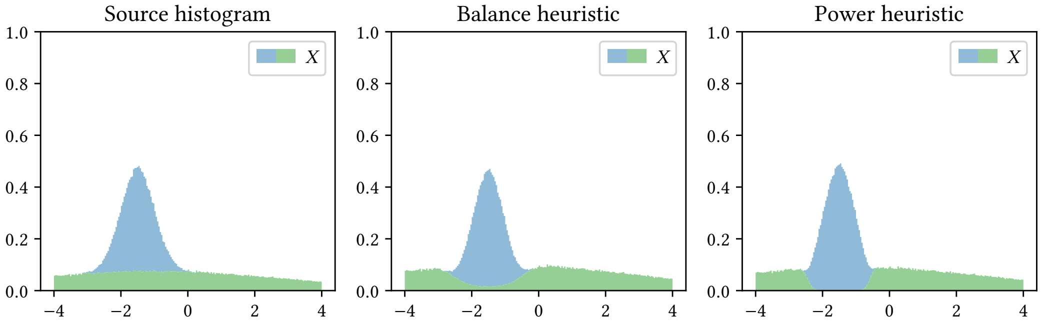


FIGURE 3.8 – The combined density of MIS depends on the weighting heuristic and deforms the source density of samples. On the left we plot the source distribution of samples, in the middle the combined MIS density with the balance heuristic, and on the right the combined MIS density with the power heuristic using $\beta = 4$.

3.5.3 Variance of MIS

The balance heuristic has been proved to minimize the variance of the one-sample estimator Veach and Guibas [1995] and is extensively used in practice. In the case of the multi-sample model with no a priori knowledge regarding the different strategies, the balance heuristic is still the best possible choice since no other weighting heuristic has significantly lower variance. If one of the strategy is known to perform better than any other strategy, using the power heuristic will preponderantly weight the strategy of highest PDF, hence reduce the variance of the overall combination by assigning low weights to other poor strategies.

3.5.4 Continuous Multiple Importance Sampling

Recently, West et al. proposed a continuous version of the MIS framework to handle infinite uncountable sets of techniques. The CMIS integral arises by integrating over a continuous technique space $\mathcal{T}$ instead of summing over a discrete set of techniques like in Equation 3.32, given that the integral of the continuous weighting functions equals one, $\int_{\mathcal{T}} w(t,x)\mathrm{d}t = 1$:

$$\begin{aligned} I_{\text{cmis}} &= \int_{\Omega} f(x)\mathrm{d}x \\ &= \int_{\Omega}\int_{\mathcal{T}} w(t,x)\mathrm{d}t f(x)\mathrm{d}x \\ &= \int_{\mathcal{T}}\int_{\Omega} w(t,x)f(x)\mathrm{d}x\mathrm{d}t \\ &= \int_{\mathcal{T}} I_{t,\text{cmis}}\mathrm{d}t \end{aligned} \tag{3.37}$$

The CMIS integral is not directly practical to evaluate since it would require integrating the whole technique space for each sample – i.e. the CMIS integral over the technique space requires evaluating a marginal. Hence, the authors propose to stochastically sample a set of pairs (technique, sample) to approximate the marginal [West et al. 2020, equation 12], which constitutes the *Stochastic MIS* (SMIS) estimator. This estimator is then a stochastic discretization of the CMIS integral, while the MIS integral is a deterministic discretization of the technique space, and still gives an unbiased estimate of the integral I.

3.5.5 Discussion

"But the robustness comes at the expense of decreased overall efficiency; the MIS combination can be far from optimal and sometimes significantly better results may be achieved by ignoring all samples but the ones taken from the single best technique."

Kondapaneni et al.

It is worth noting that the strategy of highest density is assigned the highest weight in MIS models. The assumption of the weighting heuristics is that any used PDF is somewhat beneficial for at least one term of the integrand. However, if one strategy erroneously distributes a large number of samples where the integrand is close to zero, the variance of the MIS estimator consequently increases. Ideally we should avoid sampling the strategies that have very poor contributions. But in practice it is often difficult to know a priori if a sampling strategy is beneficial or not. The probability – or relative number of samples – is thus blindly assigned to each strategy without prior knowledge w.r.t the integrand.

Variance reduction for MIS

For that reason, lots of research have been done on optimizing the sample allocation or better weighing heuristics to lower the variance of MIS estimators in the past few years [Sbert et al. 2018, Grittmann et al. 2019, Karlík et al. 2019, Kondapaneni et al. 2019]. These works necessitate per strategy estimation of statistical moments, which are complex to evaluate in a rendering pipeline and require per pixel storage of these additional informations. Authors proposed coarser estimates of these informations, to remain tractable in memory but they remain costly. Kondapaneni et al. derives the optimal MIS weights that are not constrained to be positive and that provably minimize the variance of the combination. However, computing these optimal weights requires storing and inverting a matrix whose size depends on the number of strategies, which is impractical when the number of contributing techniques is high or not known a priori.

3.6 Conclusion

We presented a family of unbiased Monte Carlo estimators, that are widely used in the rendering context. Still, the MIS technique that allows combining several sampling strategies is not optimal and may increase the variance when one of the strategies is a poor importance sampler of the integrand, but is assigned a significant weight. In fact, MIS fails at detecting and ignoring strategies that poorly contributes. For that reason blindly sampling a set of strategies, where some of them distribute their samples outside of the positivity domain of the integrand, results in large variance in the MIS estimator that increases with the number of ghost strategies. This is the case in the context of multiple view rendering as we shall see in chapter 6.

Chapter 4

Unidirectional path construction

We now present how to explicitly generate light paths in a scene to evaluate and average their contributions per pixel integral. There exists several way of constructing such paths, but the most common is to use an incremental and iterative process.

We start from the camera and incrementally construct path vertices until it reaches a light source. Note that the inverse process is also useful for bidirectional methods but we will not cover these aspects in this dissertation since we focus on unidirectional methods. After sampling the primary camera ray the process is incremental since we will repeat the same operations after each real interaction of the path, that is:

- sampling a direction ω_i from interaction x_i,
- tracing the ray (x_i, ω_i) through the scene to find the next interaction x_{i+1}.

Repeating these steps until a light is found is the basis of a path tracer. The directions can simply be sampled regarding the material properties at interactions x_i, but the chances of hitting a light source can be small depending on the scene configuration. Therefore using connection techniques such as *next event estimation* [Shirley and Wang 1994] – i.e. importance sampling of the light sources – can improve the efficiency of the method by gathering more light contributions along the path.

Furthermore, in the presence of participating media, the ray traversal has to take into account that medium particles may be encountered. Hence some methods has been developed to sample medium interactions along a ray.

Finally to control the path length during the construction without biasing the path distribution, a mechanism called Russian Roulette is used to randomly stop the path when the carried energy – or the probability to gather energy – becomes small enough.

Some construction details and more advanced techniques are described in the book *Physically based rendering* [Pharr et al. 2016, chap. 14 and chap. 15]. For further details concerning bidirectional methods see [Pharr et al. 2016, chap. 16].

4.1 Camera sampling

After a pixel has been selected (often deterministically), we generate a ray by importance sampling the camera shutter, the exit pupil of the lens and the pixel area. The final process gives us a ray $r(x, \omega, t)$ and its PDF $p(R) = p(x) \cdot p(\omega) \cdot p(t)$. We then trace the ray towards the scene to find the next interaction point (surface or medium). Additional details concerning the camera sampling process are presented in Appendix A.

4.1.1 Exit pupil sampling

The sampled point x on the exit pupil is considered the first real surface point of the path and is often sampled uniformly on a 2D shape. We commonly uniformly sample a disk of radius r [Shirley and Chiu 1997] with PDF $p(x) = \frac{1}{\pi r^2}$, but other shapes or non uniform distribution can also be used for different effects, for example stylized bokeh.

4.1.2 Direction sampling

Even though the exiting ray direction could be sampled uniformly on a hemisphere aligned with the camera forward direction, it is not efficient in practice. Indeed, using uniform direction sampling results in a large number of rays that can not reach the sensor of the camera. A more efficient method starts by sampling the camera sensor (or pixel) and the camera exit pupil.

Then the direction of the ray ω is constrained by the position sampled within the pixel area x_{film}, the exit pupil position x, and the lens system that deviates the ray [Kolb et al. 1995, Hanika and Dachsbacher 2014]. The pixel is often described as a 2D square that is easily sampled uniformly to generate the point x_{film}, but some non-uniform spatial filter can also be sampled. Depending on the lens model the exiting direction can be difficult to construct.

4.1.3 Time sampling

The time t is sampled in the camera shutter interval with PDF $p(t)$. The density is often proportional to the temporal filter which can be non-uniform. Note that some of the camera parameters such as position or orientation can change with time and have to be interpolated for each sampled time.

4.2 Medium sampling

During the ray traversal if a medium boundary is found, we need to sample particles depth to simulate medium interactions. This process is known as *free-flight sampling*. Several methods exist but the most common is *delta tracking* which combines both free-flight sampling and event type sampling (e.g. null, absorption, scattering) [Novák et al. 2018, Miller et al. 2019].

4.2.1 Distance sampling

This method importance samples a homogenized version of the participating medium. The extinction coefficient of the homogenized medium $\bar{\mu}$ is an upper-bound of the real one μ_t. It is expressed as the sum of the real extinction coefficient and a fictitious complementary one $\bar{\mu} = \mu_t + \mu_n$. We hence generate an interaction at distance t

using either unbounded or bounded sampling of the majorant medium section:

Unbounded sampling $t \in [0, \infty[$	Bounded sampling $\hat{t} \in [0, t_{max}[$
$u \propto \mathcal{U}(0,1)$ $t(u) = \frac{-\log(1-u)}{\bar{\mu}}$ $p(t) = \bar{\mu}\bar{T}(t) = \bar{\mu}(1-u)$	$u \propto \mathcal{U}(0,1)$ $\hat{t}(u) = \frac{-\log(1-u(1-\bar{T}(t_{max})))}{\bar{\mu}}$ $p(\hat{t}) = \frac{\bar{\mu}\bar{T}(\hat{t})}{1-\bar{T}(t_{max})} = \bar{\mu} \cdot (\frac{1}{1-\bar{T}(t_{max})} - u)$

(4.1)

where $\bar{T}$ is the combined transmittance. Using unbounded sampling is convenient when we still want to traverse the medium since distance interactions can pass through the whole section. On contrary bounded sampling forces the interaction to happen inside the medium section. Note that the channel or wavelength can be randomly selected before free-flight sampling and combined with MIS to track spectral extinctions coefficients.

Additional free-flight sampling techniques exist such as regular tracking for piecewise constant media which is costly for high resolution grids, or ray marching which uses constant marching steps inside the medium which is biased. For additional details regarding delta tracking and derived methods we invite the reader to refer to the volumetric rendering survey [Novák et al. 2018].

4.2.2 Event sampling

When the distance t has been sampled we can construct the interaction position using:

$$x_i = x_{i-1} + \omega \cdot t \tag{4.2}$$

where x_{i-1} is the preceding interaction and ω is the ray direction. We then gather all possible events probability at the interaction position denoted x:

$$P_n(x) = \frac{\mu_n(x)}{\bar{\mu}} \qquad P_a(x) = \frac{\mu_a(x)}{\bar{\mu}} \qquad P_s(x) = \frac{\mu_s(x)}{\bar{\mu}} \tag{4.3}$$

and randomly sample one of the event with probability $p_e(x)$ with $e \in \{a, n, s\}$.

If an absorption event is sampled, the path will terminate by gathering the medium emissions. If a scattering event is sampled, the delta tracking loop will terminate to sample the phase function and continue the path. If a null interaction is sampled, we continue the delta tracking loop until either we leave the medium or a real interaction is found in the medium or on a surface. Note that in the case of an homogeneous medium the null scattering coefficient μ_n equals zero hence the interaction is necessarily a real interaction. The delta tracking algorithm depicted in Figure 4.1 and is reproduced in Algorithm 1 in its spectral MIS version [Miller et al. 2019].

Additionally it is convenient to modify the probabilities assigned to each event when the volume is not emissive, since sampling absorption interactions will gather zero contributions and increase the variance. The modified probabilities writes:

$$P_n(x) = \frac{\mu_n(x)}{\bar{\mu}} \qquad P_a(x) = 0 \qquad P_s(x) = \frac{\mu_t(x)}{\bar{\mu}} \tag{4.4}$$

The chain of null interactions sampled between two real interactions – i.e. the real endpoints of the segment (medium or surface) – is refered as a *null scattering chain*.

Algorithm 1 Medium sampling routine inside a medium section.

```
function DeltaTracking($x, \omega$)
    $i \leftarrow$ RandomChannel$(c_1, \dots, c_n)$    ▷ select channel
    $p \leftarrow [1, \dots, 1]_n$    ▷ init PDF
    while True do
        $u \leftarrow \mathcal{U}(0,1)$    ▷ uniform sample
        $t \leftarrow \frac{-\log(1-u)}{\bar{\mu}^i}$    ▷ distance sample
        $x \leftarrow x + t\omega$    ▷ interaction position
        $e \leftarrow$ RandomEvent$\left(P_n^i(x), P_a^i(x), P_s^i(x)\right)$    ▷ event sample
        $p \leftarrow p \circ \left[\bar{T}^{c_1}(t)\mu_e^{c_1}, \dots, \bar{T}^{c_n}(t)\mu_e^{c_n}\right]$    ▷ update PDF
        if $e \in \{a, s\}$ then
            return $\{e, x, \frac{1}{n}\sum_j p_j\}$    ▷ return event, sample and MIS PDF
```

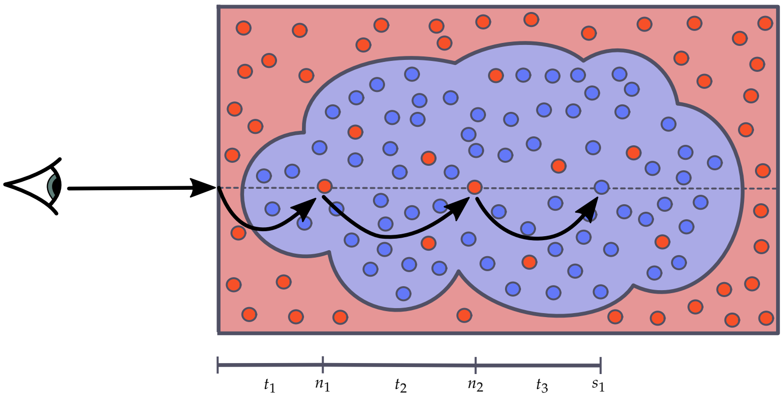


FIGURE 4.1 – Illustration of Delta Tracking in heterogeneous medium. The real medium parameterized by coefficients $\{\mu_a(x), \mu_s(x)\}$ in blue is bounded by the combined homogenized medium with majorant extinction $\bar{\mu}$ (red+blue). Given a ray direction that traverses the medium, free flight distances are sampled in the combined medium until a real medium interaction (blue particle) is found. After each step, the type of medium interaction (null, absorption, scattering) is randomly chosen hence several null interactions (red particles) may be sampled before a real one happens.

4.2.3 Discussion

Delta tracking is a simple method for unbiased sampling of arbitrary participating media. However when the majorant extinction $\bar{\mu}$ becomes large, the free flight sampling steps becomes very small hence the number of DT trials increases. This makes the method cost unbounded, since the null scattering chain length is, but is required to integrate finer details in heterogeneous media.

4.3 Surface and volume intersections

During the ray traversal of the scene, if a surface is reached we found the next real interaction of the path. To test the ray versus scene intersections, it is common to use efficient *bounding volume hierarchies* or other space partitioning structures [Parker et al. 2010, Wald et al. 2014]. In presence of motion blurred animation we use spatio-temporal partitioning structures instead [Woop et al. 2017]. In presence of participating media, the volume boundaries (interfaces) are also stored in a similar acceleration structure that can be intersected to know when a rays enters or exits a medium section. Furthermore, if the volumes are discretized in 3D, we often use *sparse voxel octree* structures [Museth et al. 2019] that allow efficient storage and traversal of discrete voxel sets.

If the surface is emissive – i.e. is a light – the path construction process ends and the contribution of the path can be evaluated. If the surface is not emissive we have to sample a direction to continue the path w.r.t to the surface material.

4.4 BSDF sampling

At every surface interaction (resp. scattering medium interaction) of the path the BSDF (resp. phase function) is sampled to find a direction to continue the path as illustrated in Figure 4.2. Several analytic models allow importance sampling of the shape of the scattering distribution, but numerical approximations or measured data can be used when no closed form is available.

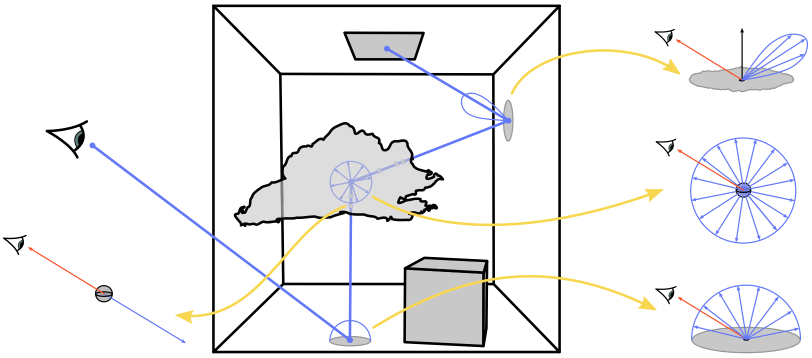


FIGURE 4.2 – An example path of real length $k = 4$ that traverses a scene. The path is built incrementally by sampling a diffuse BSDF at the first bounce, an isotropic phase function at the second bounce in the medium, and a rough BSDF at the third interaction to finally reach a light source. Note that the phase function is sampled only at real scattering interactions, and the ray continues in the forward direction at every null interaction.

Given an outgoing direction ω_o (towards the camera), the incident direction ω_i (towards the light) is sampled with PDF $p(\omega_i, \omega_o)$. We refer the reader to the literature cited in subsection 2.1.5 for further details regarding importance sampling of common BSDF, since most of the models are derived with their importance sampling routine.

4.5 Light sampling

As suggested before, the iterative path construction can be inefficient at finding some light contribution – i.e. reaching a light source. For that reason several methods has been proposed to sample and connect path vertices to light sources directly, which we refer to as *next event estimation* illustrated in Figure 4.3 (right).

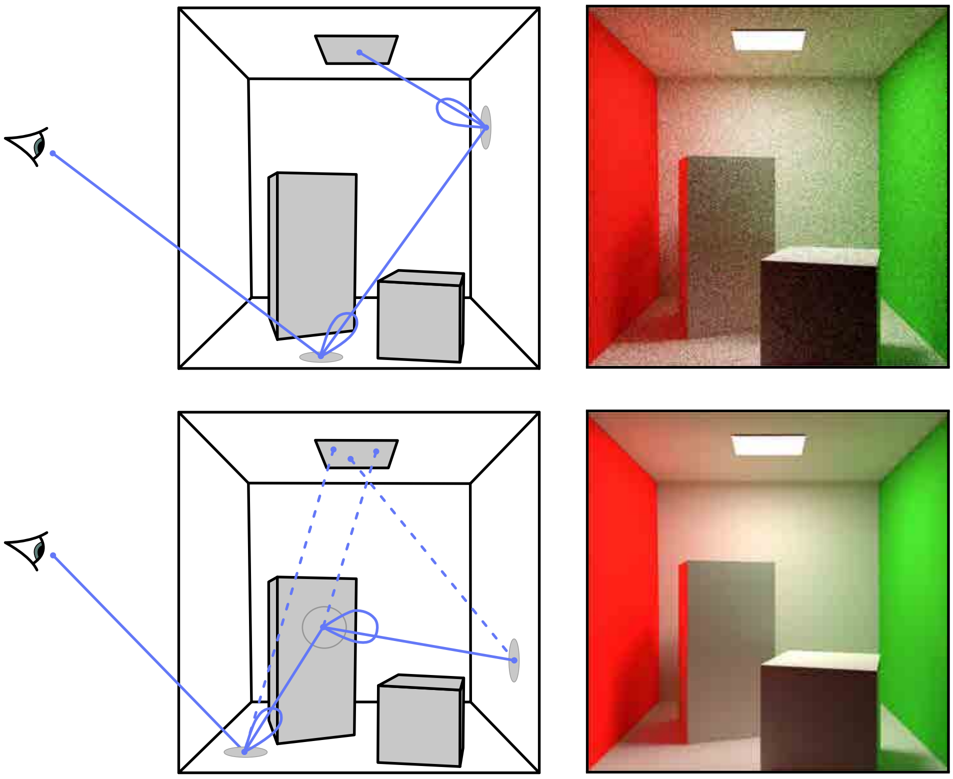


FIGURE 4.3 – Simple BSDF sampling (top row) takes several bounces on the scene to get a light contribution. That is when the sampled direction points by chance towards a light and the path has not been occluded before reaching the light source. However when the light sources are small this technique becomes very inefficient as lights are never found, resulting in higher variance. Next event estimation (bottom row) on contrary performs at every interaction a connexion to a light source position that has been sampled. Which in most situations lead to a dramatic variance reduction. The bottom row render, using light sampling, exhibits almost zero noise at equal rendering time on comparison with BSDF sampling only.

The common approach is to build a discrete CDF w.r.t total area or total power over all available light sources, sample the discrete set of sources and sample a point on the selected source at each path interaction. A connection ray or shadow ray is then traced between the interaction position and the light position to ensure the mutual visibility and to evaluate the transmittance between both points. With this technique, each path interaction has a chance to gather a direct contribution which allow for faster convergence. Note that if the current BSDF is a Dirac delta function (e.g. perfect mirror or glass), NEE has no chance to gather any contribution, since the unique contributing direction can not be found by sampling sources. Furthermore, since a light can be found with either BSDF sampling or light sampling, their respective contributions are generally combined with MIS [Veach and Guibas 1995].

Several improvements exist such as resampling techniques which find sources that better contribute locally [Talbot et al. 2005, Bitterli et al. 2020], the use of light hierarchies to avoid sampling sources that are far from the interaction [Conty Estevez and Kulla 2018, Moreau and Clarberg 2019, Yuksel 2020, Tokuyoshi and Harada 2016], and manifold versions of NEE that better find light sources in presence of discontinuities, e.g. through dielectric interfaces [Hanika et al. 2015] or through highly forward scattering media [Weber et al. 2017].

4.6 Path guiding

Since BSDF sampling or light sampling produces variance in some situations, it could be more efficient to directly sample the product of BSDF and light contributions. Path guiding consists in learning the spatial-directional distribution of light which can be stored in screen space or in scene space, see [Vorba et al. 2019] for a survey. Combining classical BSDF sampling and guiding structure sampling results in a substantial variance reduction. The structure is often learned online, by updating and refining its distribution with new random samples, and simultaneously sampled to generate samples with lower variance.

4.7 Russian Roulette

To increase the efficiency of the estimator we should focus on the sampling of paths that best contributes to the pixel. In fact, paths that bounce around the scene hundreds of times with low contributions increases the variance and are costly to sample. We hence use a Russian Roulette (RR) that stochastically terminate paths and still maintain the pixel estimator unbiased [Pharr et al. 2016, sect. 13.7].

We define a probability $P_i \in [0,1]$ at each interaction x_i, that represents the probability of the path to continue. Conversely we can say that the path is rejected with probability $1 - P_i$. The RR trial is achieved by drawing a uniform random number $u = \mathcal{U}(0,1)$ and comparing it to P_i:

$$\begin{array}{ll} \text{if } u < P_i & \text{the path continues and the estimator is divided by } P_i \\ \text{if } u >= P_i & \text{the path terminates} \end{array} \tag{4.5}$$

Dividing by the continuation probability accounts for the energy lost by terminated paths.

The choice of probability P_i can be arbitrary, for example using a binary coin with probability $P_i = 0.5$ works, but increases the variance when the light contribution of the path is high. In fact, this probability should be proportional to the light contribution of the path to the pixel. It is common instead to use local materials properties (BSDF-based RR) or the current importance (throughput-based RR) carried by the path to get a coarse estimate of what its contribution will be. However these local strategies sometimes fail in scenes dominated by penumbrae and non uniform light distributions. An optimal choice for P_i would require to know the expected contribution of the current path to the pixel which is often not available – but an estimate of its contribution can be used instead [Vorba and Křivánek 2016] using adjoint (i.e. bidirectional) methods.

4.8 Discussion

We have presented the classical building blocks of a state of the art volumetric path tracer which are resumed in Algorithm 2. Note this is a very simplified version to showcase the simplicity of the main unidirectional path tracing loop. Several blocks were omitted such as spectral wavelength discretization [Wilkie et al. 2014, Evans and McCool 1999] or subsurface scattering for skin and translucent materials [Jensen et al. 2001, Christensen 2015] – which is a simplified version of volumetric light transport.

In the next chapter we present path space transformations that build upon path construction methods and allow path reusing in several existing rendering algorithms.

Algorithm 2 Path tracer pseudocode. The `PathTracing` function incrementally construct a path and gather its contribution. In the `Render` function, we sample camera primary rays and average *n* contributions for each pixel of the film. The BSDF sample step can be mixed with a guiding estimate to better find light contributions.

```
function PathTracing(Ray r, Scene s)
    Color c = {0,0,0}
    Interaction x = DefaultInteraction()
    while True do
                                                    ▷ find real interaction
        x = Intersect(r, s)
        if x.medium then
            x = DeltaTracking(x, r, s)
        if not x then return c
                                                    ▷ hit a light
        if x.emission > 0 then
            return c+x.weight*x.emission
                                                    ▷ sample light sources if possible
        if x.bsdf.type $\neq$ BSDF_DIRAC then
            c = c + NextEventEstimation(x, r, s)
                                                    ▷ sample bsdf
        BsdfSample bs = x.bsdf.Sample(x, r)
        x.weight = x.weight*bs.weight
                                                    ▷ russian roulette
        p = x.ContinuationProbability()
        if p $\geq$ Uniform(0,1) then return c
        x.weight = x.weight / p
                                                    ▷ update ray data
        r = Ray(x.position,bs.direction,r.time)

function Render(Camera c, Scene s, int n)
    for Pixel i in c.film do
        for int j in range(0,n) do
            Ray r = c.Sample(i)
            c.film[i] = c.film[i] + PathTracing(r, s) / n
```

Chapter 5

Shift mappings

In this chapter we introduce sample transformations, often referred to as *shift mappings* or *warps*, which denotes the deterministic modification of a sample (uniform or not) that results in a change of its density. Path transformations are widely used to generate non uniform distributions – i.e. importance sampling – or to design more efficient path space exploration. In fact slightly modifying a base sample in a controlled way to better match the integrand distribution results in a reduction of the variance.

We first review the principle and required properties of transformations. We then review several techniques that have been used in rendering applications to modify samples for path reusing purposes.

5.1 Sample transformations

Given a base n-dimensional sample $x = \{x_1, \dots, x_n\}$ with PDF $p(x)$ and a bijective transformation T that deterministically transform the sample such that $x' = T(x)$. The PDF of such transformed sample hence writes:

$$p(x') = p(x) \left|T'\right|^{-1} \tag{5.1}$$

where $|T'|$ is the Jacobian determinant of the transformation. The base PDF $p(x)$ may be a joint PDF depending on how the components have been sampled. Note that several transformations can be applied separately on each component of the n-dimensional sample (e.g. $T_1, \dots, T_n$), or combined on several components at once (e.g. $T_{1,2,3}, \dots, T_{n-1,n}$). We denote these transformations $\{T_i | i \in 1, \dots, m\}$ and compute the PDF of a transformed sample using the product of their inverse Jacobian thanks to the chain rule:

$$p(x') = p(x) \prod_{i=1}^{m} \left|T_i'\right|^{-1} \tag{5.2}$$

The transformed sample x' is constructed by successfully applying each transformation on its respective components. Formally we could define each transformation T_i as an identity for every unchanged dimensions and the respective shifts for each transformed dimension – i.e. those that are modified by the shift. Analogously this

Jacobian naturally appears in a change of variable of integration:

$$\begin{aligned} I &= \int_{\Omega} f(x)\mathrm{d}x \\ &= \int_{T^{-1}(\Omega)} f(T(x)) \left| \frac{\mathrm{d}T(x)}{\mathrm{d}x} \right| \mathrm{d}x \\ &= \int_{T^{-1}(\Omega)} f(T(x)) \left| T' \right| \mathrm{d}x \end{aligned} \tag{5.3}$$

and takes into account the change of density. Furthermore, combining samples from base technique with PDF $p(x)$ and transformed samples with PDF $p(x')$ is possible through MIS. A transformation operation can be applied in any measurable space hence shift mappings has been proposed both in *primary sample space* – i.e. in the unit n-dimensional hypercube – and in *path space*.

Relationship to Metropolis mutations strategies. These transformations can be seen as deterministic versions of *stochastic mutations* from Metropolis rendering methods (MLT), see [Šik and Křivánek 2018] for a survey. These mutations have been introduced to perform local and global exploration of the path space [Veach and Guibas 1997, sect. 5.3][Cline and Egbert 2005, Cline et al. 2005, Kelemen et al. 2002, Jakob and Marschner 2012]. Some of them have been later extended as deterministic transformations, for example:

- *pixel mutations* that correspond to *pixel shifts* in gradient-domain rendering methods,
- *lens mutations* that correspond to *lens shifts* in multiple view rendering methods.

A path mutation is accepted or rejected using a stochastic process where the mutation probability accounts for the change in density. Analogously the PDF of a transformed sample takes into account the change in density through the inverse Jacobian determinant.

5.2 Applications to path reusing

Path reusing consists in finding some point within the scene visible from one pixel, which we refer to as *pivot* point, and reconnecting it towards other pixels for which the pivot is also visible. The chain of events connecting the pixel point on the film to the pivot point in the scene forms the path *prefix*. The prefix can be comprised of several interactions on surfaces areas and inside volumes. The key idea of path reuse is to mutualise the construction and the evaluation of a subpath following the pivot, which is referred to as the path *suffix*. As the number of pixels that are connectible to the same pivot point increases, the construction cost of the path suffix is amortized as its contribution is massively reused among pixels.

These methods has been explored to reuse path contributions after the second interaction in *discrete path reusing* and *path space filtering*, and after the first interaction for *multiple view rendering*. Additionally, *gradient domain rendering* exploits shift mappings to sample correlated gradient paths through neighboring pixels in a similar way as path reusing.

5.2.1 Discrete path reusing

This first class of methods aims at reusing indirect illumination for several pixels by transforming paths between the first and the second bounce [Bekaert et al. 2002, Xu and Sbert 2007, Bauszat et al. 2017, Bitterli et al. 2020]. The original method proposed by Bekaert et al. starts by sampling a path from each pixel or a subset of pixels of an image tile. Then, for each base path a shift mapping is applied consisting of a *path reconnection* from its first interaction to another second interaction on a different pixel path. The latter pixel path suffix is copied to complete the transformed path. This process is repeated for each different pixel that traced a path. Hence at the cost of a single visibility test, several paths suffix can be reused among different pixels as illustrated in Figure 5.1.

Here the pivot points are the second interactions of each traced paths which are connected to all visible first interactions. Each initial path can be seen as an importance sampling technique of the scene area, hence MIS is used to weight contributions of base and transformed paths within a pixel. Note that several improving techniques allows arbitrary repartition of pixels that first generates path. Additionally, connections are not limited to the second interaction, and may happen at any interaction of another base path if correctly weighted.

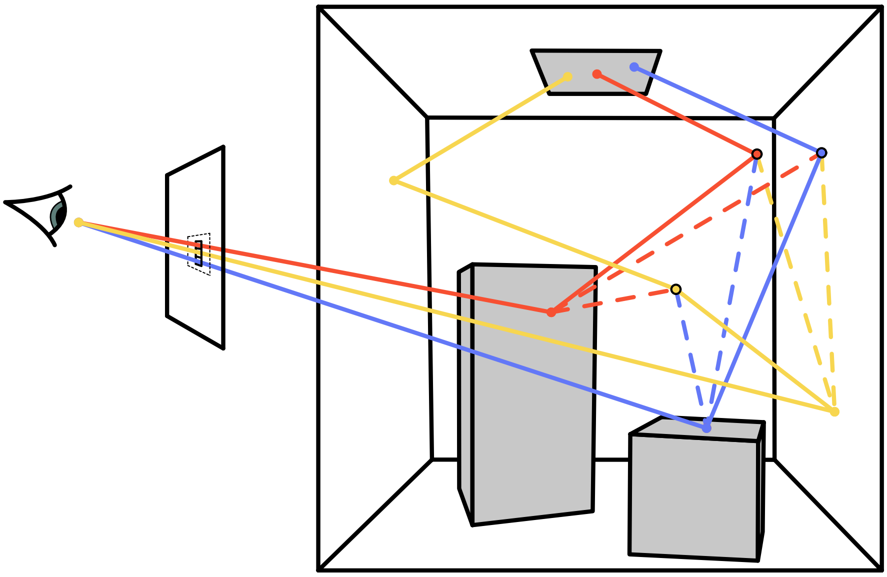


FIGURE 5.1 – Discrete path reusing as proposed in [Bekaert et al. 2002]. Each pixel of a tile in screen space (dashed) traces a complete path that gathers a light contribution. In a second step, the first interaction of a pixel path is connected to any visible pivot point (circled interactions). Finally, several new paths consisting of the initial prefix, a connection segment, and another suffix can be constructed at reduced cost (a visibility test).

5.2.2 Path space filtering

Similarly to discrete path reusing, *path space filtering* methods [Keller et al. 2014, Binder et al. 2019, West et al. 2020] use some path connection shift in path space to benefit from suffix path reuse. However, contributions are averaged for other paths in the vicinity of the prefix endpoint instead of performing connection to every other available strategies as depicted in Figure 5.2. The original biased weighting heuristics has been replaced by unbiased SMIS estimations in [West et al. 2020].

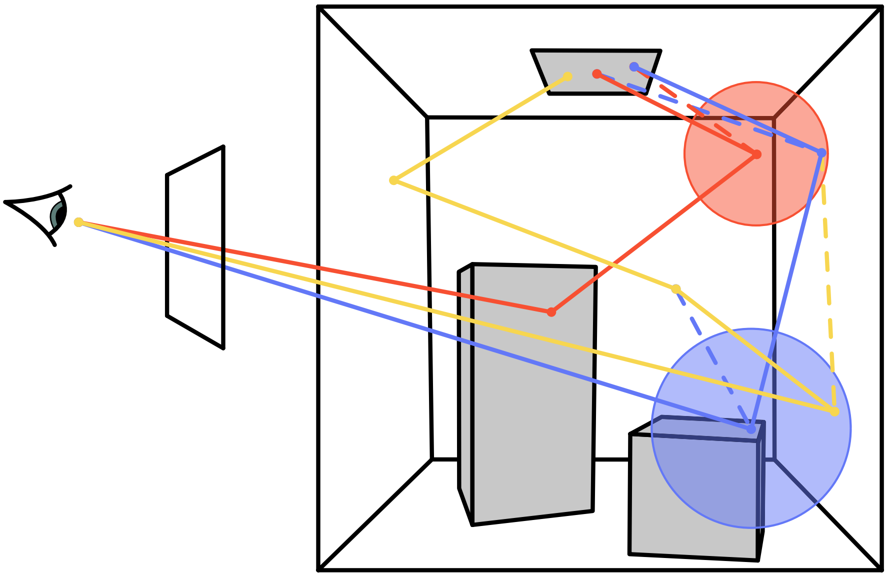


FIGURE 5.2 – Path space filtering proposed in [Keller et al. 2014]. Several pixels trace a complete path that gathers a light contribution. In a spatial grid the neighbors in the vicinity (blue and red circled areas) of interactions several contributions are gathered and averaged with some weighting heuristics.

5.2.3 Gradient domain rendering

The idea behind gradient domain rendering approaches is to sample and estimate gradients as well as an estimate of the image. A final reconstruction step is used to combine gradients and pixel estimates. This process has been successfully extended for different rendering algorithms [Gruson et al. 2018, Bauszat et al. 2017, Manzi et al. 2016, 2015, Kettunen et al. 2015, Lehtinen et al. 2013], see [Hua et al. 2019] for a survey.

To construct gradient images, the use of uncorrelated paths through neighboring pixels and finite difference of their contributions is very ineffective due to the variance of Monte Carlo methods. Instead, a *base* path is first sampled through a pixel i using standard path construction, and then slightly transformed such that it traverses a desired neighboring pixel. The resulting transformed path is referred to as the *offset path*. The more the offset path is close to the base path, the better are the gradients estimates – i.e. the Jacobian of the transformation is closer to one. The conservation of common subpaths in gradients path construction is a sort of path reusing as illustrated in Figure 5.3.

For that reason several shift mappings has been designed to construct offset paths that resemble base paths. A common strategy is to reuse the same uniform random numbers that generated the base interactions, known as *primary sample space copy* [Kelemen et al. 2002, Bitterli et al. 2017], primarily used for mutating paths in primary sample space Metropolis methods. This method assumes that samples

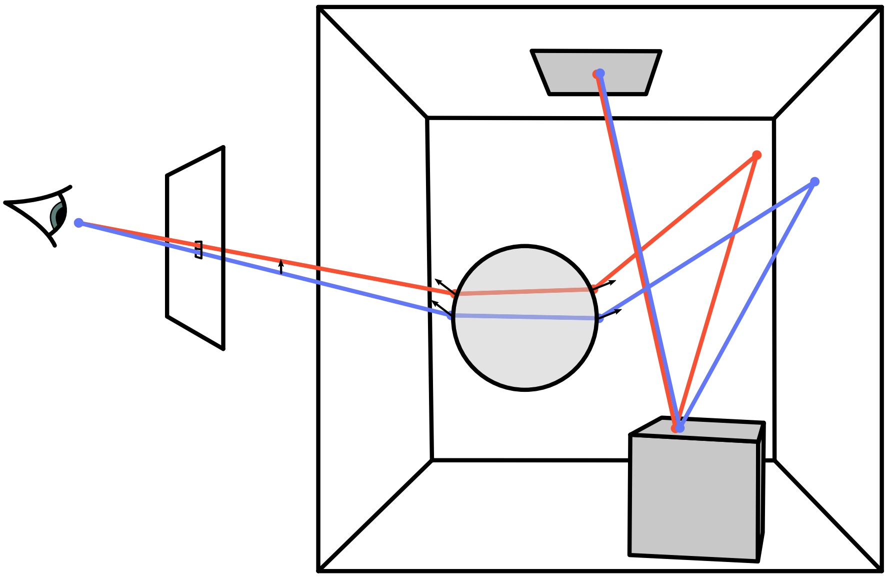


FIGURE 5.3 – Gradient domain path tracing as proposed in [Kettunen et al. 2015]. A base path traversing pixel i is constructed (in blue). Then an offset path (in red) is constructed by applying several transformations to the base path: the initial pixel is shifted to a neighboring pixel, resulting in a different intersection on the scene, the half-vector sampled to traverse the transparent sphere are reused and finally the offset path is reconnected to the base path at the second consecutive diffuse interaction – i.e. the suffix remains unchanged.

generated with the same random numbers are likely to be similar in a close neighborhood. However reusing random numbers sometimes fails to generate paths that are correlated, for example when a segment of the offset path gets occluded. Other strategies have been developed to better constrain the offset path to resemble the base path:

- *Pixel shift mapping* is used to correlate the base film position and the offset film position [Lehtinen et al. 2013, fig. 5],
- *Time shift mapping* is used to correlate base and offset paths in the temporal domain [Manzi et al. 2016, fig. 3],
- *Half vector copy* is used to correlate glossy BSDF samples by reusing the sampled BSDF microfacet normal h [Kettunen et al. 2015, fig. 10],
- *Path reconnection* is used at the first two consecutive diffuse bounces to keep a common path suffix between the base and the offset path, increasing the correlation and reducing the cost of the offset path construction (firstly proposed in the context of path reusing [Bekaert et al. 2002, fig. 1]),
- *Manifold exploration* is used to correlate a chain of specular events connecting two diffuse (i.e. connectible) interactions [Jakob and Marschner 2012, fig. 2 and 6],
- *Photon shift* and *mixed shift* are used to correlate the position of base and offset photons for gradient domain photon density estimation [Hua et al. 2017, fig. 3][Gruson et al. 2018, fig. 4],

- *Depth copy* is used to correlate an homogeneous bounded transmittance sample (i.e. the gather point) from the base path to the offset path for gradient domain volumetric photon mapping [Gruson et al. 2018, fig. 3]. Note that both linear scaling and primary sample copy has also been proposed to shift these homogeneous transmittance samples.
- several other shift mappings are used to correlate higher-order primitives for volume photon density estimation [Gruson et al. 2018, fig. 6].

Combining some of these shift mappings techniques allows for bidirectional, photon density and unified methods to construct offset paths that are reversible and closely resemble base paths. When a shift trial fails and results in a paths with a null PDF – e.g. when a shifted segment gets occluded – the algorithm falls back to finite differences since two base paths are generated pairwise for pixel i and j.

Even though shift mappings in bounded homogeneous media have been proposed, no method has been proposed to handle heterogeneous participating media – i.e. shift mapping of null scattering chains. We develop new volumetric shift mappings as will be discussed in chapter 8.

5.2.4 Multiple view rendering

Multi-view rendering consists in path reusing for disjoint pixel integrals – i.e. with different positions and orientations on separate sensor planes. In fact, path connections can be done at the first real interaction found in the scene, for example in areas or volumes where the visibility is shared between several pixels, as illustrated in Figure 5.4. Again, a path prefix can be reused among several pixels amortizing its construction cost. Multi-view rendering methods can be used to render animated sequences of a static scene or the rendering of several points of view during a single simulation. Even though both classes of methods are similar in nature as they perform path space shifts, a scene with different camera locations causes large transformations of path prefixes, and hence large variations of the Jacobian.

Early approaches of multiple view rendering were designed for surfaces and simple materials only (e.g. diffuse and rough materials) [Henrich et al. 2011, Méndez Feliu et al. 2006, Havran et al. 2003, Adelson and Hodges 1995, 1993]. They first sample a base path from one pixel and then transform the path lens interaction to the lens of another camera using a *lens shift*. The connection between the first real interaction and this new lens vertex constrain the new pixel position on the film plane. Finally the suffix of the transformed path remain unchanged after the pivot point. Note that in the case of a glossy or specular material at the connection pivot, a transformed path may gather a null contribution for its respective pixel integral, hence increasing its variance. Detecting these poor paths is a complex task, and has not been explored. Furthermore, these technique has not been extended for heterogeneous medium since no shift mapping of null scattering chain is available, though some work has been done for shift in homogeneous media [Tessari et al. 2017, Gruson et al. 2018].

For camera and scene animation Jorge Schwarzhaupt reuses paths along the time dimension. He reevaluates occlusions due to object or camera motion, but not the shading to remain tractable. This approximation results in a minor bias and requires storing entire base paths to reevaluate transformed positions after the time shift.

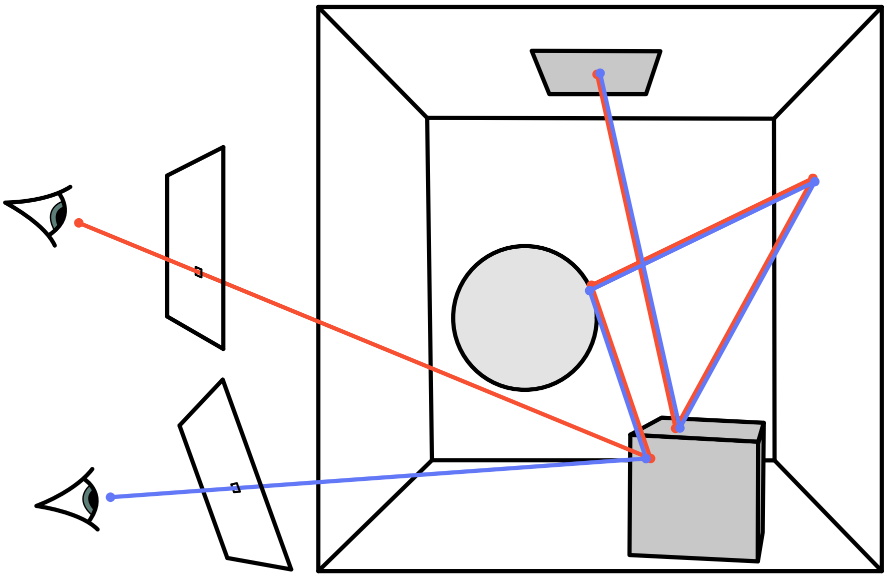


FIGURE 5.4 – Multiple view rendering as proposed in [Havran et al. 2003]. A base path traversing pixel i is constructed (in blue). Then an offset path (in red) is constructed by applying several transformations to the base path: the initial lens interaction is shifted to another camera lens, the pivot intersection on the scene remains unchanged, the pixel position j is then constrained by the lens and scene points. The offset path is then connected to the base path at the pivot interaction and the suffix is unchanged.

Other multi-view approaches reuse at fixed time only, for animations of static scenes, stereo rendering or lightfield rendering.

Additionally, base and transformed paths can be combined using MIS [Méndez Feliu et al. 2006, Henrich et al. 2011, Schwarzhaupt 2019]. However, the authors do not deeply discuss or explore the fact that MIS may severely increase the variance in multiple view rendering.

Finally, another approach of multiple view rendering consists in reusing path using lens shifts for integrated depth of field supersampling [Schwarzhaupt 2019], which consists in reusing a sample that is out of focus several times through different locations on the lens to importance sample the circle of confusion.

5.3 Discussion

We have presented path transformations and their applications in context of path reusing, in particular in multi view rendering methods. As we shall see in chapter 6 and chapter 8, we propose to build new shift mappings of null scattering chains to allow path reusing in presence of heterogeneous mediums. Our new mappings can further be combined with existing shifts to allow for richer path transformations, and thus better path space exploration for path reuse purposes. We demonstrate our new methods for volumetric multiple view rendering in chapter 9 and chapter 10.

Chapter 6

The multi-view rendering problem

In this chapter we briefly formalize the multi view rendering problems arising in the state of the art methods that are solved in the next chapters.

6.1 The multi-view pixel integral

To formalize the multi view problem, we consider every pixel that is present on a scene as a source sampling strategy that can generate prefixes in the scene. The idea behind multiple view rendering is to shift a base prefix, towards other pixels to reuse a common suffix path. However, we do not know a priori if a prefix sampled from pixel i could be transformed to reach another pixel j. That is when pixels i and j share a common subdomain of the path space.

Let $T_{i\to j}$ be a bijective transformation that can transform a path generated from pixel i and to pixel j and conversely with its inverse, we extend the path space associated with pixel j as the union of several pixel domains:

$$\bar{\Omega}_j = \bigcup_{i=1}^{k} T_{i\to j}\left(\Omega_i\right) \tag{6.1}$$

Note that while we have increased the size of the path space, the additional paths do not contribute to the pixel value. This overall does not change the estimated integral. We can then rewrite the path integral of pixel j as a multi strategy integral on the extended domain (similar to the MIS integral) which lets appear the transformed samples $T_{i\to j}(\bar{x}_i)$, the Jacobian determinant that accounts for the change in density and some weighting functions $\bar{w}_i$ associated with each strategy:

$$\begin{aligned} I_j &= \int_{\Omega_j} f_j(\bar{x})\mathrm{d}\bar{x} \\ &= \int_{\bar{\Omega}_j} \underbrace{\sum_{i=1}^{k} \bar{w}_i(\bar{x})}_{=1} f_j(\bar{x})\mathrm{d}\bar{x} \\ &= \sum_{i=1}^{k} \int_{T_{i\to j}(\Omega_i)} \bar{w}_i(T_{i\to j}(\bar{x}))f_j(T_{i\to j}(\bar{x})) \left|\frac{\partial T_{i\to j}(\bar{x})}{\partial \bar{x}}\right| \mathrm{d}\bar{x} \end{aligned} \tag{6.2}$$

Note that some paths generated from pixel i and transformed with T may not contribute to pixel j due to occlusions. This is problematic as such samples with zero contributions are still accounted with MIS and classical weightings heuristics [Veach and Guibas 1995]. In fact, the integrand equals zero but the sample MIS weight is non null which increases the variance of the estimator. We further detail this problem and some possible solutions in the next sections. Additionally, we require the ability to transform sampled prefixes from one pixel to another and the Jacobian of the transformation in closed form which we will further explore for volumetric rendering. Finally, we discuss the solutions to build a common suffix path for several prefix.

6.2 The multi-view MIS variance problem

Variance of Importance Sampling

As previously explained when all samples drawn have a significant contribution then the estimator has low variance. On the contrary if a lot of samples have zero contribution then the estimator suffers from high variance. Note that when using importance sampling, we are relying on a single sampling strategy. For that reason its sampling domain should at least contain the compact support of the integrand – i.e. its positivity domain – to accurately estimate its integral. Furthermore, the closer the sampling domain is to the integrand support, the better is the integral estimate as illustrated in Figure 6.1.

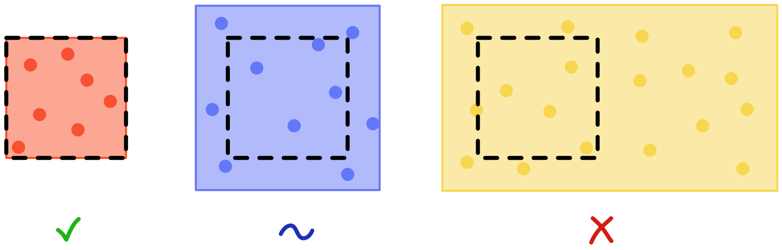


FIGURE 6.1 – Different sampling domains (colored regions) of an integrand positively defined on the dashed support. Distributing samples where the integrand is positively defined (left – red) results in a good estimate. Distributing samples a bit larger than the support (middle – blue) increases the variance but can still produce a reasonable estimate. Distributing most samples outside of the support (right – yellow) leads to high variance estimates and should be avoided.

Variance of Multiple Importance Sampling

Given multiple strategies combined with MIS, the union of the sampling domains should cover the support of the integrand. In fact, at least one of the available strategies should be non zero when the integrand is, to ensure covering the whole support. However nothing guarantees that every strategy significantly contributes within the pixel support. If the contributing strategies are not known beforehand, it is thus common to assign a fixed amount of samples to each of the available strategies, even to those that do not contribute at all as illustrated in Figure 6.2.

Multi-view overlapping domains

In the multiple view rendering context, each pixel corresponds to a specific integration domain. However, the specific domain of different pixels can overlap in the

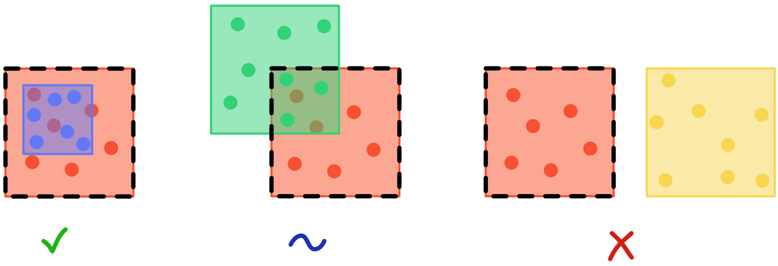


FIGURE 6.2 – Different sampling domains (colored regions) of an integrand positively defined on the dashed support. On the left several IS strategies (red and blue) distribute samples where the integrand is positively defined, which results in a good estimate. In the middle, several IS strategies (green and red) distribute samples a bit larger than the support of the integrand (middle – red and green), which increases the variance but still can produce a reasonable estimate. On the right we use several IS strategies (red and yellow) with some of them that distribute samples outside of the support, which lead to high variance estimates.

global path sampling space – i.e. the union of every pixels domains – which lets us reuse parts of sampled paths as depicted in Figure 6.3. But more importantly, the

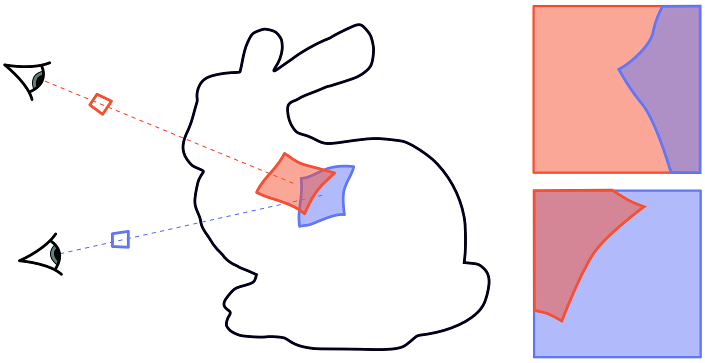


FIGURE 6.3 – Overlapping domains of different pixels in a simple bunny scene. The shared projected domain over the scene has a non trivial extent, since it is constrained by the scene geometry, volumes, etc.

field of view of a camera and the occlusions by objects of the scene decide the visibility of a point for a given pixel and shape the bounds of both its support and its own sampling domain. Thus, a scene point can be visible and sampled from one view but occluded for another, and it is hard to determine the extent of the path space shared between pixels, in particular in the presence of participating media and complex geometry.

The multi-view MIS variance problem

Still, we can use MIS to combine samples that have been generated from different pixels – i.e. different importance sampling strategies of the scene space. However, blindly reusing visible samples from any pixel sampler performs poorly in practice since some of them mostly distributes their samples outside of the desired area, but are still accounted in the estimator. The MIS combination hence has a large variance and reusing samples generated from other pixels is not beneficial. Such a configuration is depicted in Figure 6.4.

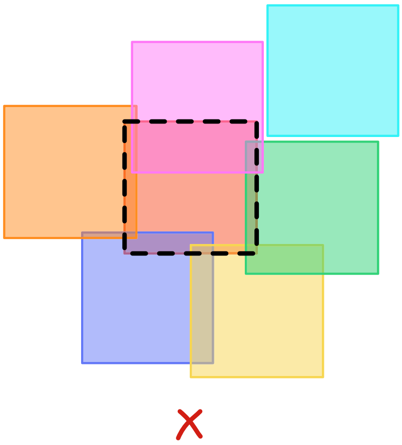


FIGURE 6.4 – Different domains (colored regions) associated with different pixels, combined with MIS to integrate an integrand positively defined on the dashed support. Some pixels better distribute samples than other regarding the target dashed region. However, some of them still have a large sampling area yielding zero contributions, thus produce a considerable amount of variance in the estimator.

Discussion

Several solutions come to mind to reduce the variance of naively reusing every available strategy and combining them using MIS, such as:

- pre-computing a list of overlapping pixels for each pixel,
- using an on-line estimate of the list of overlapping pixels for each pixel,
- stochastically selecting a set of pixels to reuse for each pixel to limit the variance (SMIS) [West et al. 2020],
- selecting a deterministic set of pixels to reuse for each pixel to limit the variance (e.g. the ones that best overlaps).

However none of these strategies is simple to set up, due to precomputations and memory footprint, and mostly build upon heuristics that are not guaranteed to reduce the variance of the MIS combination.

The problem there can be formulated as follows: can we discard the zero contributing samples and still get an accurate estimate of the integral ?

Proposed solution

Fortunately, there exists a family of weighted Monte Carlo estimators that are biased but consistent that circumvents these difficulties by ignoring contributions outside of a target support as depicted in Figure 6.5. In the context of multi view rendering, the target support (dashed) is always known, in fact it corresponds to the pixel sampling domain (the red strategy). We can use this knowledge to compare each reused sample – i.e. generated by other pixels – with the primary sampling strategy of the pixel using a weighting function that discards non contributing samples. We propose a new multiple strategy estimator in chapter 7, that benefits from MIS and weighted estimators, and allows us to reuse parts of paths between pixels without introducing further variance, without precomputations or additional memory.

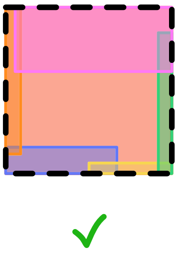


FIGURE 6.5 – Our proposed weighted multiple strategy estimator allows us to only take into account samples that have a non zero contribution and drastically reduces variance in comparison to MIS.

Motivating example

To further illustrate the problem, we compare several multiple strategy estimators in 1D in Figure 6.6 and Table 6.1 to demonstrate that MIS estimators may have significant variance in a simple multi strategy scenario.

Estimator	Biased	Consistent	Variance	Memory	Figure 6.6
$\hat{I}_k = \frac{1}{n} \sum_{i=1}^{n} w_{j,\text{mis}}(x_{ij}) \frac{f_k(x_{ij})}{c_j p_j(x_{ij})}$	No	Yes	High	No	(2,3,4,5)
$\hat{I}_k = \frac{1}{\hat{n}} \sum_{i=1}^{\hat{n}} w_{j,\text{mis}}(x_{ij}) \frac{f_k(x_{ij})}{c_j p_j(x_{ij})}$	Yes	No	Low	No	(7)
$\hat{I}_k = \frac{\hat{c}}{\hat{n}} \sum_{i=1}^{\hat{n}} w_{j,\text{mis}}(x_{ij}) \frac{f_k(x_{ij})}{c_j p_j(x_{ij})}$	Yes	Yes	Low	Yes	(6)
$\tilde{I}_k = \frac{\sum_{i=1}^{n} w_{j,\text{mis}}(x_{ij}) \frac{f_k(x_{ij})}{c_j p_j(x_{ij})}}{\sum_{i=1}^{n} w_{j,\text{mis}}(x_{ij})}$	Yes	No	High	No	(1)
$\tilde{I}_k = \frac{\sum_{i=1}^{\hat{n}} w_{j,mwis}(x_{ij}) \frac{f_k(x_{ij})}{p(x_{ij})}}{\sum_{i=1}^{\hat{n}} w_{j,mwis}(x_{ij})}$	Yes	Yes	Low	No	(8)

TABLE 6.1 – Formula and properties of the different multi-view estimators compared in Figure 6.6. We denote n as the total number of samples drawn, j as the strategy that sampled x_{ij}, c_j the relative number of samples per strategy, $\hat{n}$ as the estimated number of samples with non zero contributions, $\hat{c} = \frac{k}{\hat{k}}$ as a renormalization factor where $\hat{k}$ it the estimated number of contributing techniques – i.e. the techniques that generate at least one non zero sample – and k the total number of techniques.

6.3 The missing volumetric shift mappings

We introduced shifts mapping techniques for path reusing in chapter 5 and exposed that transformations in heterogeneous media have not been explored yet. With the

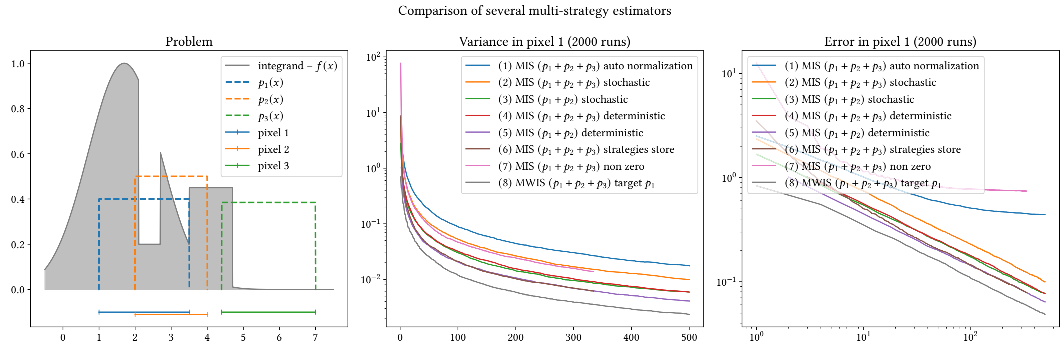


FIGURE 6.6 – We estimate the integral of the blue 1D pixel (pixel 1) with several Monte Carlo estimators and report their properties in Table 6.1. Estimators $(2, 3, 4, 5)$ are classical one-sample MIS estimators with varying numbers of strategies and stochastic or deterministic choice of the source strategy. Although those estimators are unbiased they still suffer from variance, especially when the strategy associated with pixel 3 is taken into account, and samples only zero contributions regarding pixel 1. Estimator (1) is a naive tentative of renormalizing the estimator with the sum of weights, which also leads to a biased and non consistent estimate that does not decrease the variance. Estimator (7) is a naive tentative of counting only non zero contributions to decrease the variance but it leads to a biased and non consistent estimate. Estimator (6) is an estimator that estimates the number of contributing strategies during the integration by maintaining a list of contributors for pixel 1 that is refined on the fly, and used to renormalize the result (cf. Table 6.1). This estimator combines our three source strategies and attains the same variance and error level as the best unbiased MIS estimator, but leads to a biased and consistent estimate. Finally, estimator (8) is the new estimator that we propose in chapter 7 which performs best in term of variance and error to combine these three strategies.

objective of extending path reusing methods, we propose in chapter 8 three new shift mappings which allow the reuse of null scattering chains in heterogeneous medium. We demonstrate the efficiency of our method in the context of multiple view rendering, in a setup similar as the one depicted in figure Figure 6.7.

6.4 The wavefront path construction

Most of state of the art methods for path reusing presented in chapter 5 sample one or several complete paths at once – e.g. one per pixel – and then reuse each path suffix for every base pixel with multi-sample estimators, such as MIS or SMIS multi-sample models. For example, gradient domain approaches samples gradients pairwise, discrete path reusing approaches sample several pixels within a tile, and multiple view rendering approaches sample a complete path before reconnecting towards other pixels.

However, this construction has several disadvantages. First, it requires selecting pixels sets a priori, using either a discrete (e.g. a tile in screen space) or a stochastic process (e.g. random selection in the vicinity of a pixel). This selection always rely on heuristics and a subset of pixels is not guaranteed to allow reuse between every pixels pairs (unless computed and stored). As explained before, it is difficult to evaluate the extent of shared path space between pixels. Second, constructing one complete path per selected pixel requires storage which is impractical with long medium paths. Third, these methods often suffer from artifacts due to the construction of the set of samples to reuse, for example in discrete path reusing the tiles used

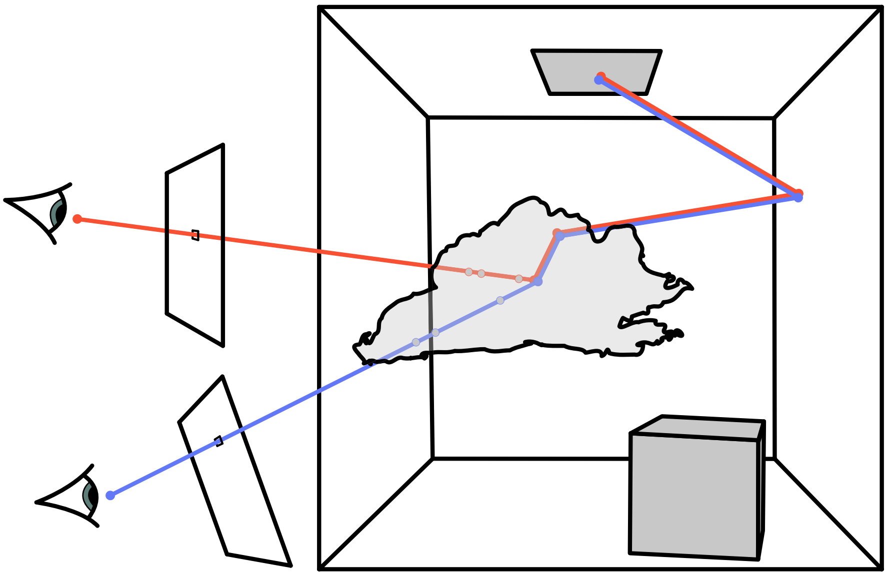


FIGURE 6.7 – Multiple view rendering through heterogeneous media in [Fraboni et al. 2022]. A base path traversing pixel i is constructed (in blue). Then an offset path (in red) is constructed by applying several transformations to the base path: the initial lens interaction is shifted to another camera lens, the pivot in the medium remains unchanged, the pixel position j is constrained by the lens and the pivot points. Additionally, several null interactions in the medium are shifted to the offset prefix using our new shift mappings and the suffix path remains unchanged.

may be visible in the final image [Bauszat et al. 2017, Figure. 2]. Last but not least, with MIS multi-sample every base path that has been traced has to be accounted, even ones with negligible contributions that increase the variance. These samples should rather not be accounted and rejected.

Instead, we propose a multi-view construction that traces a single path for several pixels, and rely on one-sample estimators to accumulate contributions on each camera [Fraboni et al. 2019, 2022]. In practice, a base prefix is traced to find a pivot point, then we construct several transformed prefixes that reconnects the pivot to other cameras, and finally after performing a similarity selection among prefixes a single suffix path is traced for every remaining prefix. This approach is similar to wavefront rendering [Fascione et al. 2018, Christensen et al. 2018, Burley et al. 2018], is simpler than the multi-sample construction as each sample is independent from the others and less information has to be stored. Additionally, the prefix selection step allows us to select the prefixes that gather the best contributions through a similarity selection of the scattering distribution associated with each prefix. In this way we ensure that the sampled direction to continue the path gather a significant contribution for several view directions. We detail our complete multi-view path construction method in chapter 9.

6.5 Discussion

We reviewed in this chapter several new solutions to address the problems that we described.

MIS variance reduction

We propose in chapter 7 new estimators to address the MIS variance problems in the presence of a large number of strategies that do not always distribute samples in the regions of interest.

Volumetric shift mappings

We propose in chapter 8 new shift mappings that allow tranformaing null scattering chains in arbitrary heterogeneous media as required for path reuse purposes. These new mappings can be used for example in the context of multiple view rendering or gradient domain rendering to extends these techniques for heterogeneous media.

Multi-view path construction

We detail our complete multi-view path construction method in chapter 9 which is simpler and requires less storage than classic multi-sample path reusing methods. We further show the results of our method in chapter 10.

Chapter 7

Weighted Monte Carlo estimators

In this chapter we first introduce a family of weighted estimators related to importance sampling, which are called *weighted importance sampling* (WIS) [Powell and Swann 1966, Spanier 1979, Spanier and Maize 1994, Bekaert et al. 2000]. They rely on an automatic renormalization of the samples mean and specific weighting functions, that can dramatically reduce the variance in comparison to standard IS when correctly used.

Secondly we propose a new family of multiple strategy estimators that combines both WIS and MIS. This new approach has two advantages: first, our estimator reduces the variance of the classic MIS estimator in presence of poor strategies, similarly to WIS. Second, our estimator optimally combines the available sampling techniques, similarly to MIS.

7.1 Weighted importance sampling with the ratio estimator

"If we really accepted the idea that a sample from one distribution is a sample from any distribution (if appropriately weighted) then we should not be surprised at the next two results stated below."

Trotter and Tukey

Instead of generating samples from a desired distribution p – i.e. using importance sampling – that is complex or impossible to sample, it is feasible to sample from a source distribution q that is convenient or simpler to sample and to reweight samples using a weighting function and a *ratio estimator*. The original *Weighted Uniform Sampling* [Powell and Swann 1966] method was only designed to transform uniform samples to arbitrary distributions. It has later been extended to handle arbitrary source densities and is known as *Weighted Importance Sampling* (WIS) [Spanier 1979, Spanier and Maize 1994, Bekaert et al. 2000], or as *Self Normalized Importance Sampling* (SNIS) [Owen 2013]. WIS estimators have been successfully used in rendering applications [Heitz et al. 2018, Bekaert et al. 2000, Keller 1996] to evaluate complex integrals.

The idea behind WIS is to virtually transform the samples from a source distribution q such that the weighted distribution of samples matches p. If the target density p is carefully chosen, the variance of the WIS estimator may be smaller than the variance of the IS estimator w.r.t q. Therefore, given a source distribution q the integral I can be approximated w.r.t p as follows:

$$\begin{aligned}
I &= \int_{\Omega} f(x) \mathrm{d}\mu(x) \\
&= \mathbb{E}_p \left[\frac{f(X)}{p(X)} \right] \\
&= \mathbb{E}_p \left[\frac{f(X)}{q(X)} \frac{q(X)}{p(X)} \right] \\
&= \mathbb{E}_p \left[\frac{f(X)}{q(X)} \frac{1}{\frac{p(X)}{q(X)}} \right] \\
&= \mathbb{E}_p \left[\frac{f(X)}{q(X)} \right] \mathbb{E}_p \left[\frac{1}{\frac{p(X)}{q(X)}} \right] \\
&\approx \frac{\mathbb{E}_q \left[\frac{f(X)}{q(X)} \right]}{\mathbb{E}_q \left[\frac{p(X)}{q(X)} \right]} \\
&\approx \frac{\mathbb{E}_q \left[w(X) \frac{f(X)}{p(X)} \right]}{\mathbb{E}_q \left[w(X) \right]} \quad \text{with} \quad w(X) = \frac{p(X)}{q(X)}.
\end{aligned} \tag{7.1}$$

where the subscript denotes the distribution that has to be sampled. The weighting function w compensate the use of the source PDF q. The associated Monte Carlo estimator, denoted $\tilde{I}_{\text{wis}}$, writes as a ratio of estimators:

$$\begin{aligned}
\tilde{I}_{\text{wis}} &= \frac{\frac{1}{n_1} \sum_{i=1}^{n_1} w(X_i) \frac{f(X_i)}{p(X_i)}}{\frac{1}{n_2} \sum_{j=1}^{n_2} w(X_j)} \\
&= \frac{\frac{1}{n_1} \sum_{i=1}^{n_1} \frac{f(X_i)}{q(X_i)}}{\frac{1}{n_2} \sum_{j=1}^{n_2} \frac{p(X_j)}{q(X_j)}}
\end{aligned} \tag{7.2}$$

It is worth noting that the sets of samples used in the numerator and the denominator can be identical, hence reducing the above equation to a weighted mean of samples:

$$\tilde{I}_{\text{wis}} = \frac{\sum_{i=1}^{n} w(X_i) \frac{f(X_i)}{p(X_i)}}{\sum_{i=1}^{n} w(X_i)} \tag{7.3}$$

instead of a simple mean like classical Monte Carlo estimators. The main benefit of Equation 7.3 is that we get rid of the normalization by the number of realizations n.

7.1.1 Properties

The WIS estimator is biased due to Jensen's inequality – $\frac{\mathbb{E}[X]}{\mathbb{E}[Y]} \leq \mathbb{E}\left[\frac{X}{Y}\right]$ – but consistent since the bias tends to zero [Spanier and Maize 1994, Spanier 1979, Powell and Swann 1966, Bekaert et al. 2000, Owen 2013] with rate $O\left(n^{-1}\right)$. Thus even though the WIS estimator is biased, the bias is asymptotically negligible, and makes the estimator asymptotically unbiased.

Note that Equation 7.1 is still valid if the weighting function $w(x)$ is known up to constant factor [Owen 2013]. The normalization is in fact done by the estimator in the denominator – the qualifier *self normalized* referring to this property.

This estimator is useful in situations where we cannot analytically construct samples from distribution p – i.e. for which we cannot invert the associated CDF – but we know the shape of p is closer to the integrand than the shape of q. The WIS estimator works best when the denominator in Equation 7.3 is close to the number of samples n, which means that the source PDF q and the target PDF p are somewhat similar.

7.1.2 Variance of WIS

The variance of the WIS estimator can be approximated with the delta method [Cochran 1977, Bekaert et al. 2000, Owen 2013] as a ratio estimator:

$$\mathbb{V}\left[\tilde{I}\right] = \frac{1}{n}\frac{\mathbb{E}_q\left[\left(w(X)\frac{f(X)}{p(X)} - w(X)I\right)^2\right]}{\left(\mathbb{E}_q\left[w(X)\right]\right)^2} \tag{7.4}$$

which can be estimated using the following Monte Carlo estimator:

$$\begin{aligned}\tilde{\mathbb{V}}\left[\tilde{I}\right] &= \frac{1}{n}\frac{\frac{1}{n}\sum_{i=1}^{n} w^2(X_i)\left(\frac{f(X_i)}{p(X_i)} - I\right)^2}{\left(\frac{1}{n}\sum_{i=1}^{n} w(X_i)\right)^2} \\ &= \frac{\sum_{i=1}^{n} w^2(X_i)\left(\frac{f(X_i)}{p(X_i)} - I\right)^2}{\left(\sum_{i=1}^{n} w(X_i)\right)^2}\end{aligned} \tag{7.5}$$

However this variance estimator is also biased and underestimate the true variance at a low number of samples.

A nice property of the WIS estimator is that it can approximately reach zero variance [Bekaert et al. 2000] even if samples are not distributed according to p as long as $p \propto f$, $p(x) > 0$ whenever $f(x) > 0$ and $q(x) > 0$ whenever $p(x) > 0$. The first two conditions are equivalent to the classic IS conditions to reach zero variance estimates. The last condition is required to get a correct estimate of I otherwise some parts of the integral would be missing or undefined and $\tilde{I}$ could not converge towards the true expected value. However, perfect importance densities are mostly not available in closed form.

7.1.3 Unbiasing ratio estimators

"The price we pay is an additional randomization inserted into the sequence and a possible increase in the mean squared error (MSE)."

McLeish

The main drawback of WIS is that the resulting estimator is biased. Several bias-correction methods exist however they all rely on additional stochastic processes [Talbot et al. 2005, McLeish 2011, Booth 2007, Bitterli et al. 2020] which we review below.

Random walks. Thomas E. Booth proposed using a geometric serie to approximate the reciprocal integral $\frac{1}{\mathbb{E}[Y]}$. An unbiased estimate of that geometric distribution can be simulated using random walks and Russian Roulette [McLeish 2011]. This method is practical and has successfully been used in several rendering applications [Zeltner et al. 2020, Qin et al. 2015, Bangaru et al. 2020], but the number of trials required to converge is unbounded and the additional variance introduced by the estimator of the reciprocal can be very large.

Discrete resampling. Recently, Bitterli et al. showed that the *Resampled Importance Sampling* (RIS) estimator [Talbot et al. 2005] is a bias corrected version of WIS.

The idea behind RIS is to generate a set of M samples from q and select one sample among them with discrete resampling [Talbot et al. 2005]. The selected sample is then approximately distributed according to the target distribution p. The number of candidates M interpolates from distribution source q ($M = 1$) to target distribution p ($M = \infty$).

The discrete resampling routine can be described as follows:

1. Generate M candidate samples from q
2. Evaluate for each candidate x_i the resampling weight $w(x_i) = \frac{p(x_i)}{q(x_i)}$
3. Build the discrete CDF of the sample weights
4. Random sample the discrete CDF to get a single sample with PDF:

$$\begin{aligned} \hat{p}_M(x_i) &= \frac{q(x_i)w(x_i)}{\frac{1}{M}\sum_{j=1}^{M} w(x_j)} \qquad \text{with} \qquad w(x) = \frac{p(x)}{q(x)} \\ &= \frac{p(x_i)}{\underbrace{\frac{1}{M}\sum_{j=1}^{M} w(x_j)}_{\text{bias correction term}}} \end{aligned} \tag{7.6}$$

The denominator in the PDF acts as a bias-correction term, hence the use of the classic unbiased Monte Carlo estimator is valid:

$$\hat{I} = \frac{1}{n}\sum_{i=1}^{n} \frac{f(x_i)}{\hat{p}_M(x_i)} \tag{7.7}$$

However applying unbiased RIS requires selecting a single sample among a set of candidates drawn from distribution q which becomes impractical when q is complex to sample. Indeed the resampling step may be either a huge waste of computational time or even unachievable when candidates have an infinitesimal chance to be sampled.

Stream resampling. The RIS technique requires storing all sample candidates and building the discrete CDF of the samples. A blind version of this algorithm that overcomes these preconditions has been proposed [Chao 1982, Bitterli et al. 2020] and further detailed in the book chapter [Marrs et al. 2021, chap. 23]. Its pseudocode is reported in Algorithm 3. This method makes use of a reservoir $R(x, w, s)$ that stores a sample x, an associated sample weight w, and the sum s of all samples weights that has been streamed in the reservoir so far.

Algorithm 3 Weighted Reservoir Sampling

```
function ReservoirSampling($R(x, w, s)$)
    $x \propto q$                              ▷ Generate candidate $x$ from $q$
    $w(x) \leftarrow \frac{p(x)}{q(x)}$        ▷ Evaluate the sample weight
    $R.s \leftarrow R.s + w(x)$                ▷ Update the sum of weights in the reservoir
    $\xi \leftarrow \mathcal{U}(0, 1)$
    if $\xi \leq \frac{w(x)}{R.s}$ then        ▷ Try to replace the sample currently in the reservoir
        $R.x \leftarrow x$                     ▷ Update sample
        $R.w \leftarrow w(x)$                  ▷ Update sample weight
    return $R$                                 ▷ Return the updated reservoir
```

This methods results in the same approximated PDF as RIS and share the same properties and limitations.

7.1.4 Results

We illustrate the WIS estimator and compare it to the IS estimator in 1D in Figure 7.1.

7.1.5 Discussion

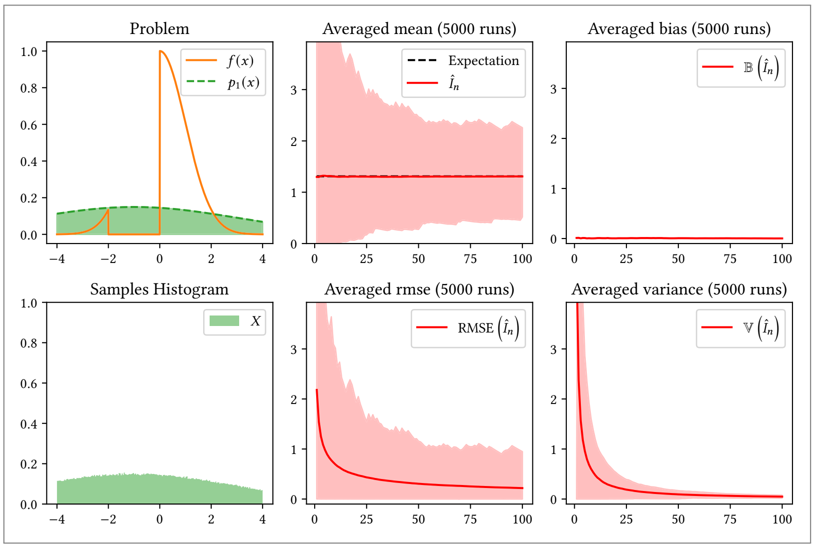
The WIS estimator is practical in several situations, e.g. when the target distribution p is difficult to sample but better fit to the integrand f, or when the number of samples n drawn is unknown a priori or costly to estimate the weighted mean of WIS preclude its evaluation, but is biased.

Although unbiased RIS and WRS successfully improves direct illumination sampling due to inexpensive light sampling techniques, we cannot rely on such resampling techniques for path reusing purposes, since generating candidate paths possibly through dense volumes and discarding them after resampling would be a waste of computational time.

Nevertheless, using WIS in path reusing application could be beneficial since it has less requirements than other estimators. In addition, path reusing virtually increases the number of path generated per integral and thus increase the number of samples per estimator. For that reason, the bias in WIS estimators will reciprocally vanish faster and hopefully will not be noticeable.

The main limitation of the WIS estimator is that it is restricted to a single source strategy. In the following section, we show how to extend the WIS framework to an arbitrary number of strategy.

Integration problem using IS with source density p_1



Integration using WIS with source density p_1 and target density p_0

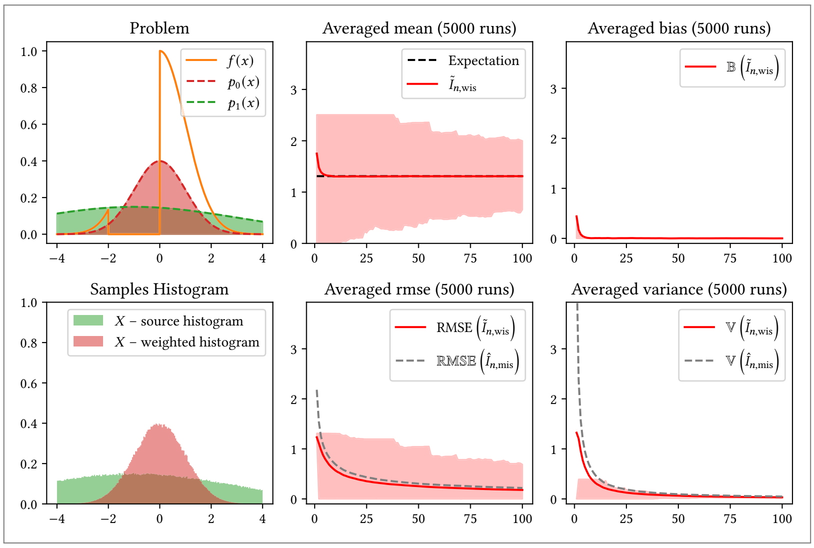


FIGURE 7.1 – Illustrative example of the difference between IS (top problem) and WIS (bottom problem). The integrand f in the top left plot (orange) is integrated using distributions p_1. However, given a known distribution p_0 (red), that is a priori a better fit to the integrand, MWIS reweights samples such that the resulting distribution perfectly matches p_0. This is well illustrated by the histograms of samples before (green) and after (red) reweighting. This results in a reduction of the error and variance at equal sample count, at the cost a slight bias that vanishes rapidly.

7.2 Multiple weighted importance sampling

As for importance sampling, the WIS estimator can be extended to combine several sampling strategies $\{1, \ldots, k\}$ with respective PDF $q_1, \ldots, q_k$ in order to evaluate the integrand w.r.t to a target density p (that possibly a mixture density itself) . To address this problem, we propose the *Multiple weighted importance sampling* (MWIS) estimator I_{mwis}, which consists of a linear combination of weighted estimators $I_{i,\text{mwis}}$ with associated normalization weights W_i:

$$I_{\text{mwis}} = \sum_{i=1}^{k} I_{i,\text{mwis}} \cdot W_i \tag{7.8}$$

The estimator $I_{i,\text{mwis}}$ associated with strategy i is, similarly to a WIS estimator, normalized by the expected value of the weighting function $w_{i,\text{wmis}}$ (cf. Equation 7.1):

$$I_{i,\text{mwis}} = \frac{\mathbb{E}_{q_i}\left[w_{i,\text{wmis}}(X)\frac{f(X)}{p(X)}\right]}{\mathbb{E}_{q_i}\left[w_{i,\text{wmis}}(X)\right]} \tag{7.9}$$

but we postpone the discussion of the choice of the ideal weighting function $w_{i,\text{wmis}}$ in the multiple strategy case (cf. subsection 7.2.1). The normalization weights W_i used in the linear combination arise naturally as a ratio estimator such that $\sum_i W_i = 1$, by taking:

$$W_i = \frac{\mathbb{E}_{q_i}\left[w_{i,\text{wmis}}(X)\right]}{\sum_{j=1}^{k} \mathbb{E}_{q_j}\left[w_{j,\text{wmis}}(X)\right]} \tag{7.10}$$

Finally, plugging the last two equations into Equation 7.8 leads to the MWIS formulation as a ratio of sum of integrals:

$$\begin{aligned} I_{\text{mwis}} &= \sum_{i=1}^{k} I_{i,\text{mwis}} \cdot W_i \\ &= \sum_{i=1}^{k} \frac{\mathbb{E}_{q_i}\left[w_{i,\text{mwis}}(X)\frac{f(X)}{p(X)}\right]}{\mathbb{E}_{q_i}\left[w_{i,\text{mwis}}(X)\right]} \cdot \frac{\mathbb{E}_{q_i}\left[w_{i,\text{mwis}}(X)\right]}{\sum_{j=1}^{k} \mathbb{E}_{q_j}\left[w_{j,\text{mwis}}(X)\right]} \\ &= \frac{\sum_{i=1}^{k} \mathbb{E}_{q_i}\left[w_{i,\text{mwis}}(X)\frac{f(X)}{p(X)}\right]}{\sum_{j=1}^{k} \mathbb{E}_{q_j}\left[w_{j,\text{mwis}}(X)\right]} \end{aligned} \tag{7.11}$$

which compares to the MIS formulation as a sum of integrals. The MWIS estimator is, similarly to WIS, a biased but consistent estimator due to Jensen's inequality.

7.2.1 Weighting heuristics

The choice of the weighting functions $w_{i,\text{mwis}}$ is crucial to get the best possible variance reduction similarly to MIS. For each sample x, the function $w_{i,\text{mwis}}$ serves two purposes: first reweighting the source density w.r.t the target density – i.e. comparably to WIS – and second combining the multiple strategies in an optimal manner – i.e. comparably to MIS. The former point takes into account that strategy i has been sampled with density q_i, but that the contribution is evaluated w.r.t the target density p. The latter point takes into account that strategy i has been selected among k available strategies to generate the sample x.

Hence we reasonably suggest that the MWIS weighting functions takes the following form:

$$w_{i,\text{mwis}}(x) = w_{i,\text{mis}}(x) \cdot w_{i,\text{wis}}(x) \tag{7.12}$$

which is a valid weighting heuristic within the MWIS framework thanks to the self-normalization of the estimator. The sufficient conditions are $p(x) > 0$ whenever $f(x) > 0$ and $\exists i, q_i(x) > 0$ whenever $p(x) > 0$. The WIS-related part of the weight is constrained by the source and target PDF used, hence:

$$w_{i,\text{wis}}(x) = \frac{p(x)}{c_i q_i(x)} \tag{7.13}$$

where q_i is the density that generated the sample x, and c_i the probability of selecting strategy i – i.e. the relative number of samples. The MIS-related part on contrary can be any heuristic following the MIS weighting conditions (cf. subsection 3.5.2). Hence, the MIS balance heuristic [Veach and Guibas 1995] is a good choice that takes into account the real local repartition of samples:

$$w_{i,\text{mis}}(x) = \frac{c_i q_i(x)}{\sum_{j=1}^{k} c_j q_j(x)} \tag{7.14}$$

where c_i is the probability of drawing a sample from the i-th strategy – i.e. the relative number of samples per strategy. Finally, using the balance heuristic the MWIS weighting function reduces to the simple form:

$$w_{i,\text{mwis}}(x) = \frac{p(x)}{\sum_{j=1}^{k} c_j q_j(x)} \tag{7.15}$$

The latter equation enlightens the fact that within MWIS we reweight the samples using a weighted mixture of distributions – i.e. the denominator – which is also the resulting distribution of MIS balance estimators. Still, MIS and MWIS weighting function are differents and serve different purposes.

7.2.2 MWIS estimator

The Monte Carlo estimator corresponding to Equation 7.11 writes as a weighted sum of estimators:

$$\tilde{I}_{\text{mwis}} = \frac{\sum_{i=1}^{k} \frac{1}{n_i} \sum_{j=1}^{n_i} w_i(x_{ij}) \frac{f(x_{ij})}{p(x_{ij})}}{\sum_{l=1}^{k} \frac{1}{n_l} \sum_{m=1}^{n_l} w_l(x_{lm})} \tag{7.16}$$

Similarly to the WIS estimator, the sets of samples used in the numerator and the denominator can be identical, hence turning the above equation into a weighted mean of samples:

$$\tilde{I}_{\text{mwis}} = \frac{\frac{1}{n} \sum_{j=1}^{n} w_i(x_{ij}) \frac{f(x_{ij})}{p(x_{ij})}}{\frac{1}{n} \sum_{j=1}^{n} w_i(x_{ij})} = \frac{\sum_{j=1}^{n} w_i(x_{ij}) \frac{f(x_{ij})}{p(x_{ij})}}{\sum_{j=1}^{n} w_i(x_{ij})} \tag{7.17}$$

where strategy i is stochastically chosen with probability c_i, or deterministically chosen with frequency $c_i = \frac{n_i}{n}$. The latter equation preclude the need to know or compute the number of realizations averaged in the estimator – even the realizations which have a zero weight. Any sample with weight $w_i(x) = 0$ is thus not taken into account, and the variance of the estimator is reduced accordingly. On contrary an unbiased MIS estimator would take into account these samples increasing its variance.

7.2.3 Variance of MWIS

Similarly to WIS, the variance of the MWIS estimator can be approximated with the delta method [Cochran 1977, Bekaert et al. 2000, Owen 2013] as a ratio estimator and shares the same properties: it can reach zero variance even if samples are not distributed according to p as long as $p \propto f$, $p(x) > 0$ whenever $f(x) > 0$ and $q(x) > 0$ whenever $p(x) > 0$, and estimating the variance with a Monte Carlo estimator underestimate the true variance at low number of samples as we can see in Figure 7.2 and Figure 7.3 (bottom row, column averaged variance).

7.2.4 Results

We illustrate our MWIS estimator and compare it to MIS estimator in 1D in Figure 7.2 and Figure 7.3. We additionally present a 2D density reconstruction example in Figure 7.4.

7.2.5 Discussion

The MWIS estimator makes it possible to sample from k arbitrary distributions with different domains, and to reweight samples such that a target distribution p is matched, further reducing the variance of the estimator if p is chosen close to the integrand. This is especially efficient when combining several importance samplers to estimates several integrals in a row and reusing samples between them.

Unbiasing MWIS. Similarly to WIS unbiasing the MWIS estimator can be computationally costly and requires additional sampling for random walks or resampling processes. This estimator is again related to the multi-sample RIS estimator [Bitterli et al. 2020, Talbot et al. 2005], as a biased but consistent self-normalizing version.

Application to path reusing. In the context of multiple view rendering, we consider each pixel as a possible sampling strategy of the scene. As explained before, combining all possible sampling techniques may fail or require additional informations with MIS.

Instead, our estimator combines multiples strategies – i.e. the different pixels that may find a pivot point – and reweights the samples w.r.t a chosen target density – which is none other than the base sampling technique of the pixel being integrated, since the target density better fits its integrand.

In this way, the contributions does not add further variance than the base technique – i.e. sampling from the pixel directly – thanks to the the reweighting. Additionally MWIS weights completely discard the samples that fall outside of the domain being integrated, hence it does not introduce further variance due to occluded prefixes, while MIS does.

We exploit this mechanism to reweight transformed samples as if they were directly sampled from the pixel they contribute to. Contrary to MIS that requires additional work (storage and/or computations) to discard strategies that poorly contribute to the estimator, MWIS has a much simpler form and naturally handles poor contributions such that they do not increase the variance. This makes MWIS highly practical with no additional storage requirement in a multi-view setting.

7.3 Conclusion

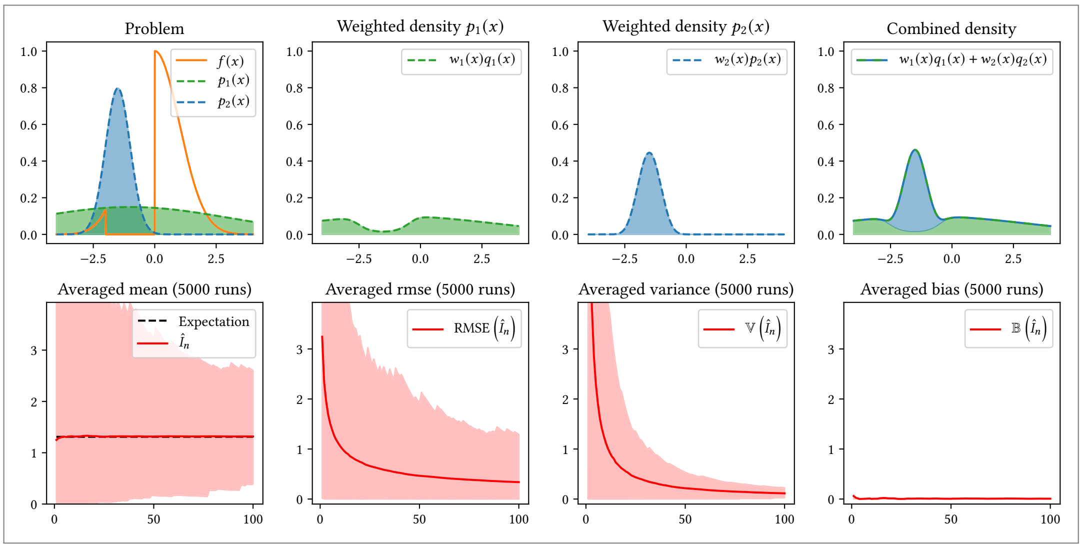
"Why did offline-rendering people publish several papers on control variates and (to our knowledge) never published a ratio estimator, which is simpler and better?"

Heitz, Hill, and McGuire

In this chapter we presented non classical Weighted Importance Sampling (WIS) ratio estimators and extended them to combine multiple source strategies in our Multiple Weighted Importance Sampling (MWIS) estimators.

This new family of estimators are similarly to WIS biased and consistent, but allow to reduce the variance in comparison to classical unbiased MIS estimators. It uses the knowledge of some target distribution that closely match the integrand to reweight the samples from different strategies. We take benefit from these estimators in our practical multiple view rendering to overcome MIS limitations, as we shall see in chapter 9 and chapter 10.

Integration problem using MIS balance heuristic and two source densities p_1 and p_2



Integration problem using MWIS balance heuristic with two source densities p_1 and p_2 and a target density p_0

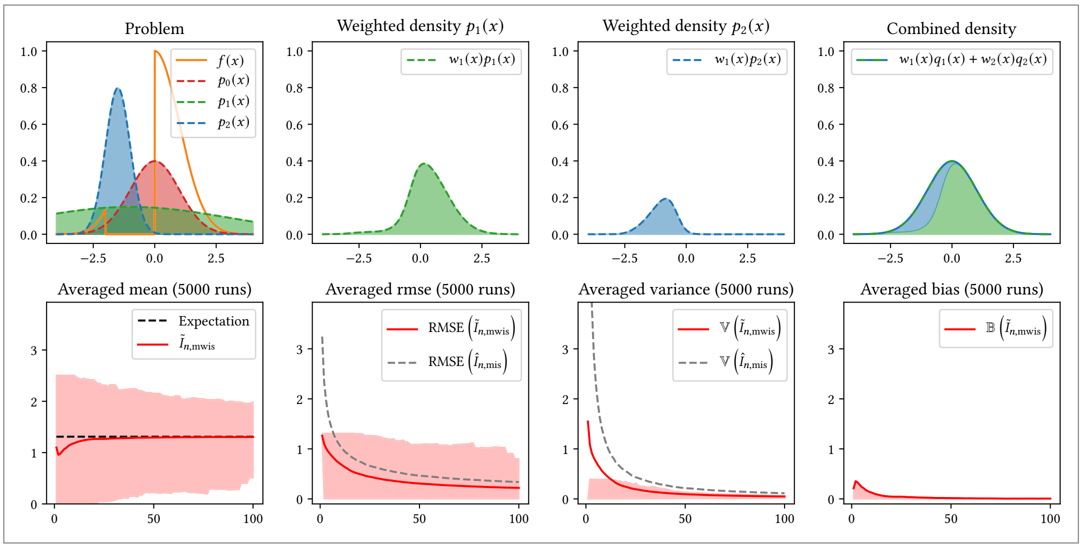
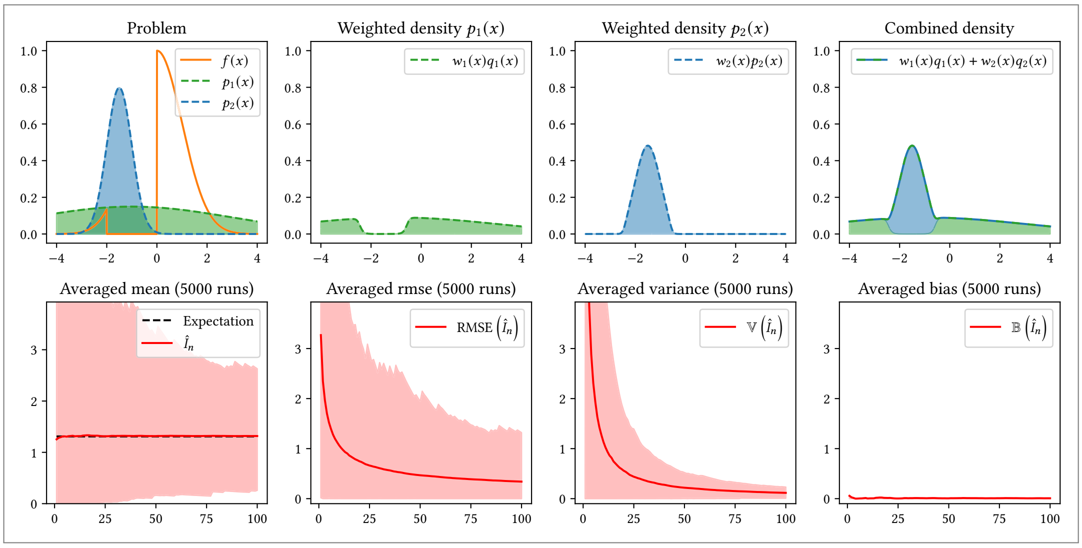


FIGURE 7.2 – Difference between MIS (top problem) and MWIS (bottom problem) to integrate f, shown in orange, using two distributions p_1 and p_2. Given an a-priori distribution p_0 (red) that better fits the integrand, MWIS reweights samples such that the combined distribution of strategies perfectly matches p_0. This results in a reduction of the error and variance, at the cost a slight bias that vanishes rapidly. Note that the averaged estimated variance of MWIS (bottom row, column averaged variance, red area) is largely underestimated at low sample count as its variations should follow the true variance (red curve).

Integration problem using MIS power heuristic and two source densities p_1 and p_2



Integration using MWIS power heuristic with source densities p_1 and p_2 and a target density p_0

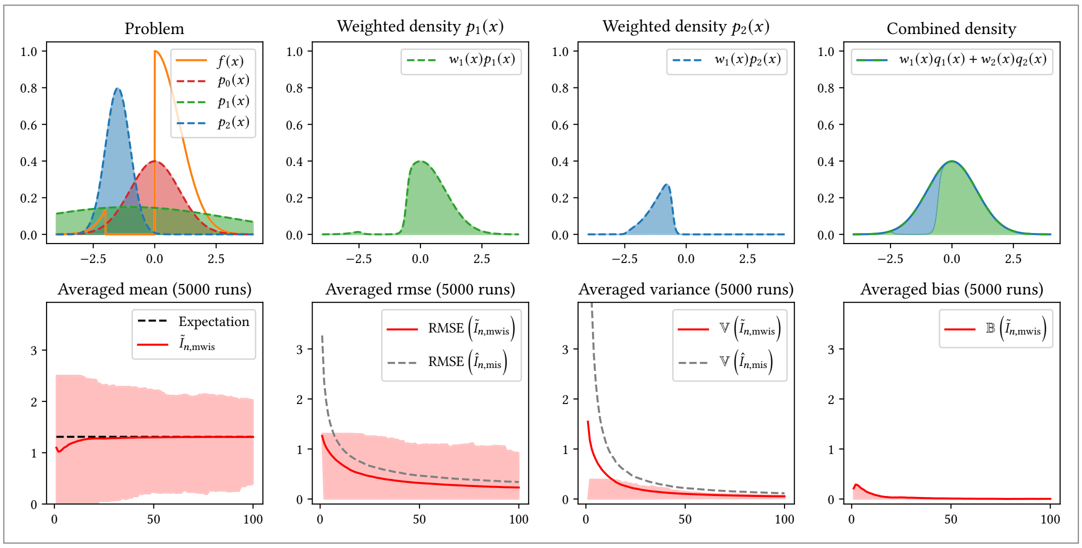


FIGURE 7.3 – Illustrative example of the difference between MIS (top problem) and MWIS (bottom problem) using the power heuristic. The integrand f in the top left plot (orange) is integrated using two distributions p_1 and p_2. However, given a known distribution p_0 (red), that is a priori a better fit to the integrand, MWIS reweights samples such that the combined distribution of strategies perfectly matches p_0. This results in a reduction of the error and variance at equal sample count, at the cost a slight bias that vanishes rapidly. Note that the bias is lower than MWIS combined with the balance heuristic, however the variance is greater, hence slightly increasing the RMSE. Nevertheless, both MWIS weighting functions (balance and power) results in lower error than MIS on this problem.

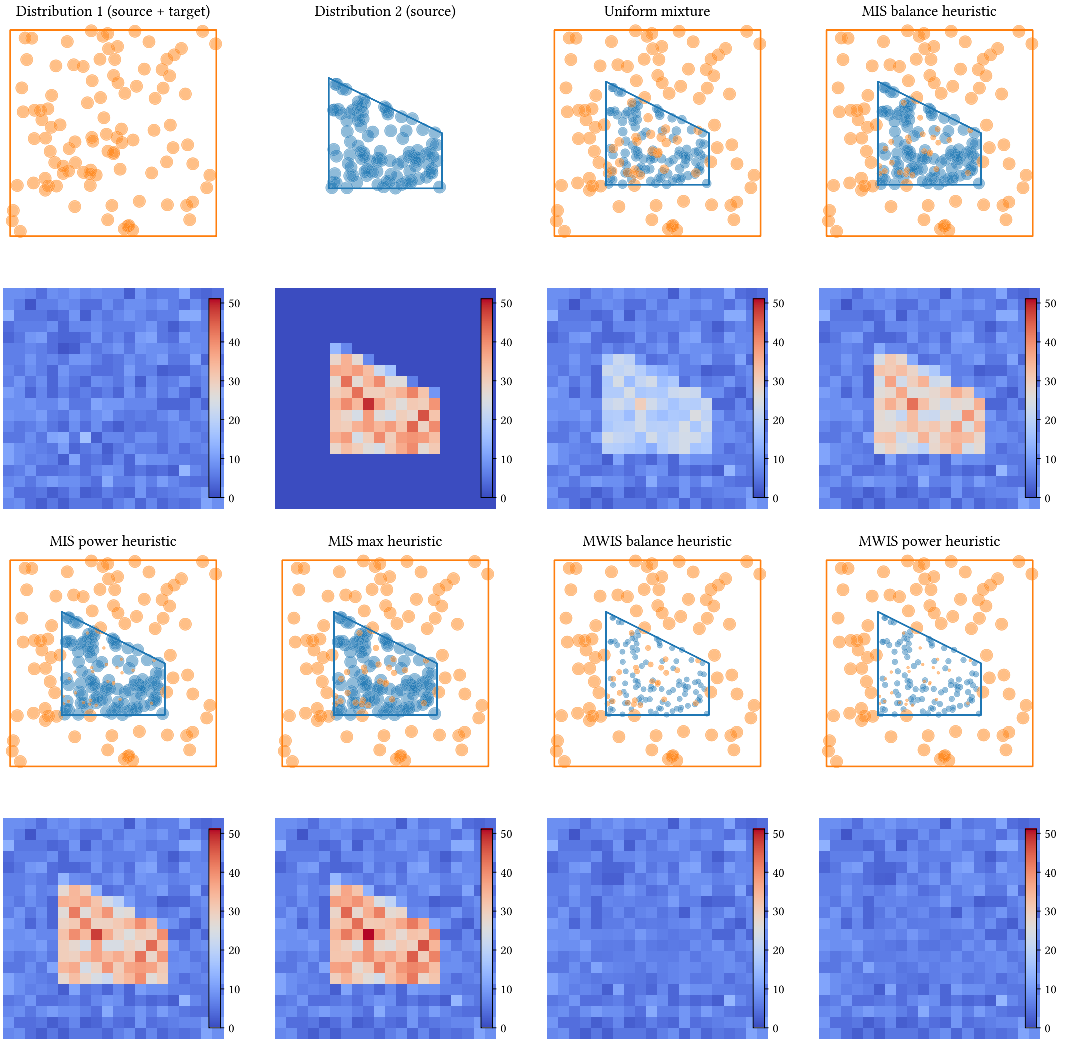


FIGURE 7.4 – Illustrative example of the resulting density of MWIS estimators in 2D. We draw samples uniformly from both the orange and blue polygons as our source densities. We aim at matching the uniform density over the orange polygon, as our target density. We show on the first and third rows a subset of 100 samples, for which the bubble radius corresponds to the weight assigned to each sample for comparison purposes. On the second and fourth rows we display the weighted histogram of 2500 samples for each multiple strategy estimator. Combining samples with uniform mixture, MIS balance, MIS power or MIS max results in a density that is preponderant in some areas of the domain: they conserves the peaks of the distributions. On the contrary the weighted histogram of our MWIS estimator perfectly match the target density. Furthermore, we can observe in both MWIS histograms the variance reduction due to the additional reweighted blue samples, in comparison to the histogram of native orange samples (top left).

Chapter 8

Null-scattering shift mappings

A stated before, no shift mapping has been proposed to shift a chain of null scattering interactions on a path segment composed of two real interaction endpoints. These shift mappings are essential to enhance path reusing applications with arbitrary heterogeneous volumes, enabling volumetric multiple view rendering or gradient domain volumetric path tracing. We propose in this chapter new null interactions shift mappings that allow path reusing in presence of heterogeneous medias leveraged from the null scattering framework [Miller et al. 2019].

8.1 Segment configuration

A path segment is a chain of random events for which both endpoints are real interactions. In between, there can be $\{0, \ldots, m\}$ null interactions in media. Our concern is to shift these null interactions from one base segment to a transformed segment for path reuse purposes, as depicted in Figure 8.1.

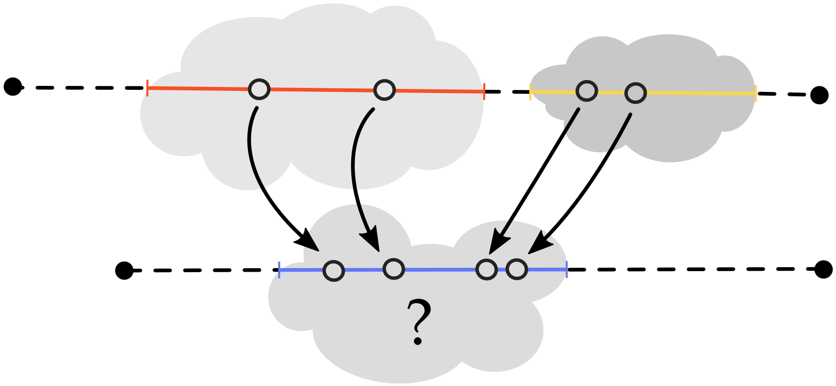


FIGURE 8.1 – Null scattering chain shift configuration. Given a base path segment (top) composed of several null interactions, we aim at deterministically shifting these interactions onto a target segment (bottom) that crosses media as well.

The shifted chain must have the same number of random events as the native one, in order to be defined on the same probability space. Note that recent techniques have been developed to reduce or expand the dimensionality of the probability space in the context of Metropolis mutations [Otsu et al. 2017, Bitterli et al. 2017, Pantaleoni 2017], but we did not explore such solutions which could eventually be applied to path transformations. Hence, if the base chain crosses medium but the shifted segment does not, we cannot construct a valid shift, since there is no medium space to shift null interactions. If neither of the chains cross any medium section, there is no null interaction to shift but the last interaction that remains unchanged. For these reason, when shifting from one segment to another we suppose that medium sections have been gathered beforehand.

The null scattering interactions may be shifted independently from each other or on contrary incrementally following the base construction ordering – i.e. from the first to the last sampled interaction. Hence, we explore several different approaches through global and local shifts on the chain.

We recall that a null interaction is sampled during the tracking process within the media (cf. section 4.2). The PDF of a null interaction x, given its free flight sample distance t, and its null probability P_n, writes:

$$p(x) = p(t)P_n(x) \tag{8.1}$$

Our goal is to define a bijective transformation such that the free flight sampled depth changes deterministically and fall inside the medium sections of the target segment. We also need its Jacobian in closed form to correctly take into account the change in density due to the transformation.

8.2 Copy-based shift mappings

We first present copy-based transformations that have been proposed and used in several previous work to shift interactions. However, these shift mappings lead to some failure cases that are depicted in Figure 8.2.

8.2.1 Raw depth copy

We first consider the naive approach of replicating the exact free flight distances t_i that have been sampled on the base segment for each null interaction x_i. Hence, from the shifted segment medium entry point x'_{min}, a first interaction is put at distance $t'_0 = t_0$ on the shifted segment direction ω'. Generalizing for each interaction gives the following formula:

$$x'_i = x'_{\text{min}} + \omega' \cdot t'_i \qquad \text{with} \qquad t'_i = t_i \tag{8.2}$$

and the shifted PDF then writes:

$$p(x'_i) = p(t'_i)P_n(x'_i) = p(t_i) \underbrace{\left|\frac{\partial t'_i}{\partial t_i}\right|^{-1}}_{=1} P_n(x'_i) = p(t_i)P_n(x'_i) \tag{8.3}$$

However, by reusing the same raw sample depth, some of the transformed volume interactions may even end up either outside of the medium section or farther than the segment endpoints. This approach has been proposed, and successfully used, in gradient domain methods [Gruson et al. 2018], although authors mention that some interactions may be invalid. But we aim at reusing interactions on segments that can be geometrically very different (e.g. two disjoint points of view observing the same pivot), hence this simple shift mapping may lead to a whole family of invalid shifted segment, as illustrated in Figure 8.2.

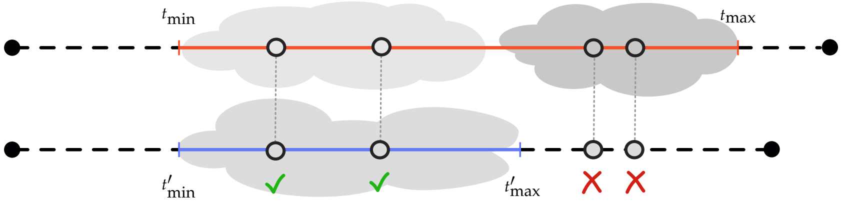


FIGURE 8.2 – Volumetric chains shift mapping failures. Given a base path segment (top) composed of several null interactions, the transformed interactions on the target segment (bottom) are constructed by reusing the same free flight distances – i.e. raw depths. This results in a family of invalid shifts, when shifted interactions fall outside of the medium sections or the segment. Not only depth copy but primary sample copy and optical depth copy shift mapping suffer from the same failure cases.

On the contrary, we want to enforce transformed null scattering interactions to fall inside medium sections, to increase our chances to find a valid shifted segment.

8.2.2 Primary sample copy

Another common approach is random sequence replay [Kelemen et al. 2002, Bitterli et al. 2017] – i.e. reusing the uniform random numbers that generated the samples. Hence, each shifted interaction x'_i is constructed using the random sample $u'_i = u_i$, that has been used to generate the base interaction x_i. Using the free flight sampling construction (cf. section 4.2) the shifted depth is computed as follows:

$$t'_i = -\frac{\log(1-u'_i)}{\bar{\mu}'} = -\frac{\log(1-u_i)}{\bar{\mu}'} = -\frac{\bar{\mu}}{\bar{\mu}'}\frac{\log(1-u_i)}{\bar{\mu}} = \frac{\bar{\mu}}{\bar{\mu}'}t_i \tag{8.4}$$

The shifted depth is thus a scaled version of the base one, and the interaction PDF writes:

$$p(x'_i) = p(t'_i)P_n(x'_i) = p(t_i)\left|\frac{\partial t'_i}{\partial t_i}\right|^{-1} P_n(x'_i) = p(t_i)\frac{\bar{\mu}'}{\bar{\mu}}P_n(x'_i) \tag{8.5}$$

But again, this way of transforming null interactions may result in invalid shifted interactions, for example when the medium majorant extinction coefficients are equals $\bar{\mu}' = \bar{\mu} \Leftrightarrow \frac{\bar{\mu}}{\bar{\mu}'} = 1$, the primary sample shift results in the same failure case as the raw depth copy mapping depicted in Figure 8.2.

Another drawback of both raw depth and PSS copy approaches is that there is no obvious solution to shift in presence of several disjoint medium sections easily.

8.2.3 Majorant optical depth copy

A last solution to consider in the family of shifts by copy, is the majorant optical depth copy $\bar{\tau}'(t') = \bar{\tau}(t)$, where the majorant optical depth of an interaction is copied to find the shifted interaction. Without going into further details, this scheme can lead to invalid shifts since shifted interactions are not bounded inside the target medium segments. For example using the medium section depicted in Figure 8.2 and equal majorants $\bar{\mu} = \bar{\mu}' = 1$ the majorant optical depth copy is equivalent to depth copy and results in the same failure case that precludes the construction of a whole family of shifts.

8.3 Scale-based shift mappings

We present three new shift mappings, summarized in Equation 8.6, that transform a base chain of null interactions, such that the resulting shifted interactions fall within the medium sections of the target segment. We construct shifted interactions using linear scales of the cumulated raw depth, the majorant optical depth and the primary samples, by deriving their correct stretching constants and their respective Jacobian determinants.

Shifts	Copy	Linear Scale
Primary Sample	$u' = u$	$u' = k \cdot u$ $k = \frac{1-\bar{T}(t'_{max})}{1-\bar{T}(t_{max})}$
Raw Depth	$t' = t$	$t' = k \cdot t$ $k = \frac{t'_{\text{max}}}{t_{\text{max}}}$
Majorant Optical Depth	$\bar{\tau}' = \bar{\tau}$	$\bar{\tau}' = k \cdot \bar{\tau}$ $k = \frac{\bar{\tau}'_{\text{max}}}{\bar{\tau}_{\text{max}}}$

(8.6)

8.3.1 Raw depth linear scale

A first solution to ensure the shifted interactions from the base segment to the target segment fall inside the medium sections is to copy the normalized cumulated distance of the base interaction. This mapping corresponds to a linear scale of the depth traversed in media as illustrated in Figure 8.3.

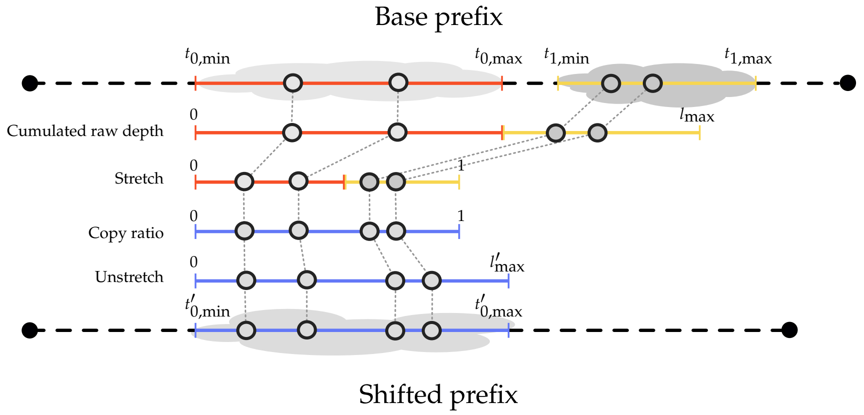


FIGURE 8.3 – Raw depth linear scale shift mapping. We compute the cumulated distance through volumes for each base interaction. After normalization by the cumulated volume depth of the base segment, we then copy the ratio to construct the shifted interaction. Multiplying by the cumulated volume depth of the target segment, gives the shifted cumulated depth. Finally, we compute the position of the shifted interaction by marching the target segment volume sections until reaching the desired cumulated depth.

Normalized cumulated distance

We define the cumulated distance l of an interaction at distance t as follows:

$$l(t) = \int_0^t V(x)dx \tag{8.7}$$

where $V(x)$ is a characteristic function that equals one if x is within a medium zero otherwise:

$$V(x) = \begin{cases} 1 & \text{if } \bar{\mu}(x) > 0, \\ 0 & \text{else.} \end{cases} \tag{8.8}$$

Note that the cumulated depth can be obtained by summing over each volume interval until reaching the distance t contained in section n_t:

$$l(t) = \sum_{i=1}^{n_t} \left(\min\left(t, t_{\max,i}\right) - t_{\min,i}\right) \tag{8.9}$$

where $t_{\min,i}$ is the entry depth of the i-eth volume section and respectively $t_{\max,i}$ is its exit depth.

Shift formulation

Now we define the shift mapping as a copy of the base interaction normalized cumulated depth:

$$\begin{aligned} \Leftrightarrow \quad & \frac{l'(t')}{l'_{\max}} = \frac{l(t)}{l_{\max}} \\ \Leftrightarrow \quad & l'(t') = \frac{l'_{\max}}{l_{\max}} \cdot l(t) \\ \Leftrightarrow \quad & l'(t') = k \cdot l(t) \end{aligned} \tag{8.10}$$

and $k = \frac{l'_{\max}}{l_{\max}}$ is a constant defined by the base and the target segment volume sections cumulated lengths. Expanding the above equation lets us express the shifted interaction depth t' as a function of the base interaction depth t:

$$\begin{aligned} & l'(t') = k \cdot l(t) \\ \Leftrightarrow \quad & \int_0^{t'} V'(x)\mathrm{d}x = k \cdot \int_0^t V(x)\mathrm{d}x \\ \Leftrightarrow \quad & \sum_{i=1}^{n_{t'}} \left(\min\left(t', t'_{\max,i}\right) - t'_{\min,i}\right) = k \cdot \sum_{j=1}^{n_t} \left(\min\left(t, t_{\max,j}\right) - t_{\min,j}\right) \\ \Leftrightarrow \quad & t' + a_1 = k \cdot [t + a_2] \\ \Leftrightarrow \quad & t' = k \cdot t + R \end{aligned} \tag{8.11}$$

where $a_1 = \sum_{i=1}^{n-1} \left(t'_{\max,i} - t'_{\min,i}\right) - t'_{\min,n}$, $a_2 = \sum_{j=1}^{m-1} \left(t_{\max,j} - t_{\min,j}\right) - t_{\min,m}$ and $R = k \cdot a_2 - a_1$. Applying the transformation for each base null interaction completes the construction of the shifted segment.

Jacobian determinant and PDF

The Jacobian of this mapping is straightforward to compute and is constant – i.e. global – on the whole segment:

$$\left|\frac{\partial t'}{\partial t}\right| = k = \frac{l'_{\max}}{l_{\max}} \tag{8.12}$$

Finally the PDF of a transformed interaction writes:

$$p(x'_i) = p(t'_i)P_n(x'_i) = p(t_i)\left|\frac{\partial t'_i}{\partial t_i}\right|^{-1} P_n(x'_i) = p(t_i)\frac{l_{\max}}{l'_{\max}}P_n(x'_i) \tag{8.13}$$

Discussion

In the presence of single base and target medium sections this mapping corresponds to a linear scale of the raw depth. This shift mapping allows a simple construction of the transformed interactions and a simple Jacobian evaluation that is global on the whole segment. However the resulting distributions of real and null interactions do not take into account the properties of the traversed medium section – e.g. the majorant extinction coefficients of each interval, or the traversed transmittance. The mapping only scales source distributions towards the target medium segment, which is not what we aim for in practice. We present in the next sections two local shift mappings – i.e. with varying Jacobian – that takes into account volume properties to better distribute the shifted interactions.

8.3.2 Majorant optical depth linear scale

One better solution is to take into account the majorant optical depth $\bar{\tau}$ of each null interaction in the volume sections, to accurately scale the interaction positions, further illustrated in Figure 8.4. Note that we use the majorant optical depth because the true optical depth does not in general have a closed form expression or one that is too complex to inverse. On the contrary, the majorant optical depth is piecewise linear on the medium sections and is thus always invertible in closed form.

Majorant optical depth

The majorant optical depth is obtained by integrating the majorant extinction coefficient from the medium entry to the interaction position:

$$\bar{\tau}(t) = \int_0^t \bar{\mu}(x)\mathrm{d}x \tag{8.14}$$

Note that the majorant is either constant or piecewise constant if several volume sections are crossed. Hence it can be obtained by summing over each interval of constant majorant $\bar{\mu}_i$ until reaching the interaction at distance t contained in section n_t:

$$\bar{\tau}(t) = \sum_i^{n_t} \bar{\mu}_i \left(\min\left(t, t_{\max,i}\right) - t_{\min,i}\right) \tag{8.15}$$

with $t_{\min,i}$ the entry depth of the i-eth volume section and respectively $t_{\max,i}$ the exit depth. In case of overlapping media, we replace the unique majorants by the sum of majorants in overlapping sections.

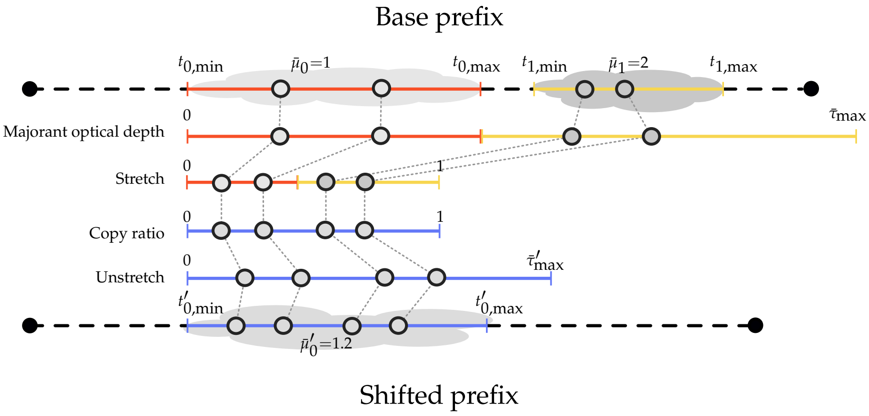


FIGURE 8.4 – Majorant optical depth linear scale shift mapping. We compute the optical depth through volumes for each base interaction. After normalization by the total optical depth of the base segment, we then copy the ratio to construct the shifted interaction. Multiplying by the total optical depth of the target segment, gives the shifted optical depth. Finally we compute the position of the shifted interaction by marching the target segment volume sections until reaching the desired optical depth.

Shift formulation

The mapping is then defined by copying the normalized majorant optical depth of the base interactions:

$$\begin{aligned} & \frac{\bar{\tau}'(t')}{\bar{\tau}'_{\text{max}}} = \frac{\bar{\tau}(t)}{\bar{\tau}_{\text{max}}} \\ \Leftrightarrow \quad & \bar{\tau}'(t') = \frac{\bar{\tau}'_{\text{max}}}{\bar{\tau}_{\text{max}}} \bar{\tau}(t) \\ \Leftrightarrow \quad & \bar{\tau}'(t') = k \cdot \bar{\tau}(t) \end{aligned} \tag{8.16}$$

where $k = \frac{\bar{\tau}'_{\text{max}}}{\bar{\tau}_{\text{max}}}$ is a constant defined by the base and the target segment volume sections maximum optical depth. From there we can express the position on the target segment t' as a function of t the distance on the base segment:

$$\begin{aligned} & \bar{\tau}'(t') = k \cdot \bar{\tau}(t) \\ \Leftrightarrow \quad & \int_0^{t'} \bar{\mu}'(x) \mathrm{d}x = k \cdot \int_0^{t} \bar{\mu}(x) \mathrm{d}x \\ \Leftrightarrow \quad & \sum_{i=1}^{n_{t'}} \bar{\mu}'_i \left(\min\left(t', t'_{\text{max},i}\right) - t'_{\text{min},i}\right) = k \cdot \sum_{j=1}^{n_t} \bar{\mu}_j \left(\min\left(t, t_{\text{max},j}\right) - t_{\text{min},j}\right) \\ \Leftrightarrow \quad & \bar{\mu}'_{n_{t'}} \cdot t' + a_1 = k \cdot \left[\bar{\mu}_{n_t} \cdot t + a_2\right] \\ \Leftrightarrow \quad & t' = k \cdot \frac{\bar{\mu}_{n_t}}{\bar{\mu}'_{n_{t'}}} \cdot t + R \end{aligned} \tag{8.17}$$

where $a_1 = \sum_{i=1}^{n_{t'}-1} \bar{\mu}'_i \left(t'_{\text{max},i} - t'_{\text{min},i}\right) - \bar{\mu}'_{n_{t'}} \cdot t'_{\text{min},n_{t'}}$, $a_2 = \sum_{j=1}^{n_t-1} \bar{\mu}_j \left(t_{\text{max},j} - t_{\text{min},j}\right) - \bar{\mu}_{n_t} \cdot t_{\text{min},n_t}$ and $R = \frac{k \cdot a_2 - a_1}{\bar{\mu}'_n}$. Applying the transformation for each base null interaction completes the construction of the shifted segment.

Jacobian determinant and PDF

The Jacobian determinant of the complete transformation writes:

$$\left|\frac{\partial t'}{\partial t}\right| = k \cdot \frac{\bar{\mu}_{n_t}}{\bar{\mu}'_{n_{t'}}} = \frac{\tau'_{\max}}{\tau_{\max}} \cdot \frac{\bar{\mu}_{n_t}}{\bar{\mu}'_{n_{t'}}} \tag{8.18}$$

Finally the PDF of a transformed interaction writes:

$$p(x'_i) = p(t_i) \left|\frac{\partial t'_i}{\partial t_i}\right|^{-1} P_n(x'_i) = p(t_i) \frac{\tau_{\max}}{\tau'_{\max}} \cdot \frac{\bar{\mu}'_{n_{t'}}}{\bar{\mu}_{n_t}} P_n(x'_i) \tag{8.19}$$

Discussion

In the presence of single base and target medium sections this mapping corresponds to a linear scale of the majorant optical depth. Hence, the resulting shifted distributions of events correspond to a piecewise stretch of the source distributions. Note that as opposed to the raw depth mapping, the Jacobian determinant is local for each interaction due to the presence of the base section majorant $\bar{\mu}_{n_t}$, associated with the base interaction at distance t, and the target section majorant $\bar{\mu}'_{n_{t'}}$ associated with the shifted interaction at distance t'.

8.3.3 Primary sample linear scale

The common free-flight sampling approach is unbounded, hence for a given uniform random number u the resulting sampled distance t lies between zero and infinity. However the primary sample can be renormalized over a bounded interval using the integral of the combined transmittance over this interval. Hence another working approach is to linearly scale the primary random sample such that the resulting free flight distance is bounded in the target volume interval. This results in a shift that perfectly maps a source majorant distribution to a target majorant distribution as we shall see in the results section 8.4. Again this shift substitutes an ideal mapping between the real and null densities that cannot be solved in closed form.

Primary random sample

The primary random sample associated with a free flight sampled distance t is expressed as a function of the combined transmittance $\bar{T}$:

$$u = \int_0^t \bar{T}(x)\mathrm{d}x = 1 - \bar{T}(t) \tag{8.20}$$

Note that the majorant is either constant or piecewise constant if several volume sections are crossed. Hence the combined transmittance can be obtained using a product of the transmittance over each interval of constant majorant $\bar{\mu}_i$, or using the inverse exponential of the piecewise optical depth, until reaching the interaction at distance t contained in section n_t:

$$\begin{aligned}\bar{T}(t) &= \exp\left(-\bar{\tau}(t)\right) \\ &= \exp\left(-\sum_i^{n_t} \bar{\mu}_i \left(\min\left(t, t_{\max,i}\right) - t_{\min,i}\right)\right) \\ &= \prod_i^{n_t} \exp\left(-\bar{\mu}_i \left(\min\left(t, t_{\max,i}\right) - t_{\min,i}\right)\right)\end{aligned} \tag{8.21}$$

with $t_{\min,i}$ the entry depth of the i-eth volume section and respectively $t_{\max,i}$ the exit depth. In case of overlapping media, we replace the unique majorants by the sum of majorants in overlapping sections.

Shift formulation

The mapping is then defined by copying the normalized random sample of the base interactions:

$$\begin{aligned}\frac{u'}{u'_{\max}} &= \frac{u}{u_{\max}} \\ \Leftrightarrow u' &= \frac{u'_{\max}}{u_{\max}} \cdot u \\ \Leftrightarrow u' &= k \cdot u\end{aligned} \tag{8.22}$$

where $k = \frac{u'_{\max}}{u_{\max}}$ is a constant defined by the integral of the combined transmittance from the last interaction to the segment endpoint, on the base and respectively the target segment, which is related to bounded transmittance sampling. From there we can express the position on the target segment t' as a function of t the distance on the base segment:

$$\begin{aligned}& u' = k \cdot u \\ \Leftrightarrow & 1 - \bar{T}(t') = k\left(1 - \bar{T}(t)\right) \\ \Leftrightarrow & \bar{T}(t') = 1 - k\left(1 - \bar{T}(t)\right) \\ \Leftrightarrow & \exp\left(-\bar{\tau}(t')\right) = 1 - k\left(1 - \bar{T}(t)\right) \\ \Leftrightarrow & \bar{\tau}(t') = -\log\left(1 - k\left(1 - \bar{T}(t)\right)\right) \\ \Leftrightarrow & \bar{\mu}'_{n_{t'}} \cdot t' = -\log\left(1 - k\left(1 - \bar{T}(t)\right)\right) - a_1 \\ \Leftrightarrow & t' = \frac{-\log\left(1 - k\left(1 - \bar{T}(t)\right)\right) - a_1}{\bar{\mu}'_{n_{t'}}} \\ \Leftrightarrow & t' = \frac{-\log\left(1 - k + k\exp(-\bar{\tau}(t))\right) - a_1}{\bar{\mu}'_{n_{t'}}}\end{aligned} \tag{8.23}$$

where $\bar{\tau}(t') = \bar{\mu}'_{n_{t'}} \cdot t' + a_1$ and $a_1 = \sum_{i=1}^{n_{t'}-1} \bar{\mu}'_i \left(t'_{\max,i} - t'_{\min,i}\right) - \bar{\mu}'_{n_{t'}} \cdot t'_{\min,n_{t'}}$. Applying the transformation for each base null interaction completes the construction of the shifted segment.

Jacobian determinant and PDF

The Jacobian determinant of the complete transformation writes as:

$$\left|\frac{\partial t'}{\partial t}\right| = \frac{\bar{\mu}_{n_t}}{\bar{\mu}'_{n_{t'}}} \frac{k}{(k-(k-1)\exp(\bar{\tau}(t)))} \tag{8.24}$$

Finally the PDF of a transformed interaction writes:

$$p(x'_i) = p(t_i)\left|\frac{\partial t'_i}{\partial t_i}\right|^{-1} P_n(x'_i) \tag{8.25}$$

Discussion

The scaling factor k of the primary sample is the ratio of transmittance normalization terms over the base and target volume intervals after the last interaction. It depends on volumes parameters in each traversed section, and on the last interaction position thus is a *more* local transformation than the optical depth shift. Similarly to the majorant optical depth shift, it performs a local transformation that requires computing the majorant extinction coefficients in each traversed section. Although this approach is a valid shift that perfectly match the target majorant density, it requires evaluating several exponentials and induces a slightly more complicated Jacobian to compute without significantly improving the distribution of the shifted samples regarding the target distributions of real and null interactions (cf. Figure 8.5).

8.4 Results

We illustrate our shift mapping operators in Figure 8.5, Figure 8.6 and Figure 8.7.

They allow for remapping distributions from an arbitrary source medium segment, to an arbitrary target medium segment. However, by construction none of them perfectly matches the true distributions of real interactions or null interactions, since we can not construct the ideal mappings from these two distributions to the target ones. Some methods such as optimal transport or dynamic numerical inversion could solve the problem of distribution remapping. However, these methods would be too computationally intensive for our use case where we deal with a large number of medium segments and shift operations.

Instead, our mappings rely on geometric quantities available in close form (e.g. raw distances, combined transmittance and majorant extinction coefficient). Hence, we do not expect them to perfectly match the target distributions (real or null).

8.4.1 Raw depth shift mapping

In practice, the raw depth shift mapping results in shifted distributions that closely resemble the source distributions. Although this is not desired for a general purpose mapping as the target distributions may be very different, this mapping works best when the source and target distributions are close to each others.

8.4.2 Primary sample shift mapping

The primary sample shift mapping instead better spread the shifted distributions to perfectly match the target majorant density. However, we aim at matching the true distribution of real or null interactions for path reuse purpose (we shift null and real interactions), which is not achieved with this mapping. Furthermore, it sometimes leads to oversampled regions at the end of the segment. Additionally its Jacobian is slightly more complex to evaluate than our other mappings.

8.4.3 Majorant optical depth shift mapping

The optical depth shift mapping is balanced between both: it better stretches than the raw depth by taking into account the local majorant coefficient, and most of the time better resembles the true distributions (real and null) than the primary sample shift mapping. Finally, it has a simple evaluation and Jacobian formula. We hence use the latter in our path reuse application as it is a good tradeoff between the quality of shifted real distributions and the complexity of evaluation.

8.5 Conclusion

We presented several shifts mappings that ensure valid transformations of null scattering chain from one base medium segment to a target medium segment. They result in remapped distributions of different quality that have their strength and drawbacks regarding the distributions we aim to match. In our work we mostly use our mapping to shift null interactions, but they could be used to shift real interactions if needed.

We demonstrate in chapter 10 the efficiency of our mappings in our volumetric multiple view rendering application. They could further be used in other path reusing applications to extend them to heterogeneous participating media.

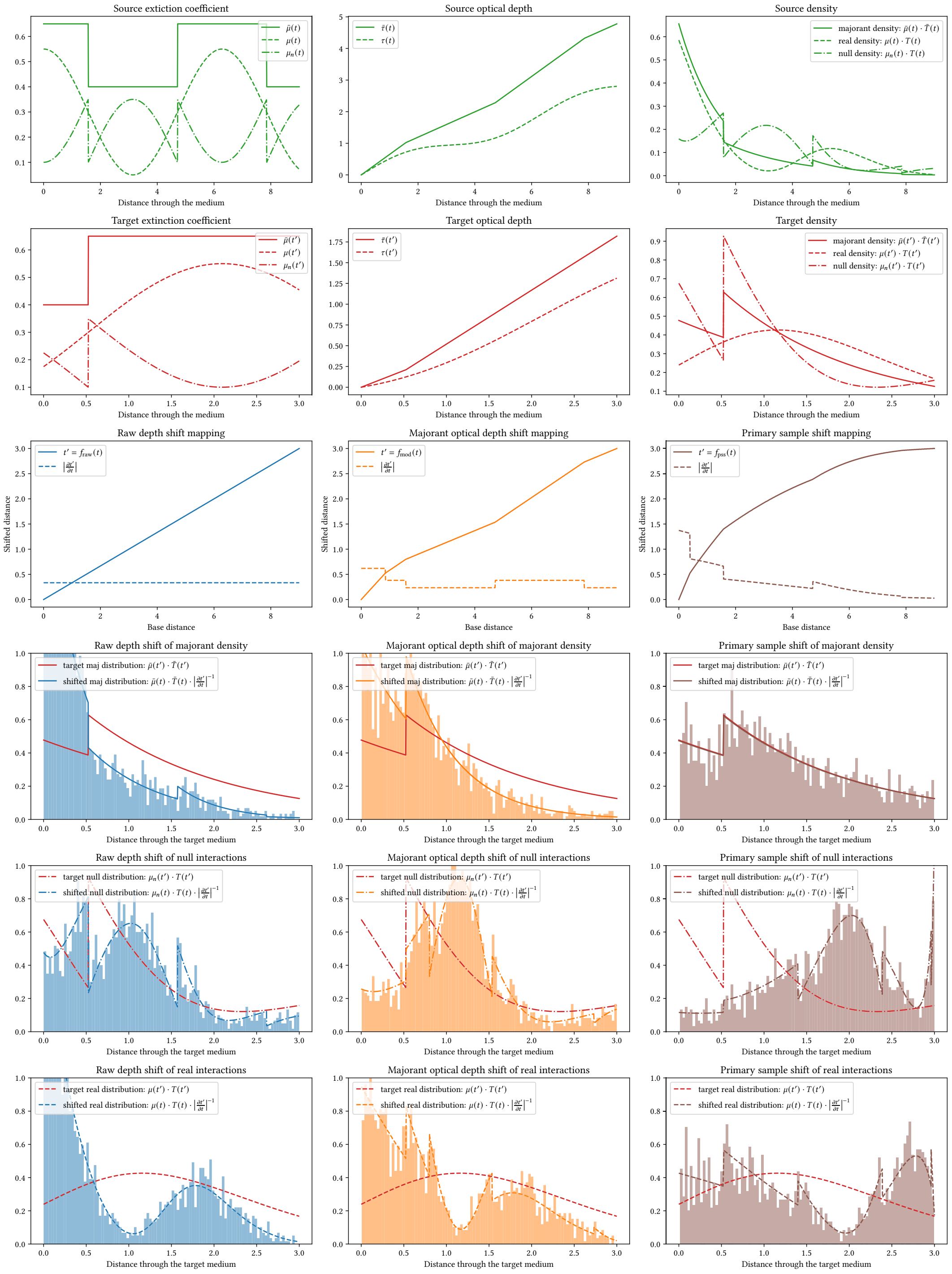


FIGURE 8.5 – A source segment on the domain $[0,9]$ is depicted in the top row (green curves) with its associated majorant, null and real extinction coefficients (top left), majorant and real optical depths (top center) and distributions of majorant, real and null interactions (top right). Note that we use varying extinctions that simulate heterogeneous mediums. A second target segment the domain $[0,3]$ is depicted in the second row (red curves) with its associated extinction coefficient (left), optical depth (center) and distributions of real and null event (right). Our shift mappings operators aims at mapping the free flight distributions of the source segment towards the target segment. The first mapping operates on raw depths (blue curves), results in a rescaling of the depth with a constant Jacobian (third row left), and thus in a stretch of the source distributions (fourth to bottom rows left). The second mapping operates on optical depths (orange curves), results in a rescaling of the majorant optical depth with a piecewise constant Jacobian (third row center), and thus in a piecewise stretch of the source distributions (fourth to bottom rows center). The last mapping operates on primary samples (brown curves), results in a rescaling of the primary samples with a more complex Jacobian (third row right), and thus more complex transformation of the source distributions (fourth to bottom rows right) but still resemble the base shape.

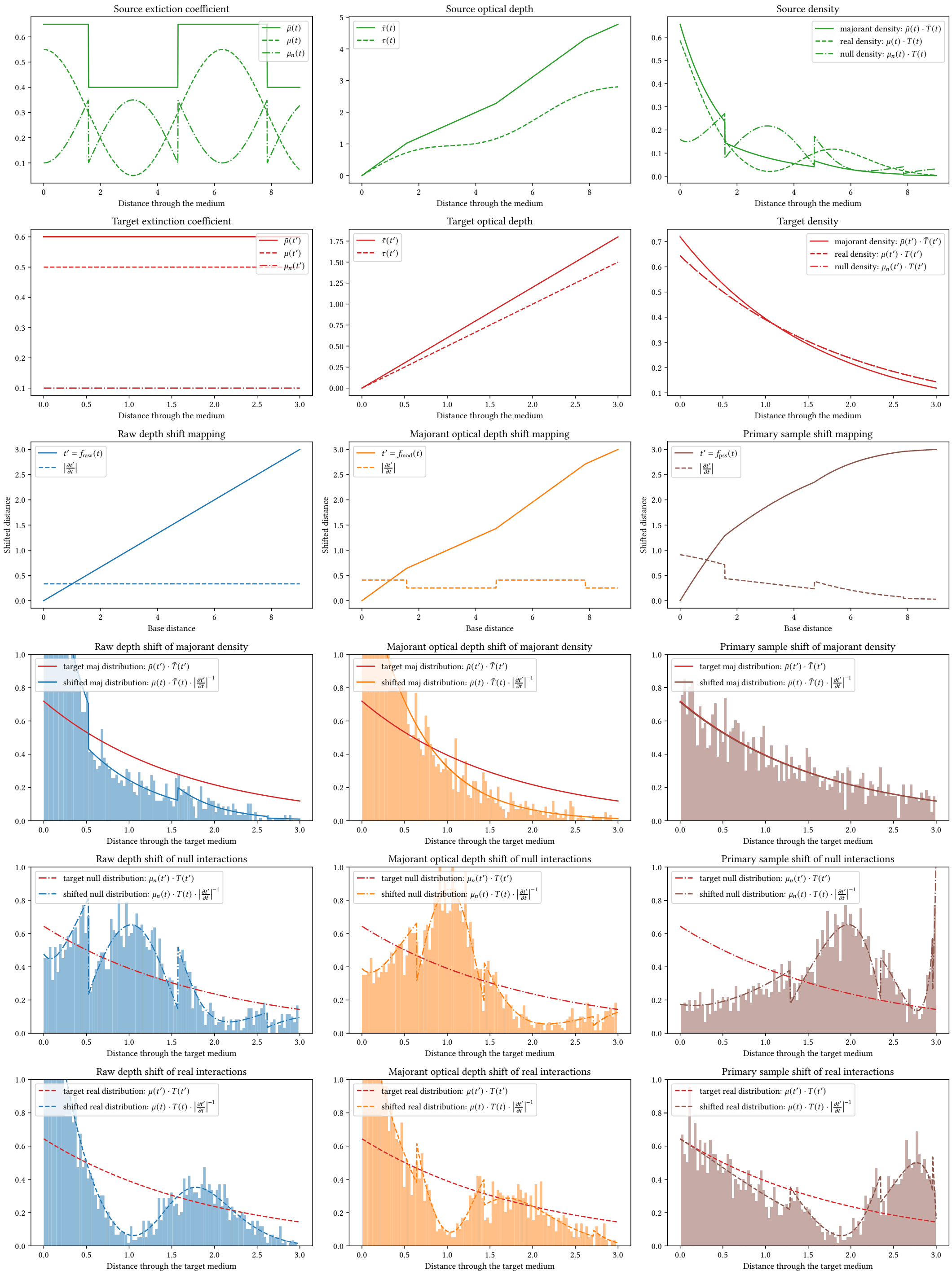


FIGURE 8.6 – Similar example as in Figure 8.5 with different target densities.

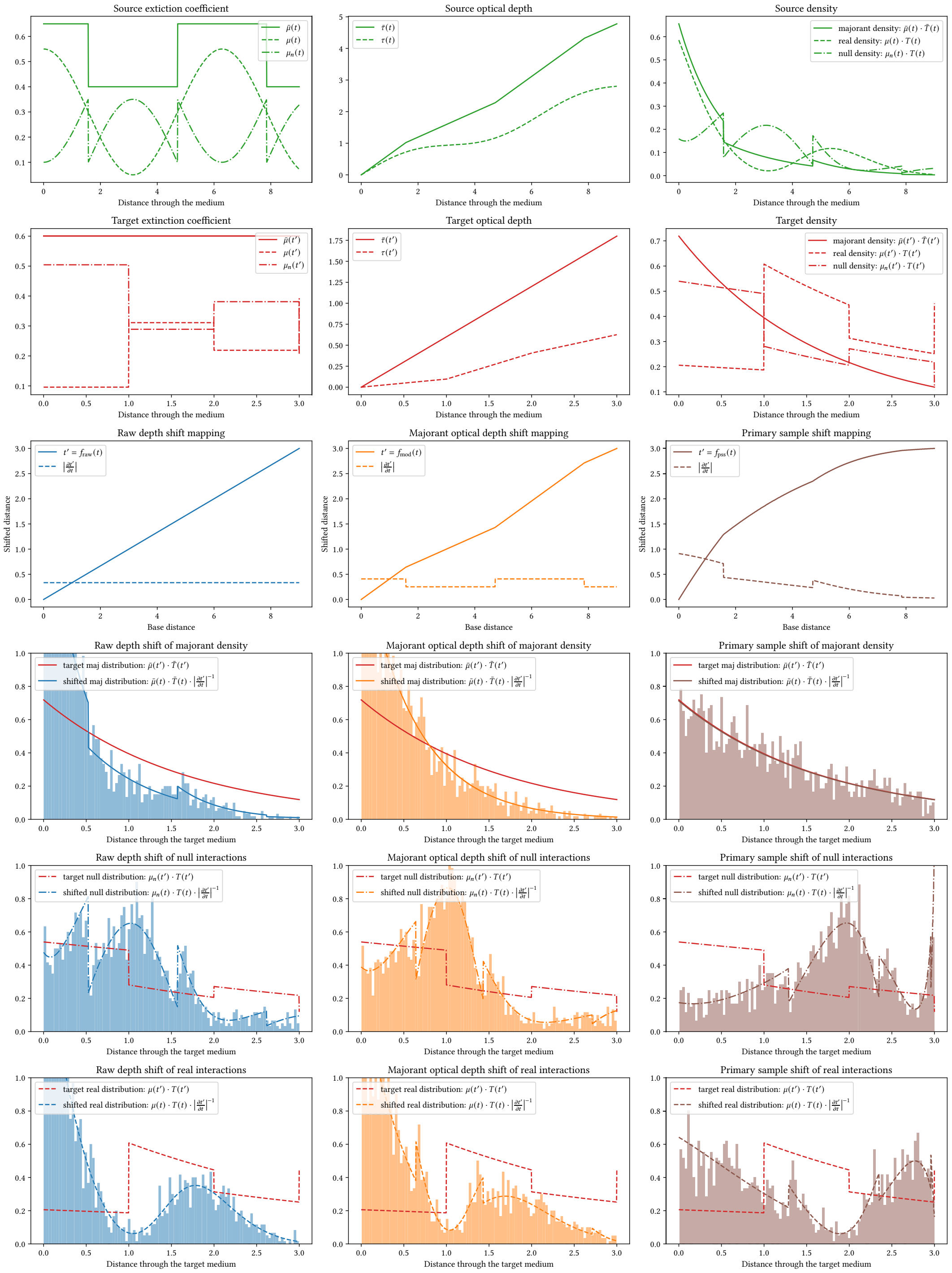


FIGURE 8.7 – Similar example as in Figure 8.5 with different target densities.

Chapter 9

Practical multi-view rendering

Construction

In this chapter we detail a practical application of our contributions to multiple view rendering covered in [Fraboni et al. 2019, 2022]. Our construction builds upon our MWIS estimator, our new volumetric chains shift mappings and a measure of the distance between scattering distributions to share significant subpaths.

9.1 Multi-view path construction

Our goal is to simultaneously reduce the error in multiple path traced images of the same scene. Instead of rendering each image separately, we share sampled paths between multiple pixel integrals. We hence benefit from shared path construction and shading with a small overhead. The efficiency gain comes from the construction of one single suffix path contributing to several pixels.

Our multi-view rendering technique first samples a *base* prefix path from a pixel. We then reconnect the first real (non-null, surface or volume) interaction of the base prefix – the *pivot* – to every other camera for which the path is expected to contribute (i.e. this interaction is visible from the new point of view as it is not occluded, falls within its view frustum, occurs during overlapping exposure time, and the scattering function has a significant component towards the new camera). We formulate a transformation of all interactions that have occurred during the simulation along the base prefix path towards the other pixels, in such a way that valid *shifted* prefixes can be built, as illustrated in Figure 9.1. Finally, we sample a direction to trace a path *suffix* that is shared among the set of selected prefixes. For each pixel integral, we combine all base and shifted prefix paths that reach the pixel using our MWIS estimator.

9.1.1 Base prefix sampling

We first generate a ray by importance sampling a camera shutter, lens and film. If a medium boundary is found, delta tracking is used to sample interaction distances across the media and event sampling to determine the interaction types [Novák et al. 2018, Miller et al. 2019]. If a null interaction is sampled, we continue the sampling routine until we either leave the medium or a real interaction is found (medium or surface).

Finally, the complete path prefix is a *chain* of random events: a film interaction, a lens interaction, $\{0, \ldots, m\}$ null interactions in the medium, and a last interaction

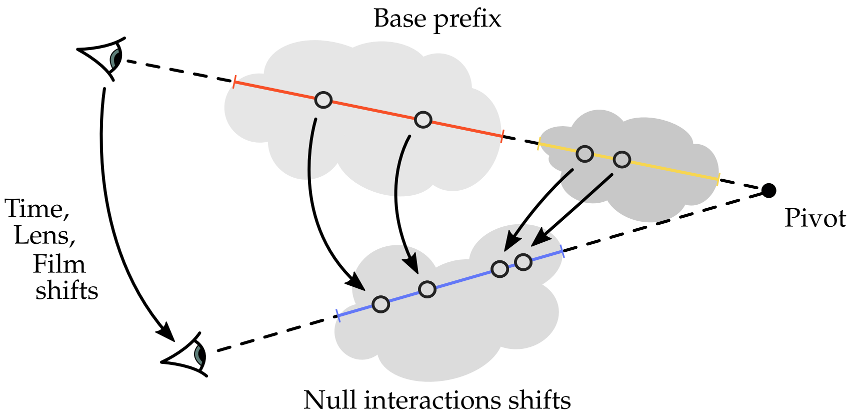


FIGURE 9.1 – Prefix chains shift configuration

either on a surface or a real medium interaction (cf. Figure 9.1). This last interaction, which we will refer to as the *pivot*, is the key to our path reuse work since it can be visible from other points of view. Before constructing the path suffix, we connect every camera to the pivot by shifting (i.e. transforming) all the events composing this *base* prefix to form *shifted* prefixes, as we describe next.

9.1.2 Prefix shift

Once a base prefix is generated and the pivot is found, we try to build shifted prefixes that connect the pivot to other cameras, the *targets* as depicted in Figure 9.1. The shifted chain must have the same number of random events as the base one, in order to be defined on the same probability space. We start by shifting the base lens interaction to the lens of a target camera. We then check that the pivot is not occluded and falls within the target camera view frustum and exposure time. Next, the remaining sampled events along the base prefix are shifted to build another prefix connecting a new lens event to the pivot interaction. A ray is traced from the target lens to the pivot to collect the target medium sections. If only one of the base or shifted prefix crosses medium sections, we cannot construct a valid shift because the number of random events differs between the two chains. If neither of the prefixes crosses any medium section, the shift is complete as there is no medium interaction, but only a surface interaction at the pivot that remains unchanged.

Our goal is then to define a bijective transformation of the base medium interactions, such that the shifted medium interactions fall within the target medium sections of the target segment. The medium interaction depths can thus be mapped back and forth to unique and valid interactions on base and target prefixes.

We also need the Jacobian of the transformation in closed form to correctly account for the change in density. All required transformations are described in the following sections, where index i denotes the pixel used to generate the base prefix and index j denotes another pixel and its shifted prefix.

Shutter time shift

We construct paths at fixed time since dealing with object motion would require reevaluating occlusions and shading on the complete path due to moving geometry

– the cost of the time shift would be equivalent to constructing a full path. Hence the base shutter time is reused $t_j = t_i$, and the Jacobian of the shift is $\left|T'_{i\to j}\right| = 1$. Finally, we can compute both the transformed pdf and the pdf as if the time sample was sampled from camera j directly:

$$\begin{aligned} &\text{pdf transformed from } i \text{ to } j: && p_{i\to j}(t_j) = p_i(t_i)\left|T'_{i\to j}\right|^{-1} = p_i(t_i) \\ &\text{pdf base from } j \text{ (target):} && p_{j\to j}(t_j) = p_j(t_j) \end{aligned} \tag{9.1}$$

Note that the second pdf is used in our MWIS weights as the target strategy.

Film shift

The lens event is shifted by reusing the random numbers that sampled the base lens interaction, similarly to the work of Schwarzhaupt and Fraboni et al.. This is equivalent to scaling the base lens interaction, such that the shifted one fall within the support of the target lens. Considering a thin lens model and given a base lens interaction $x_{\text{lens},i}$, this results in the simple transformation:

$$x_{\text{lens},j} = \frac{r_j}{r_i} \cdot x_{\text{lens},i} \quad \text{with Jacobian} \quad \left|T'_{i\to j}\right| = \frac{r_j^2}{r_i^2} \tag{9.2}$$

where r_i and r_j are base and target lens radii. Finally, we can compute both the transformed pdf and the pdf as if the lens and film samples were sampled from camera j directly (for MWIS weights):

$$\begin{aligned} &\text{pdf transformed from } i \text{ to } j: && p_{i\to j}(x_{\text{lens},j}) = p_i(x_{\text{lens},i})\left|T'_{i\to j}\right|^{-1} = p_i(x_{\text{lens},i})\tfrac{r_i^2}{r_j^2} \\ &\text{pdf base from } j \text{ (target):} && p_{j\to j}(x_{\text{lens},j}) = p_j(x_{\text{lens},j}) \end{aligned} \tag{9.3}$$

Lens shift

After shifting the lens event, the film position is then constrained by the refraction of the ray connecting the lens to the pivot. Hence the film interaction does not require a shift in itself but we have to compute its position $x_{\text{film},j}$ from $x_{\text{lens},j}$ and $x_{\text{pivot},j}$, as in light tracing or bidirectional methods. Finally, the change of measure between the base film plane and the shifted one is accounted with the following Jacobian [Lehtinen et al. 2013]:

$$\left|T'_{i\to j}\right| = \frac{g(x_{\text{lens},i}, x_{\text{pivot},i})}{g(x_{\text{lens},i}, x_{\text{film},i})} \cdot \frac{g(x_{\text{lens},j}, x_{\text{pivot},j})}{g(x_{\text{lens},j}, x_{\text{film},j})} \tag{9.4}$$

where $g(a,b) = \frac{D(a,b)}{\|b-a\|^2}$ and $D(a,b) = |\vec{\omega_{ab}} \cdot \vec{n_b}|$ if b is on a surface else $D(a,b) = 1$.

Finally, we can compute both the transformed pdf and the pdf as if the lens sample was sampled from camera j directly (for MWIS weights):

$$\begin{aligned} &\text{pdf transformed from } i \text{ to } j: && p_{i\to j}(x_{\text{film},j}) = p_i(x_{\text{film},i})\left|T'_{i\to j}\right|^{-1} \\ &\text{pdf base from } j \text{ (target):} && p_{j\to j}(x_{\text{film},j}) = p_j(x_{\text{film},j}) \end{aligned} \tag{9.5}$$

Null interactions shift

We then apply our *majorant optical depth* shift mapping (see chapter 8) for every null interactions along the base segment between the camera lens and pivot. Thus, by construction the shifted interactions are ensured to fall within the target medium sections.

We copy the normalized majorant optical depth from a null interaction at depth t_i on the base segment to find the shifted position t_j on the target segment. The code that performs the transformation is reproduced in Appendix B. Denoting n (resp. m) the index of the medium section of interaction t_i (resp. t_j) on the base segment (resp. on the target segment), the Jacobian determinant of the transformation writes:

$$\left|\frac{\partial t_j}{\partial t_i}\right| = \frac{\tau_{\max,j}}{\tau_{\max,i}} \cdot \frac{\bar{\mu}_{n,i}}{\bar{\mu}_{m,j}} \tag{9.6}$$

Finally, we can compute both the transformed pdf and the pdf as if the null interaction was sampled on the target segment directly (for MWIS weights):

$$\begin{array}{ll} \text{pdf transformed from } i \text{ to } j\text{:} & p_{i\to j}(t_j) = p_i(t_i)\left|T'_{i\to j}\right|^{-1} = p_i(t_i)\frac{\tau_{\max,i}}{\tau_{\max,j}} \cdot \frac{\bar{\mu}_{m,j}}{\bar{\mu}_{n,i}} \\ \text{pdf base from } j \text{ (target):} & p_{j\to j}(t_j) = p_j(t_j) \end{array} \tag{9.7}$$

Pivot shift

The last interaction can be either on a surface or in a medium but its location stays fixed in space, hence we copy its position: $x_{\text{pivot},j} = x_{\text{pivot},i}$ and $\left|T'_{i\to j}\right| = 1$. Finally, we can compute both the transformed pdf and the pdf as if the pivot was sampled on the target segment directly (for MWIS weights):

$$\begin{array}{ll} \text{pdf transformed from } i \text{ to } j\text{:} & p_{i\to j}(x_{\text{pivot},j}) = p_i(x_{\text{pivot},i})\left|T'_{i\to j}\right|^{-1} = p_i(x_{\text{pivot},i}) \\ \text{pdf base from } j \text{ (target):} & p_{j\to j}(x_{\text{pivot},j}) = p_j(x_{\text{pivot},j}) \end{array} \tag{9.8}$$

Jacobian and pdf evaluation

The complete shift generates a new chain in path space that connects the target camera to the pivot. The Jacobian of the chain transformation is the product of each independent shift Jacobian:

$$\left|T'_{i\to j}\left(\bar{x}_i\right)\right| = \prod_{k=0}^{n}\left|T'_{i\to j}\left(x_{i,k}\right)\right| \tag{9.9}$$

and the joint pdf of the base interactions writes:

$$p_i\left(\bar{x}_i\right) = \prod_{k=0}^{n} p_i\left(x_{i,k}\right) \tag{9.10}$$

Finally the pdf of the path $\bar{x}_i$ from pixel i transformed to path $\bar{x}_j$ for pixel j is given by:

$$p_{i\to j}\left(\bar{x}_j\right) = p_i\left(\bar{x}_i\right)\left|T'_{i\to j}\left(\bar{x}_i\right)\right|^{-1} \tag{9.11}$$

which is further used to compute the MWIS weights. Similarly the pdf used as a target in MWIS weights – i.e. as if the samples were sampled directly from pixel j – writes:

$$p_{j\to j}\left(\bar{x}_j\right) = p_j\left(\bar{x}_j\right)\underbrace{\left|T'_{j\to j}\left(\bar{x}_j\right)\right|^{-1}}_{=1} = \prod_{k=0}^{n} p_j\left(x_{j,k}\right) \tag{9.12}$$

9.1.3 Computing path suffix

After finding a base pivot point and constructing the possible prefixes by shifting all events along the base prefix, we can evaluate direct and indirect illumination at once for all cameras with successful shifts.

Direct illumination

The direct illumination part is simple in the sense that sampling light sources does not generally depend on prefix directions, using for example *next event estimation*. We hence apply regular light source sampling: we generate a light sample, compute the common part of the contribution for all prefixes, and finally multiply it by the scattering function and the path throughput associated to each prefix j independently. We can further improve the results by combining multiple direct illumination sampling techniques [Miller et al. 2019].

In case of a directional component in direct illumination, for example using path guiding or product importance sampling, the importance sampling of the light sources depends on the prefix directions. Hence, a single sample taken for one prefix is not guaranteed to contribute to every other prefixes. We rather propose to use the approach described in the next section for selecting prefixes that are close to the base prefix using the product importance pdf instead of the scattering distribution pdf.

Indirect illumination sampling

Indirect illumination is more complex. We need to sample a single scattering direction to start building the shared path suffix, but the scattering distribution function at the pivot interaction is generally view-dependent. This is notably the case for mostly glossy and perfect specular materials, and highly anisotropic medium phase functions. Hence, we have to ensure that every prefix contribution is significant. For that reason, we discard shifted prefixes whose scattering distribution function differs too much from the base one.

The path suffix sampling process should not increase the variance of the base pixel, nor of the target pixels. We chose to proceed with stochastic acceptance or rejection of shifted prefixes. We accept a candidate shifted prefix if the associated scattering distribution at the pivot point is similar to that of the base prefix (Figure 9.2). Doing so requires to compare scattering distributions, which is a difficult problem.

We first proposed an ad-hoc similarity heuristic [Fraboni et al. 2019] only designed for surface materials that gave good results at selecting similar bsdf. However, we found a more robust and generic approach using a true statistical distance. Thus,

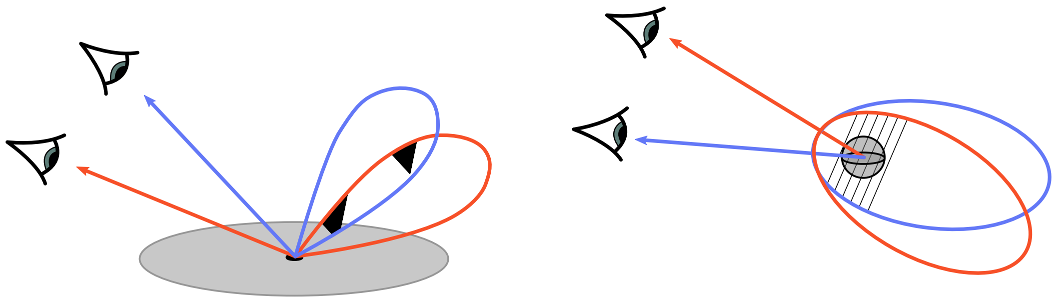


FIGURE 9.2 – At the pivot point, the scattering function of all prefixes should be similar to the base one to avoid increasing the variance. We use an approximate measure of the shared volume of both distributions (hatched region) to accept or reject the prefix. Our approximation handles both surface and volume scattering distributions.

we use an acceptance probability based on a discrete total variation (TV) approximation [Fraboni et al. 2022] between pairs of scattering distributions. This TV approximation is evaluated on the fly and produces a well-defined distance for general distributions, including any bsdf and phase function. We evaluate the TV distance on a small set of directions $\{\omega_i\}_i$, and their density for both distributions, before normalizing both discrete pdf sets and computing the approximate TV distance as:

$$\mathrm{TV}(p, q, \{\omega_i\}_i) = \frac{1}{2} \sum_i \left| \frac{p(\omega_i)}{\sum_k p(\omega_k)} - \frac{q(\omega_i)}{\sum_k q(\omega_k)} \right| \tag{9.13}$$

This approximation is inexpensive for a small number of directions. In practice, we use two directions chosen according to the type of scattering distribution at the pivot interaction. For surface interactions, we use the mirror directions of both viewing directions, that is the peak of the distribution in Cook-Torrance microfacet models [Cook and Torrance 1982]. For strong retroreflective surfaces, we can use the viewing directions directly. For medium interactions, we use the forward directions of both viewing directions if the phase is forward scattering, and the backward directions in case of backward scattering, that are the peaks of the distribution.

This approximation overestimates the true TV distance but gives correct estimates for rough single and bi-layered materials (i.e. rough plastics, conductors, dielectrics, coatings) and phase functions with varying anisotropy.

We compare every candidate prefix to the base prefix and reject those with large TV distances. The acceptance probability of a candidate prefix reads: $P_{i\to j} = 1 - \mathrm{TV}\left(p_i, p_j, \{\omega_k\}_k\right)$ and is further used to compute the MWIS weights. In practice, this random selection does not introduce noticeable fireflies, since paths are rejected when distributions mismatch. To avoid unnecessary computations, this test can be performed before the null interactions shift. To finalize the suffix, a direction is sampled through the uniform mixture of scattering distributions associated to the accepted prefixes, and the remaining part of the suffix is built by regular path tracing as it is view-independent.

As stated in the preceding section, a very similar approach could be used to select prefixes in case of a direct illumination technique that is view dependent (e.g. product importance sampling). This could be done by replacing the scattering pdf by the modified pdf taking into account light sources into the total variation approximation. Additionally, the discrete set of directions chosen to evaluate the approximation should take into account the peaks of the distributions.

9.1.4 Robust computation of MWIS weights

After generating path prefixes and their associated Jacobians, we compute MWIS weights for each of them. We compute the weights at the first visible interaction once and use them to accumulate a single contribution for both direct and indirect illumination. This allows us to use a single image buffer. To avoid precision issues we re-write MWIS weights in term of ratios of pdfs and Jacobians. In practice, we compute these ratios event per event as a chain is transformed, since the ratios of final pdfs may suffer from precision issues due to large pdfs in dense heterogeneous volumes. Denoting $w_{i\to j}(\bar{x}_j)$ the MWIS weight for path $\bar{x}_j$ in pixel j that results from transforming path $\bar{x}_i$ from pixel i, we obtain a more numerically stable expression:

$$
\begin{aligned}
w_{i\to j}(\bar{x}_j) &= \frac{p_{i\to j}(\bar{x}_j)}{\sum_k p_{k\to j}(\bar{x}_j)} \cdot \frac{p_{j\to j}(\bar{x}_j)}{p_{i\to j}(\bar{x}_j)} \\
&= \frac{c_i q_i(\bar{x}_i)|T'_{i\to j}|^{-1}}{\sum_k c_k q_k(\bar{x}_k)|T'_{k\to j}|^{-1}} \cdot \frac{c_j q_j(\bar{x}_j)|T'_{j\to j}|^{-1}}{c_i q_i(\bar{x}_i)|T'_{i\to j}|^{-1}} \\
&= \frac{\frac{c_j q_j(\bar{x}_j)|T'_{j\to j}|^{-1}}{c_i q_i(\bar{x}_i)|T'_{i\to j}|^{-1}}}{\sum_k \frac{c_k q_k(\bar{x}_k)|T'_{k\to j}|^{-1}}{c_i q_i(\bar{x}_i)|T'_{i\to j}|^{-1}}}
\end{aligned}
\tag{9.14}
$$

The above equation reduces to a simple form with a common denominator for all prefixes using the chain rule $|T'_{a\to b}| \cdot |T'_{b\to c}| = |T'_{a\to c}|$ and the identity $|T'_{a\to a}| = 1$:

$$
\begin{aligned}
w_{i\to j}(\bar{x}_j) &= \frac{\frac{c_j q_j(\bar{x}_j)|T'_{j\to j}|^{-1}}{c_i q_i(\bar{x}_i)|T'_{i\to j}|^{-1}}}{\sum_k \frac{c_k q_k(\bar{x}_k)|T'_{k\to k}|^{-1}}{c_i q_i(\bar{x}_i)|T'_{i\to k}|^{-1}}} \\
&= \frac{\frac{c_j q_j(\bar{x}_j)}{c_i q_i(\bar{x}_i)} \cdot |T'_{i\to j}|}{\sum_k \frac{c_k q_k(\bar{x}_k)}{c_i q_i(\bar{x}_i)} \cdot |T'_{i\to k}|} \\
&= \frac{\frac{p_{j\to j}(\bar{x}_j)}{p_{i\to j}(\bar{x}_j)}}{\sum_k \frac{p_{k\to k}(\bar{x}_k)}{p_{i\to k}(\bar{x}_k)}}
\end{aligned}
\tag{9.15}
$$

We thus efficiently compute and sum up the ratios of source and target pdf multiplied by the Jacobian during the prefix shift step for each candidate pixel. We finally compute MWIS weights when adding sample contributions to the image buffer normalizing the ratios by the sum of ratios.

9.1.5 Pixel accumulation

The complete pixel MWIS estimator averaging n samples writes:

$$
\tilde{I}_{k,n} = \frac{\sum_{i=1}^{n} w_{j\to k}(\bar{x}_{k,i}) \frac{f_k(\bar{x}_{k,i})}{p_{k\to k}(\bar{x}_{k,i})}}{\sum_{i=1}^{n} w_{j\to k}(\bar{x}_{k,i})}
\tag{9.16}
$$

where k denotes the pixel being integrated, i the sample index, and j the index of the base pixel that generated the path prefix. In practice, instead of accumulating the numerator and denominator separately we compute an incremental (online) weighted mean:

$$\tilde{I}_{k,n} = \tilde{I}_{k,n-1} + \frac{w_{j\to k}(\bar{x}_{k,n})}{\sum_{i=1}^{n} w_{j\to k}(\bar{x}_{k,i})} \left(\frac{f_k\left(\bar{x}_{k,n}\right)}{p_{k\to k}\left(\bar{x}_{k,n}\right)} - \tilde{I}_{k,n-1} \right) \tag{9.17}$$

where each channel red, green and blue of the image buffer directly stores its mean and the sum of MWIS weights is stored in the alpha channel. This allows us to preview the final render directly on the GPU by updating a single RGBA openGL texture.

9.2 Discussion

Several design choices were made in our construction.

We prefer to operate with a single image buffer per camera, hence requiring the prefix selection step to ensure that every prefix receive a significant contribution. Another approach would be to store the direct and indirect illumination estimators in two separate buffers. In this way, we could compute different MWIS weights for direct and indirect illumination and could possibly remove the prefix selection step. The former weight for direct illumination would be identical to the one we derived in this section, hence taking into account every interactions until the pivot point. The latter weight for indirect illumination could also take into account the bsdf or the phase function making the prefix selection unnecessary to reduce the variance. Still, the path prefixes with very low contributions after transforming from one base prefix should be rejected as they increase the overhead of the method for very small or no gain.

We choose to use the optical depth shift mapping for two reasons: first the distribution of shifted samples is better than the raw depth shift mapping since it takes into account the local majorant of the medium. Second, the associated Jacobian is very similar to the raw depth shift mapping, composed of ratios of values, which is less costly than the primary sample shift mapping that requires several exponentials evaluation. Additionally, the distribution of the shifted samples using the primary sample shift mapping is very close to the optical depth shift distribution, hence their quality of results are indistinguishable.

9.3 Conclusion

We detailed the building blocks of our method with the associated computations to build a multi-view pixel estimator with volumetric path reuse. In the next chapter we present our implementation and the results of this method.

Chapter 10

Practical multi-view rendering

Implementation and results

In this chapter we detail the implementation of our multiple view rendering framework described in chapter 9. We aim at making a practical implementation for multi-view rendering with the same constraints as in a production pipeline. Thus, we focus on usability, genericity and simplicity. Our work should respect the following principles:

- **Reuse paths as much as possible**. In order to save significant computational time we aim at reusing subpaths as much as possible, since path construction (i.e. intersection / visibility tests) and shading operations are costly in a path tracer. Hence, every base sample is transformed and reused if possible.
- **Handle arbitrary surfaces or volumes**. The technique should be as generic as possible and should work for any material in order to handle real productions use-cases, which is the case thanks to a combination of several shift mappings.
- **Simple to implement in a path tracer**. As a complex implementation is not likely to be useful in production, our work only decouples the first bounce on the scene and then falls back to regular path tracing.
- **Compatible with existing techniques**. Our work should be compatible with classical techniques for direct and indirect illumination since a lack of flexibility is not acceptable in production. As previously stated, our framework supports any combination of direct and indirect illumination techniques.
- **Efficiency**. Our method should be fast and use the minimum amount of required memory. We avoided precomputations in our methods, which only require cheap on-the-fly operations that readily integrate in a path tracer. Additionally our multi-view technique requires a single image buffer per camera.

10.1 Implementation details

We implemented our method in a custom renderer that includes a state of the art volumetric integrator using the spectral heterogeneous volumetric path sampling approach of Miller et al., with MIS between direct illumination techniques (i.e. light sampling+ratio tracking and bsdf sampling+delta tracking), and adaptive sampling of the image space to fill in undersampled areas. We detail specific aspects in the following sections.

10.1.1 Adaptive sampling

Sharing paths between cameras leads to inhomogeneous sample density over the film due to occlusions and our selective path reuse, as illustrated in Figure 10.1. This is a classical drawback when reusing paths as pixels lacking samples cannot be predicted. For this reason we use a multi-pass adaptive strategy to fill in undersampled areas.

Our adaptive refinement pipeline [Fraboni et al. 2019, 2022] improves on previous adaptive refinement approaches based on several observations: we combine a *total variation noise estimate* [Heitz et al. 2018, sect. 4.2] that is good at finding perceptible noise, that is fed with an *f-divergence* error criterion [Rigau et al. 2003], that is good at finding difficult and undersampled areas of the image more robustly than variance estimates.

We first proceed with several pilot iterations which distribute samples uniformly over all pixels. We then compute an error estimation based on the square root of χ^2 divergence [Rigau et al. 2003, Eq. 29] for each pixel. Let m_1 denotes the samples first moment (the sum of sample contributions), m_2 the samples second moment (the sum of sample squared contributions) and n the number of samples, the error criterion associated with pixel j writes:

$$\alpha_j = \frac{\sqrt{n \cdot m_2 - m_1^2}}{n^2} \tag{10.1}$$

Due to the need of the per pixel second statistical moments an additional buffer is required for adaptive sampling. Next, we evaluate a total variation noise estimate of the error gradient, similar to the metric proposed by Heitz et al. in Eq. 29. The noise estimate is computed using integrals of the magnitude of second order derivatives of the error criterion along m uniformly rotated lines l_i traversing the pixel j:

$$\beta_j = \frac{1}{m} \sum_{i=1}^{m} \int_{l_i} \left| \frac{\| \mathrm{d}^2 \alpha(l_i(t)) \|}{\mathrm{d}t^2} \right| \mathrm{d}t \tag{10.2}$$

In practice, we strictly follow Heitz et al. construction: we use $m = 4$ asterisk shaped lines with a random per-pixel offset angle, a random radius around pixel j, and estimate the integrals using second order finite differences. Finally, the resulting per pixel criterion β_j is used to adaptively distribute a sample budget over the image plane similarly to several production renderer [Fascione et al. 2018, Christensen et al. 2018, Burley et al. 2018]. This scheme is iterated until a time limit, a number of iterations or a target error is reached.

The adaptive scheme both redistributes samples in under sampled areas, making them imperceptible, and slightly reduces the noise in areas where error is important. In fact, the noise estimation highlights noisy areas and high gradients, which have been proved to be correlated with the variance [Manzi et al. 2016]. We empirically found that our criterion works best among several refinement criteria, such as variance estimates, f-divergences estimates alone, noise estimates alone, standard deviation estimates and relative standard deviation estimates. Nevertheless, other adaptive error criteria which focuses on undersampled areas could be used.

10.1.2 Core engine

The core source code is composed of 35000 lines of C++ all made during the course of this PhD. It has been used to render thousands of hours of images and all figures from our publications [Fraboni et al. 2019, 2022]. We showcase some renders produced by our renderer in Figure 10.2. It includes the following list of features:

- Integrators
  - Unidirectional volumetric path tracing integrator (vpt) [Miller et al. 2019]
  - Multiple view volumetric path tracing integrator (mvpt) and its adaptive version (amvpt) [Fraboni et al. 2022]
  - Multiple view path tracing integrator [Fraboni et al. 2019] (extended to volumetric support for comparisons)
  - Depth of field supersampling[1] integrator (surfaces only)
  - Discrete path reusing integrator (surfaces only) [Bekaert et al. 2002]
  - Miscellaneous: Ambient occlusion, Normal, Light tracing
- Materials
  - smooth models: diffuse reflection/transmission, conductor, dielectric with absorption, plastic with absorption, bi-layered model (coating and substrate) with absorption
  - rough models: diffuse reflection/transmission, conductor, dielectric with absorption, plastic with absorption, bi-layered model (coating and substrate) with absorption
  - microfacet models Beckmann and GGX
  - energy compensation [Turquin 2017]
  - mix BSDF
  - thin film interferences
  - BSDF validation methods [Heitz 2014]
  - BSDF similarity methods [Fraboni et al. 2022]
- Direct illumination techniques
  - BSDF sampling
  - Next event estimation [Shirley and Wang 1994]
  - Equiangular sampling [Kulla and Fajardo 2012]
  - Combinations with MIS [Veach and Guibas 1995, Miller et al. 2019]
  - RIS with CDF inversion or reservoirs (ReSTIR) [Talbot et al. 2005, Bitterli et al. 2020]
- Homogeneous and heterogeneous participating media
  - Spectral absorption, scattering and emission support

---

[1] We extended our path reuse work to handle importance sampling of the circle of confusion. This has been proposed independently from [Schwarzhaupt 2019] and is summarized in Appendix C.

  - Density functions
    - Uniform and 3D textures density
    - Sparse voxel density support with NanoVDB [Museth 2021]
    - Procedural 3D noise density (value, perlin, simplex, ridged) and 4D for animated media support with FastNoise [Peck 2020, Musgrave et al. 1994]
  - Phase functions
    - Uniform
    - Henyey-Greenstein
    - Mie approximation with numerical CDF inversion
  - Transmittance estimators
    - Spectral delta tracking and spectral ratio tracking [Miller et al. 2019]
    - Unbiased ray marching approximation [Kettunen et al. 2021]
- Textures and UV mapping
- Depth of field with thin lens model [Kolb et al. 1995]
- Time integration for motion blur
- Fast ray - scene intersection with Intel Embree [Wald et al. 2014]
- Progressive rendering with OpenGL preview
- Adaptive rendering with various error criteria [Kajiya 1986, Rigau et al. 2003, Heitz et al. 2018, Fraboni et al. 2019, 2022]

10.1.3 Toy engines

We additionally built two toy renderers based on a minimalist codebase [Georgiev et al. 2012, SmallVCM]:

- SmallMVPT a minimalist multi-view renderer [Fraboni et al. 2019, `https://github.com/bfraboni/SmallMVPT`] (surfaces only) which accompanies the article.
- SmallMVVPT a minimalist multi-view volumetric renderer (surface + homogeneous volumes) which is unpublished yet.

Note that the second one served as proof-of-concept before starting the complete heterogeneous media support in the core engine.

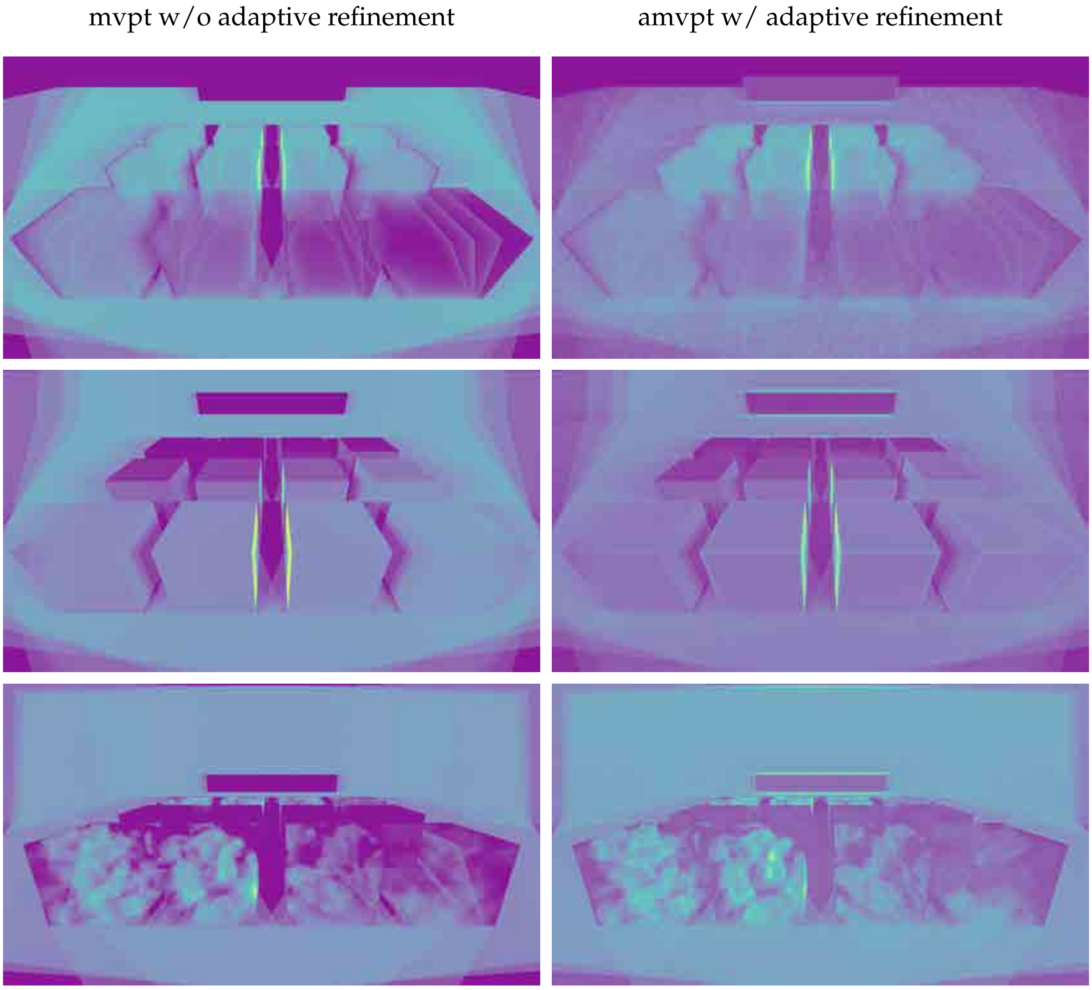


FIGURE 10.1 – We shows the samples repartition in both mvpt (left) and amvpt (right) on the anisotropy clouds scene, the varying roughness cubes scene and on the mixed cloud and specular cubes scene Figure 10.7. When the anisotropy of volumes increases less paths are reused between cameras due to our similarity test between scattering distributions, leading to undersampled areas (top row left – dark cubes in the front row). When the roughness of surfaces decreases the less paths are reused between cameras due to our similarity test between scattering distributions, leading to undersampled areas (middle row left – dark cubes in the back row). Specular surfaces can not benefit from path reuse without using special shifts for specular chains [Jakob and Marschner 2012, Kettunen et al. 2015], leading to undersampled areas (bottom row left – dark cubes in the middle row). Note that our adaptive refinement strategy (right column) redistributes samples in these dark areas making them imperceptible.

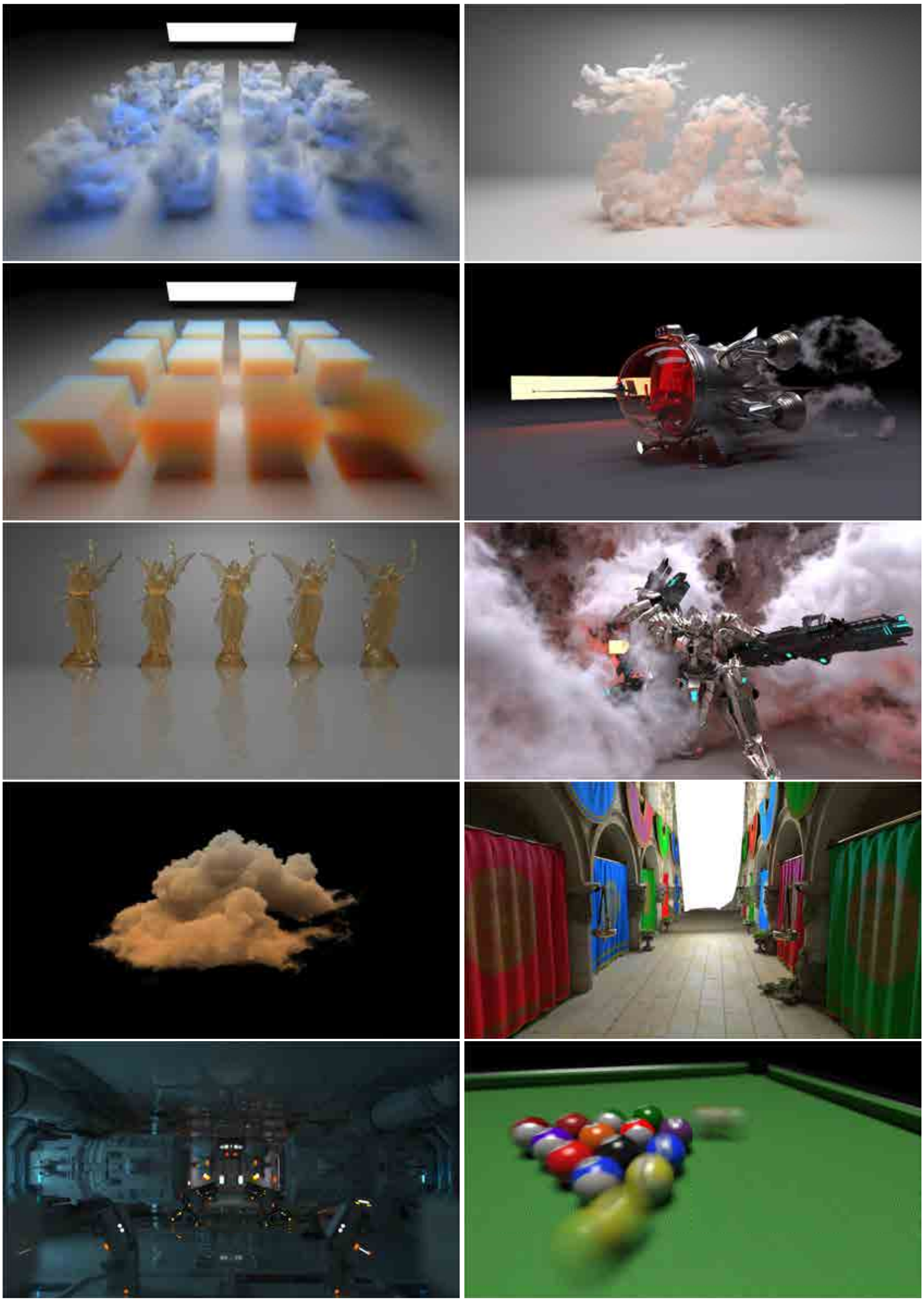


FIGURE 10.2 – Showcase of the rendering engine. The dragon model courtesy of Stanford Computer Graphics Laboratory. The spaceship model courtesy of thecali. Lucy model courtesy of Stanford Computer Graphics Laboratory. The shogun model courtesy of Zeroswat. The Disney Cloud Data Set courtesy of Walt Disney Animation Studios. The sponza atrium model courtesy of Marko Dabrovic. Zero day model courtesy of Beeple. The pool table model courtesy of Chamouleau.

10.2 Results

We illustrate our rendering technique on several applications, including the rendering of lenticular images, lightfield images, holographic stereograms, virtual walkthrough in static scenes. We also designed several test scenes with varying materials and camera setups to evaluate our algorithm under different conditions and to evaluate our adaptive refinement. Finally we rendered several videos of dynamic scenes including motion blur. In the following examples, except for the Disney cloud, all volumes are procedural and made from ridged multifractal or simpler functions [Musgrave et al. 1994].

10.2.1 Applications

Lenticular images. We print lenticular images that consist of 10 views with small baseline and horizontal parallax (Figure 10.3, top). A lenticular sheet of 60 lenticles per inch was used, images were printed in 600 dpi, leaving 10 views per lenticle.

Lightfield images. Our lightfield display – a 8K Looking Glass display – renders horizontal parallax light fields consisting of 45 views (Figure 10.3, bottom).

FIGURE 10.3 – Top row. We print lenticular images using 10 views. Bottom row. Our lightfield display renders images consisting of 45 views. We demonstrate our method on 3 lightfield images.

Holographic stereograms. Holographic stereograms are holograms that are printed using many ordinary renderings of the same scene (as opposed to rendering an interference pattern). Cameras are typically positioned extremely close to the object, and baseline is very small. Each rendered image is displayed on a spatial light modulator, traversed by a laser light beam, and a system of lenses prints its optical Fourier transform on a tiny part of a sensitive film, called hogel. The grid of hogels is composed of 114×171 views. Rendering $114 \times 171 \approx 20k$ views at once would be intractable due to memory limits. In practice, we rendered tiles of 7×7 views with our method. A generated monochrome hologram can be seen in Figure 10.4 using a prototype printer.

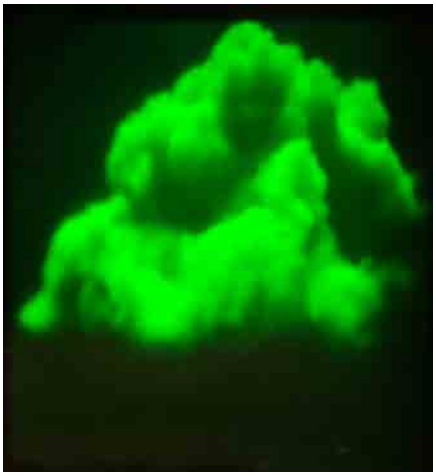


FIGURE 10.4 – Hologram printed from 114 × 171 views of the Disney cloud rendered via tiles of 7 × 7 views.

Virtual walkthroughs. Camera paths in static scenes can also benefit from our method. Our supplementary video showcases a walktrough over the dragon scene with 100 frames, each considered as one view.

Video motion blur. We demonstrate the benefit of our method on a sequence of an animated cloud. In that case, the temporal filtering (i.e., motion blur) reuses paths from views that are temporally adjacent. We use a filter size of 5 frames.

Stereo pairs. Note that our methods also works in stereo rendering setups, however the gain is of less visual interest than with other multi-view setups with more cameras. In fact, with only two times more samples the error is at most divided by half. In our tests, the gain with stereo rendering depends on the scenes and camera overlapping regions but is at most 1.7 times more samples a equal time on a complete diffuse scene.

10.2.2 Comparisons

Full render statistics and equal-time comparisons for all our results are provided in Table Figure 10.9, while rendered images with comparison to single-view path-tracing (vpt) can be seen in Figure 10.5, and comparisons to Fraboni et al. [Fraboni et al. 2019] and Mendez et al. [Méndez Feliu et al. 2006] are shown in Figure 10.8. We demonstrate the ability of our adaptive refinement strategy (amvpt) to fill in the undersampled regions that occurs with the (mvpt) integrator in Figure 10.7. In Figure 10.6 we demonstrate how the relative mean square error (relMSE) and symmetric mean absolute percentage error (SMAPE) decrease as we increase the number of views.

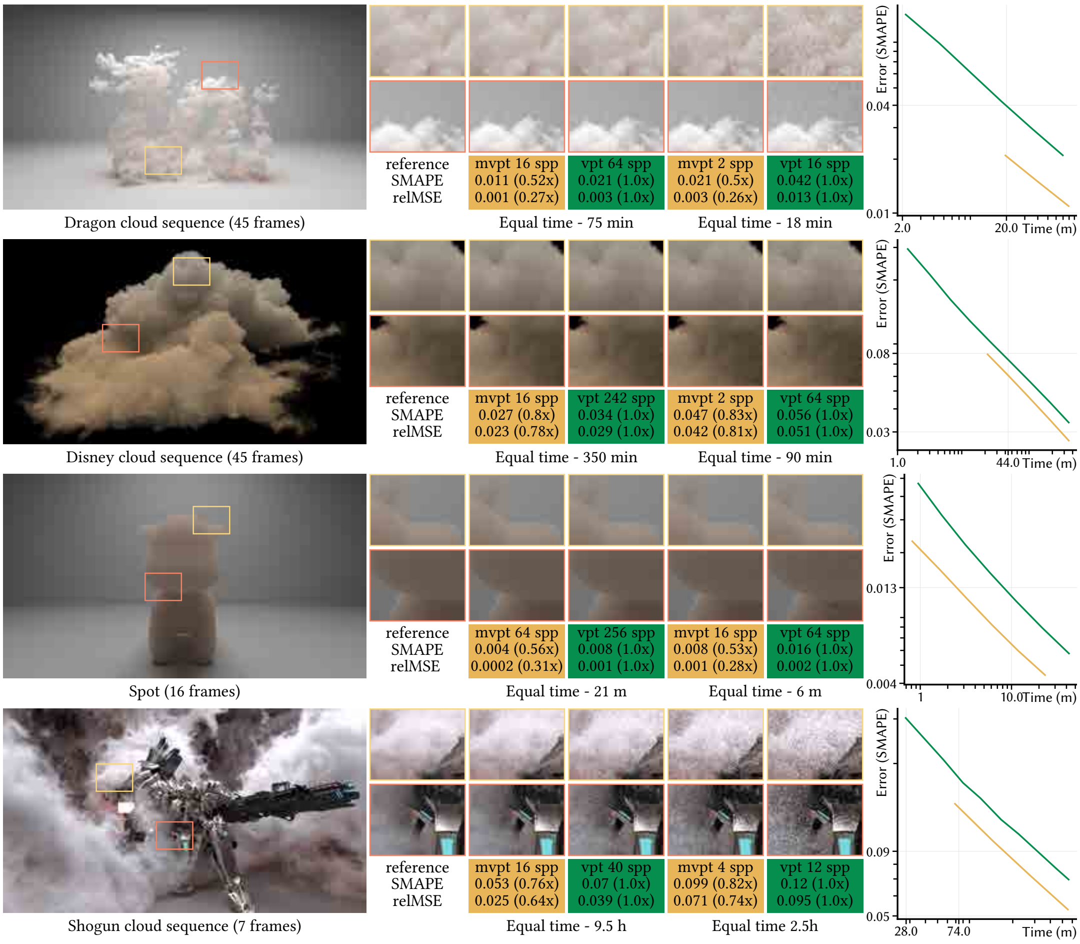


FIGURE 10.5 – Side by side comparison of our technique (mvpt) against independent volumetric path tracing (vpt) on different sequences.

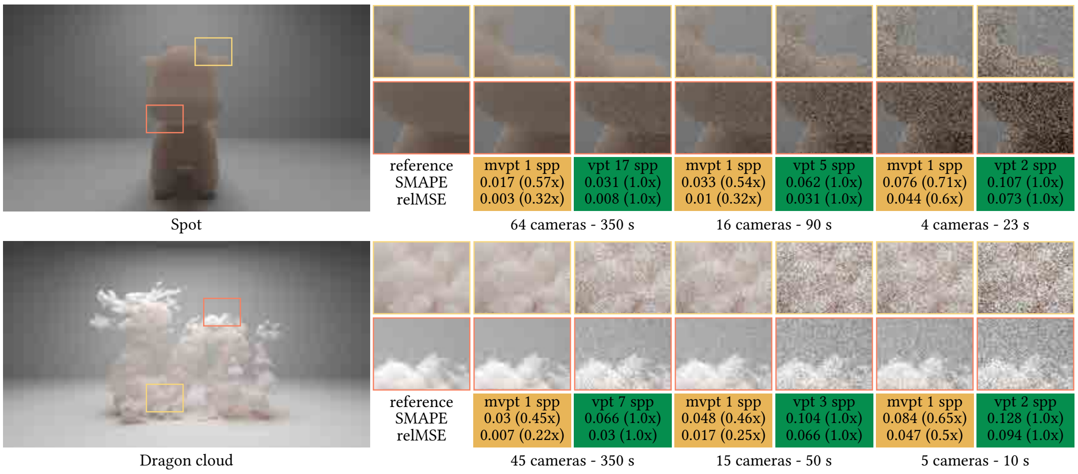


FIGURE 10.6 – Comparisons with variable number of cameras. Increasing the number of views results in lower error at equal rendering time.

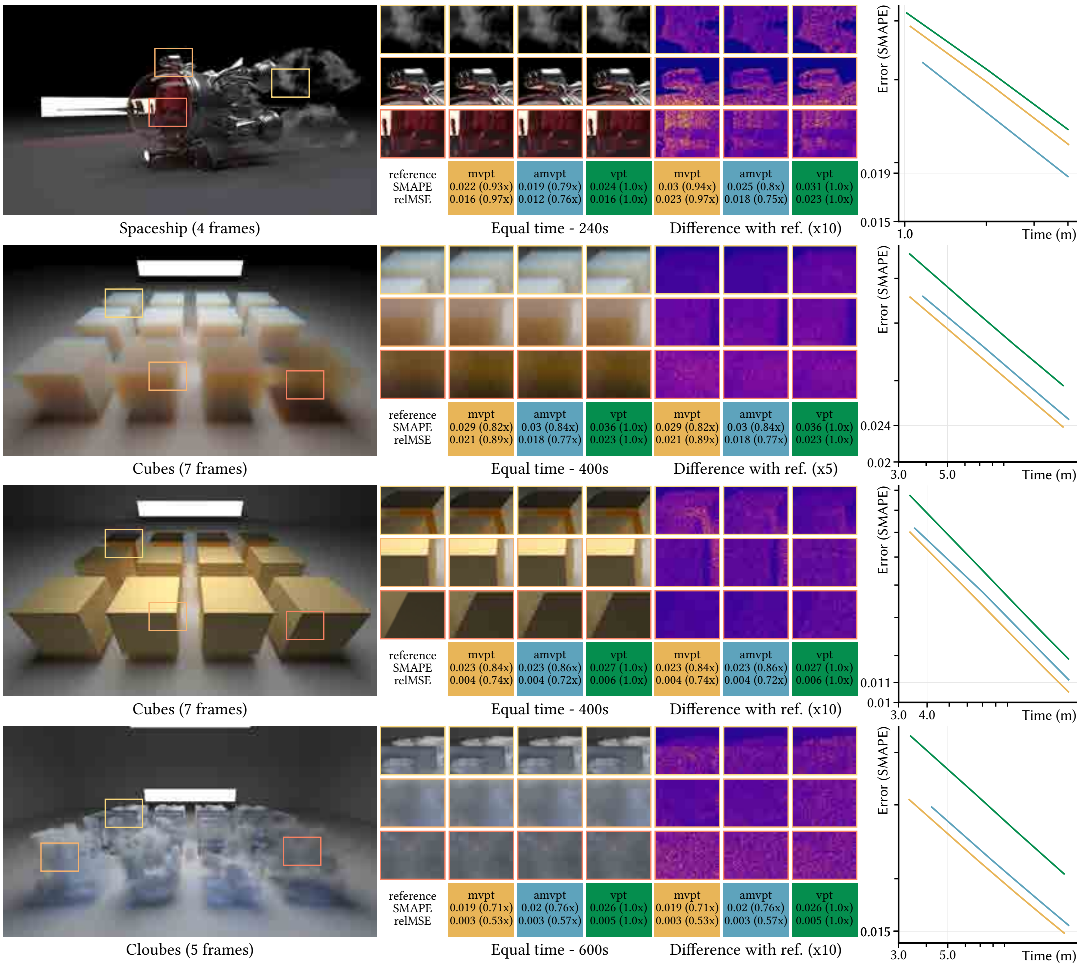


FIGURE 10.7 – We illustrate our adaptive refinement (amvpt). The top row shows a scene with isotropic phase function and difficult-to-render purely refractive dielectrics and glossy metals. The second row shows clouds of increasing anisotropic phase function, from left to right and top to bottom the mean cosine parameter g of the phase function varies from 0 to 0.98, we expect less reuse in strongly anisotropic media. The third row shows cubes of increasing linear roughness [Burley and Studios 2012], from left to right and top to bottom the parameter of the BSDF varies from 0.001 to 0.98, we expect less reuse on low roughness surfaces (mirror-like). The botton row shows perfectly specular dielectric cubes in a scene, for which we expect no reuse. In all cases, the refinement helps distribute samples in undersampled and difficult areas. The adaptive method has a slight overhead due to error and sample maps evaluation, and distributes less samples in regions where the multi-view already performs well.

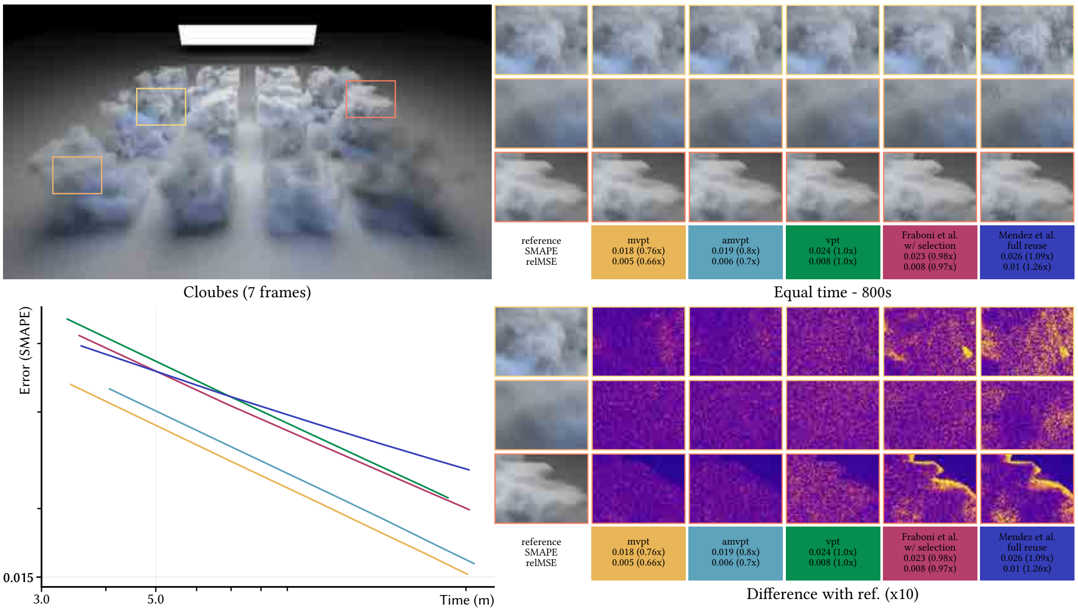


FIGURE 10.8 – We compare our methods (mvpt) and its adaptive version (amvpt) that use our MWIS estimator, to state of the art multiple view rendering estimators [Fraboni et al. 2019, Méndez Feliu et al. 2006] that use MIS and which assume per pixels normalization terms constant. Note that we used our new shifts for transforming prefixes in participating medias, and our new similarity distance between scattering distribution as path rejection criterion, to replace the ad-hoc heuristic from Fraboni et al. method. Reusing all prefixes disregarding the variation of the Jacobians and the scattering function similarity, akin to Mendez et al. method (dark blue curve and insets), results in strongly visible fireflies and bias that precludes convergence. Although the rejection proposed by Fraboni et al. (red curve and insets) limits the degradations due to large variations of Jacobians, it still exhibits some fireflies and a visible bias that do not vanishes as the number of samples increases. On the contrary our multi-view (mvpt) and adaptive (amvpt) solutions exhibit lower levels of noise and no visible bias (nor in the curve slopes) or artifacts even at low sample count and performs better than independent path tracing (vpt).

Scene	Multi-View					Single-View			
	MSE	Time	Native SPP	Total SPP	# views	MSE	Time	SPP	Resolution
Shogun ^{1} (7 frames)	**0.025**	9.5h	16	94	7	0.039	9.5h	40	1280x720
Shogun ^{1} (lenticular)	–	15.5h	16	122	10	–	–	–	1280x720
Shogun ^{1} (video)	–	55h	8	–	50	–	–	–	1280x720
Spot ^{2} (4 cameras)	**0.044**	23s	1	3	4	0.073	23s	2	1280x720
Spot ^{2} (16 cameras)	**0.01**	90s	1	15	16	0.031	90s	5	1280x720
Spot ^{2} (light field)	**0.003**	350s	1	52	64	0.008	350s	17	1280x720
Spot ^{2} (lenticular)	**0.001**	7m	16	–	10	0.002	7m	64	1920x1080
Dragon ^{2} (5 cameras)	**0.047**	10s	1	5	5	0.094	10s	2	1280x720
Dragon ^{2} (15 cameras)	**0.017**	50s	1	13	15	0.066	50s	3	1280x720
Dragon ^{1} (light field)	**0.001**	75m	16	580	45	0.003	75m	64	1280x720
Cloud ^{2} (lenticular)	**0.011**	45m	32	–	10	–	–	–	1920x1080
Cloud ^{2} (light field)	**0.023**	350m	16	290	45	0.029	350m	242	1280x720
Cloud ^{3} (hologram)	**0.009**	201d	4	80	19494	0.011	201d	64	1920x1080
Motion blur cloud ^{1} (video)	–	21h	16	48	20	–	21h	20	1280x720
Dragon flyover ^{2} (video)	**0.0011**	96m	4	–	100	0.0014	96m	62	1024x640

FIGURE 10.9 – Equal time comparison for various scene setups. Compute setups: ^{1} Desktop computer 1, ^{2} desktop computer 2, ^{3} compute cluster with 424 jobs of 7x7 camera tiles.

10.3 Discussion and limitations

We discuss several specific aspects of the method that could further enrich end improve the results and the current limitations of our implementation.

10.3.1 Reuse through specular interfaces: specular manifold techniques

Our method is useful only on non truly specular surfaces, since no reuse at the first real interaction can occur in such situations. This results in undersampled regions of the image and we rely on adaptive sampling to alleviate the problem.

However, solutions have been explored to transform constrained specular paths, such as specular manifold exploration [Jakob and Marschner 2012, Zeltner et al. 2020] or techniques for direct lighting through specular chains [Hanika et al. 2015, Weber et al. 2017, Loubet et al. 2020] and specular shifts (i.e. half vector copy) for gradient domain rendering [Gruson et al. 2018, Bauszat et al. 2017, Manzi et al. 2016, 2015, Kettunen et al. 2015, Lehtinen et al. 2013]. Our shift mappings for null scattering chains could be combined with such techniques, to handle specular manifold exploration, independently from our contributions as illustrated in Figure 10.10. Such combination could further reduce the sample inhomogeneity in multi-view images in presence of specular or near specular surfaces.

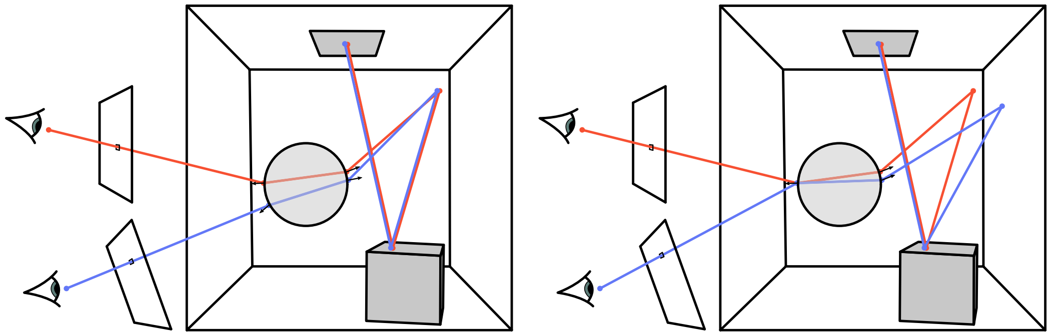


FIGURE 10.10 – Multiple view rendering with reuse at specular interfaces. On the left specular manifold exploration is used after disconnecting the lens point from the base camera (red) to the target lens to maintain the specular constraint of the prefix chain (blue). On the right half vector copy is used after connecting the pivot point to the target lens to continue through specular surfaces, and reconnecting once two consecutive diffuse vertices has been found.

10.3.2 A-priori versus a-posteriori methods: denoising

A posteriori methods have also been proposed in the context of spatio-temporal sequences [Vogels et al. 2018, Zimmer et al. 2015] to reduce variance (or noise). However, they require exporting a large number of decomposed layers, which first requires implementing exporters for example to compute specular motion vectors [Zimmer et al. 2015], and second would further increase the memory footprint of our multi-view method that already stores every image buffer. Additionally they are not designed in essence for arbitrary camera locations, but rather for consecutive frames of animated sequences, hence can not be employed in all cases our algorithm covers. Still our method could benefit from a spatio-temporal denoising post processing pass.

Other spatial denoising methods [Áfra 2018, Zwicker et al. 2015, Parker et al. 2010, Boughida and Boubekeur 2017] can be applied in a final pass to remove the remaining noise, similarly to path tracing renders, further benefiting of the reduced variance due to our path reuse. In fact, the results of our methods attain a desired noise level faster than classic single frames approaches.

10.3.3 Bias impact

Our MWIS estimators is biased and consistent. In practice we did not notice visible bias at fixed sample budget in comparison to single frame path tracing on scenes of different complexity. This is due to two reasons: first since we use our estimator for path reuse purposes, the number of samples per pixel quickly increases, hence rapidly reducing the supposedly visible bias as the bias vanishes in $O\left(n^{-1}\right)$. Second, the bias is supposed to be visible at low sample count (less than 50). However, the complex light transport integrand results in noisy estimates that in practice hide the bias. In fact, the mean squared error of an estimator is the sum of the variance and the squared bias, if the bias is negligible w.r.t the variance then it has very low impact on the final error.

10.3.4 Performances

Our research prototype is designed as proof of concept and is not representative of the true performance that an optimized version could reach. Hence, there is still room for improvements.

Cache coherency

Due to the high number of image buffers, using adequate structures (i.e. sample queues) to maintain cache coherency while writing contributions could further improve the performances of our method. Currently this is the main performance bottleneck we encounter that precludes our implementation to get closer to the theoretical efficiency of the method (n cameras → n times more samples at equal time). Still, the gain of our method is significant with our current implementation.

Shift mapping complexity

Our shift mappings of null scattering chains requires storing the prefix chains for every visible and selected pixel. In dense heterogeneous mediums the chains of events may be very long. Storing and shifting such long chains can be costly in practice and of limited gain if the suffix path built after is short. For example, integrating direct illumination with our multi view method works fine but may converge slower than the classic frame by frame approach due to the overhead of the shifts. For that reason, it could be favorable to detect the areas where mvpt is less efficient to fall back to single frame vpt.

10.4 Conclusion

In this section we presented the results of our multi view rendering framework combining our contributions: new volumetric shift mappings, a new MWIS estimator, a new similarity metric for arbitrary distributions and a new adaptive refinement criterion.

We showed that our method allows to render complex scattering in participating media in an efficient way by shifting and sharing the construction of difficult paths across views. Our multi view method improves the results by an order of magnitude in comparison to classical frame by frame volumetric path tracing when the scene benefits from reuse. This signify that we reach a desired quality faster, or that our results exhibit less noise at equal time.

Chapter 11

Conclusion

We conclude this dissertation with a summary of our contributions and the doors they open.

11.1 Self-normalized multi strategy estimators

As a first contribution, we introduced a new family of Monte Carlo estimators, *Multiple Weighted Importance Sampling*, that builds upon two existing family of estimators, the well known *Multiple Importance Sampling* and the less known *Weighted Importance Sampling*. We combined the best of both methods to overcome some limitations of unbiased MIS estimators in some multiple strategies scenarios. We showed that our MWIS estimator can effectively reduce the variance of the combination of arbitrary techniques (mixed poor and good) when MIS cannot.

In practice, we use a target technique that is known to be a good sampling technique, to reweight the contributions of the different strategies as if they were directly sampled using the target strategy. This is even more efficient when one of the source strategies is also the chosen target. Furthermore we show that our estimator simply reduces to a weighted mean of contributions, which is as efficient to evaluate as a classic mean for unbiased Monte Carlo estimators. We exploit the benefits of our estimator in a practical rendering application that suffers from variance or require complex extra work with MIS estimators.

Our MWIS estimator is biased and consistent, but in our experiments the bias is imperceptible even at low sample counts, in which case the error is dominated by the variance or the bias has vanished thanks to the high number of reused contributions. Even though our estimator is biased, we believe that the combination of MIS and WIS is a powerful tool that could benefit to other applications. In fact, ratio estimators have already proved their efficiency in several rendering applications. Finding new applications that could benefit from MWIS is left for future investigation. Still, an unbiased version of such estimator, that do not require stochastic resampling or additional random walk is also an interesting area of future work.

11.2 Path reusing in presence of heterogeneous media

In the second part of our contributions, we unify surfaces and volumetric path reuse through new shift mappings in presence of arbitrary media.

Our shifts are designed to ensure that a valid deterministic bijective mapping exists between a source and a target chain of medium interactions. Furthermore, they are simple to evaluate with their Jacobian in closed form. However, they do not perfectly reproduce the desired distributions of events on the target chain. Hence, finding efficient mappings of better quality is an interesting direction for future work.

Our mappings straightforwardly combines with existing mappings to allow richer path transformations. This is a step towards extending classical path reusing methods to volumetric rendering with null scattering methods. Notably, extending gradient-domain rendering to null scattering is currently in course of research as a direct application of our shift mappings, since this method cannot currently handle arbitrary heterogeneous media. In fact, constructing correlated base and offset paths to sample image gradients in presence of heterogeneous media requires shifting the null collisions encountered on the base path towards the offset path, which can be done using our mappings. Similarly, other path reusing methods, such as discrete path reusing and depth of field supersampling, could be extended to null scattering. Further investigations could be done to use our shift mappings within bidirectional methods which has been left aside during the thesis.

11.3 Practical multi-view rendering

In the third part of our contributions, we propose a practical volumetric multi view rendering framework that builds upon our MWIS estimator and our shift mappings. We further propose a similarity metric between scattering distributions to select the prefixes that can share significant suffix contributions.

We implemented our method in a custom rendering engine with state of the art methods for volumetric rendering and direct illumination. This is the first path reusing application that handles heterogeneous media. Additionally we develop a new adaptive sampling error criterion that correctly focuses on under-sampled areas and noise, that classical adaptive error criteria do not achieve. We compare our multi-view and adaptive multi-view methods to frame by frame rendering, and improves the results by an order of magnitude on several test scenes. An open source implementation of the method in a common open source research oriented renderer [Jakob 2010, Nimier-David et al. 2019, Pharr 2018] could be an interesting future work.

List of Figures

Appendix A

The camera importance function

I have read several articles and path tracer implementations that uses different angles or distances in the primary rays pdf and camera importance computation. This did not help me to understand what was the terms involved, even while reading the well-known book *PBRT* [Pharr et al. 2016]. I found out that other computer graphics developers shared the same doubts in The Missing Primary Ray PDF in Path Tracing. So after a bit of reading regarding camera optics and ray generation, I rewrote the complete derivation for camera ray pdfs and importance function.

A.1 The thin lens approximation

First described in 1984 in [Cook et al. 1984], the thin lens model is a good approximation for depth of field effect in rendering. But the clean derivation of the light transport integral is not given in the paper. Later published, the derivation in [Kolb et al. 1995] gives the equation of the radiance measured at a point x_0 on the film plane and let appear the "cosine fourth" falloff law due to the lens system (cf. Figure A.1). Another form of the derivation (and its extension to thick lenses) is described in [Hanika and Dachsbacher 2014], in which the falloff term is described as the change of measure between the film area and the exit direction. Indeed, the lens system deviates the observed direction, that is why the final density on the sensor is not homogeneous.

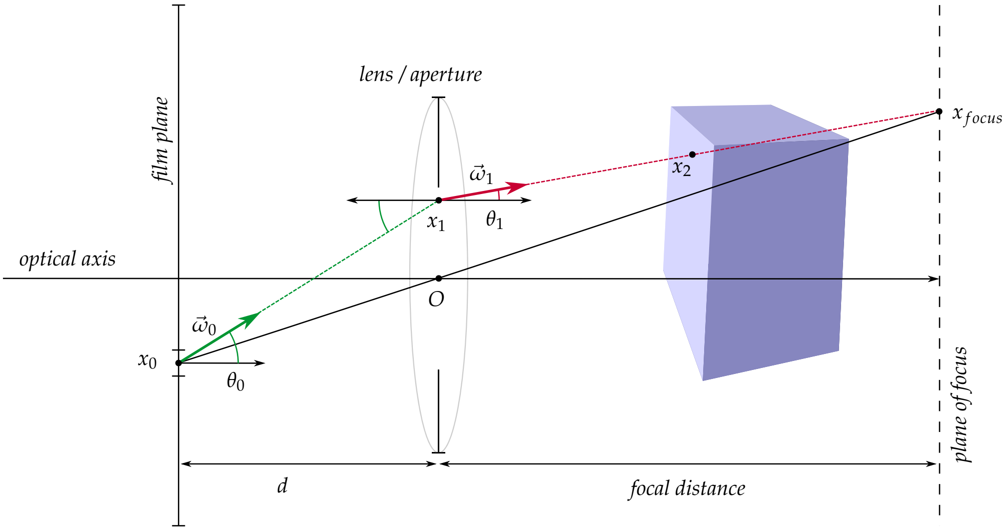


FIGURE A.1 – Lens - Image Plane configuration

We notice that these two last references are using the same terms (cf. Figure A.1):

- d – the distance between the film point x_0 and the lens point x_1,
- θ_0 – the angle between the direction $\vec{\omega}_0$ connecting the film point to the lens point and the camera forward direction,
- θ_1 – the angle between the exit ray direction $\vec{\omega}_1$ in the scene and the camera forward direction.

A.2 Interaction with the lens

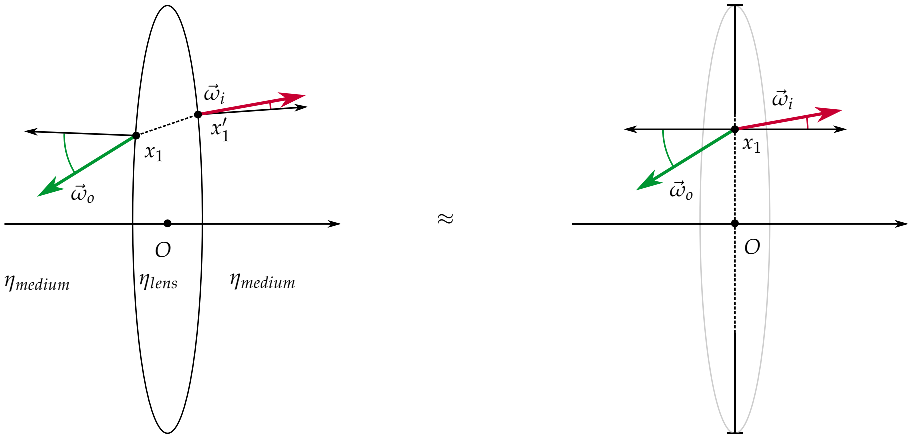


FIGURE A.2 – The thin lens approximation

We briefly recall the assumptions defining the thin lens model:

- The thickness of the lens is neglected. The entry point and the exit point are assumed to be the same, here x_1 (cf. Figure A.2). The interaction inside the lens is not taken into account, thus there is no geometric term G involved while passing through the lens.
- The medium at the entry and at the exit of the lens is the same. A light ray passing through the lens undergoes two consecutive refractions. Thus the interactions involved are *medium* → *lens* at the entry then *lens* → *medium* at the exit. Deriving Snell's law of refraction (see [Veach 1998, chap. 5.2] and [Pharr et al. 2016, sect. 8.2.3]) two times for two inverse interactions cancel out the terms accounting for the change of refractive media.
- The refraction is perfect through the lens and the radiance carried by a ray is integrally transmitted. This implies that the ray throughput and the directional probability density are unchanged while passing through the lens.
- Given an outgoing direction $\vec{\omega}_o$, the corresponding incident direction $\vec{\omega}_i$ deviated by the lens system is unique (and conversely).

A.3 Derivation of the camera importance function

The camera importance function W_j introduced by Veach [Veach 1998] hides the sensor response of the camera and a filter function over exposure time and image space. The Monte Carlo simulation of light transport involves sampling schemes and their associated probability densities (pdf) to compute the images. That is why in practice the camera importance function is baked in a smart way to cancel out some terms.

We recall that integrating the camera importance function over the exit pupil point x_1 of the camera – the first real surface point in the scene usually confound on the lens surface for thin lens model (but this is untrue for other complex lens models) – and the exit ray direction writes:

$$\int_A \int_\Omega W_j(x_1, \omega_1) \cos\theta_1 \mathrm{d}x_1 \mathrm{d}\omega_1 \tag{A.1}$$

This equation must integrate to one to respect the normalization criterion. This requires canceling the cosine term between the camera forward direction and the ray direction leaving the pupil and the probability densities that may rise from sampling the lens point and the ray direction (defined w.r.t area times solid angle measure), hence:

$$W_j(x_1, \omega_1) = \frac{p(x_1)p(\omega_1)}{\cos\theta_1} \tag{A.2}$$

Note that when integrating over the time dimension for motion blur in animated scenes, we similarly take into account the time sampling pdf in the camera importance function. Assuming a simple thin lens camera model, the exit point is uniformly sampled over the lens surface, for which the pdf writes:

$$p(x_1) = \frac{1}{\pi r^2} = \frac{1}{A_{\text{lens}}} \tag{A.3}$$

where r is the lens radius. Until here [Pharr et al. 2016] describe the same equations. However, the camera ray direction pdf is constrained by the sampled film point pdf $p(x_0)$ and the inverse change of measure between the film and the lens given by the following the relation:

$$\mathrm{d}A = \mathrm{d}\omega \frac{\cos\theta}{d^2} \Leftrightarrow \mathrm{d}\omega = \mathrm{d}A \frac{d^2}{\cos\theta} \tag{A.4}$$

Finally the exiting ray pdf writes:

$$\begin{aligned} p(\omega_1) &= p(x_0) \frac{\|x_0 - x_1\|^2}{\cos\theta_0} \\ &= \frac{1}{A_{\text{film}}} \frac{\|x_0 - x_1\|^2}{\cos\theta_0} \\ &= \frac{d^2}{A_{\text{film}} \cos^3\theta_0} \qquad \text{given} \qquad \|x_0 - x_1\| = \frac{d}{\cos\theta_0} \end{aligned} \tag{A.5}$$

Putting pieces together we get the final form of the camera importance function as follows:

$$W_j(x_1, \omega_1) = \begin{cases} \frac{d^2}{A_{\text{lens}} A_{\text{film}} \cos^3\theta_0 \cos\theta_1} & \text{if } r > 0, \\ \frac{d^2}{A_{\text{film}} \cos^4\theta_1} & \text{if } r = 0. \end{cases} \tag{A.6}$$

A.4 PBRT Erratum

In [Pharr et al. 2016, sect. 16.1.1], the authors provides some details about this formulation which suffer from a lack of explanations and clear figures. A complete figure of the system (see Figure A.1) is easier to understand that the split ones exposed in the book. I noticed that authors wrongly assume $\theta_0 = \theta_1$ in their formulation (and in the source code also), instead of distinguishing them. Note that the Equation A.6 reduces to the one given in [Pharr et al. 2016, section 16.1.1 equation 16.4] if and only if:

- the film is at distance $d = 1$ from the lens, which is possible since it is a user parameter of the camera model,
- the lens radius is $r = 0$, hence $\theta_0 = \theta_1$, which does not hold if a thin lens is used and rise a division by zero in their equation.

A.5 Code

We reproduce the corrected code that returns the camera importance value for a ray given as parameter.

```
// compute importance for a given ray and return the associated pixel
float we(const Ray& ray, vec2& pixel) const
{
    // check if ray belongs to the time window
    if( ray.time() > close || ray.time() < open  ) return 0.f;

    // check if ray points out of the camera
    Point p_lens = w2c(ray.origin());
    Vector w_lens = w2c(ray.direction());
    if (w_lens.z >= 0) return 0;

    // find point of focus
    float t_focus = std::abs(z_focus / w_lens.z);
    Point p_focus = p_lens + t_focus * w_lens;

    // find pixel position
    Point p_film = c2r(p_focus);
    pixel = vec2(p_film.x, p_film.y);

    // return zero importance for out of bounds points
    if( !valid(pixel) ) return 0;

    // direction between camera origin, film point and point of focus
    Vector w_focus = normalize(Vector(p_focus));

    // find true film point in camera space
    float t_film = std::abs(z_film / w_focus.z);
    Point p_film(t_film * w_focus);

    // lens to film direction in camera space
    Vector w_film = normalize(p_lens - p_film);

    // compute probability densities
    float pdf_p_lens = lens.eval();
    float pdf_w_lens = z_film * z_film / (film_area * w_film.z * w_film.z * w_film.z);
    float pdf_time = close-open > 0 ? 1 / std::abs(close-open) : 1;

    // return camera importance value
    return std::abs(pdf_time * pdf_w_lens * pdf_p_lens / w_lens.z);
}
```

Appendix B

Majorant optical depth shift code

We reproduce the code extract that performs our majorant optical depth transformation for completeness. Note this snippet focuses on readability mor the performance and could further be optimized.

```
struct MediumRecords
{
    std::vector<float> distances; // depth entry / exit of each medium section (2*n)
    std::vector<Medium*> mediums; // medium associated to each section (n)
    Ray ray;
    int size() const { return mediums.size(); }

    // i-th section majorant optical depth
    float mod(int i, const int channel) const
    {
        return (distances[2*i+1] - distances[2*i]) * mediums[i]->barMu(channel);
    }

    // total sections majorant optical depth
    float mod(const int channel) const
    {
        float m = 0.f;
        for(int i = 0; i < size(); ++i)
            m += mod(i, channel);
        return m;
    }

    // return normalized majorant optical depth in [0,1] from real depth
    float depthToModRatio(const float t, const int channel) const
    {
        float modt = 0.f;
        for(int i = 0; i < size(); ++i)
        {
            const float a = distances[2*i];
            const float b = distances[2*i+1];
            // cumulate section length until we reach t
            modt += (std::min(t,b) - a) * mediums[i]->barMu(channel);
            if( t <= b )
                break;
        }
        float modmax = mod(channel);
        float r = std::min(1.f, std::max(0.f, modt / modmax));
        return r;
    }

    // return real depth from normalized ratio of majorant optical depth
    float modRatioToDepth(const float ratio, const int channel) const
    {
        if( ratio == 1.f ) return distances.back();
        if( ratio == 0.f ) return distances.front();

        const float modt = ratio * mod(channel);
        float mod = 0.f;
        for(int i = 0; i < size(); ++i)
        {
            const float a = distances[2*i];
            const float b = distances[2*i+1];
            mod += (b - a) * mediums[i]->barMu(channel);
            // the real position is in this section
            if( modt <= mod )
            {
                // from the end of the section we have to move backwards of
                // delta = (mod - modt) / barMu to find the real depth
                return b - (mod - modt) / mediums[i]->barMu(channel);
            }
        }
        return distances.front();
    }
};
```

Appendix C

Lens supersampling

We describe in this appendix an approach to reuse sample paths by multiple lens connections using a shift mapping operator and importance sampling of the circle of confusion.

Thin lenses

The thin lens model is used to simulate depth of field by deviating camera rays [Kolb et al. 1995, Hanika and Dachsbacher 2014] (see Appendix A for further details). The parameters required by the model are:

- d_{film} the distance between the lens and the film plane,
- d_{focus} the distance between the lens and plane in focus,
- R the lens radius.

Circle of confusion

The depth of field effect is due to the optical properties of lens systems. The projected image of an object is a blurred circle over the film plane. This region is known as the circle of confusion. The blur circle of diameter C in the focus plane times the magnification ratio m gives us the diameter of the circle of confusion c (cf. Figure C.1).

$$c = Cm \tag{C.1}$$

The magnification is the ratio between the film distance and the focus distance.

$$m = \frac{d_{film}}{d_{focus}} \tag{C.2}$$

The blur circle only depends on distances d_P and d_{focus}, and the lens radius R. Indeed, the intercept theorem (Thales) gives the relation:

$$\frac{C}{2R} = \frac{|d_P - d_{focus}|}{d_P} \tag{C.3}$$

Putting every pieces together let us express the diameter of the circle of confusion w.r.t known distances as:

$$c = 2R\frac{|d_P - d_{focus}|}{d_P}\frac{d_{film}}{d_{focus}} \tag{C.4}$$

Hence, for every point P visible from the camera, we can easily compute the associated circle of confusion area over the image plane:

$$A_{circle} = \pi \left(R \frac{|d_P - d_{focus}|}{d_P} \frac{d_{film}}{d_{focus}} \right)^2 \tag{C.5}$$

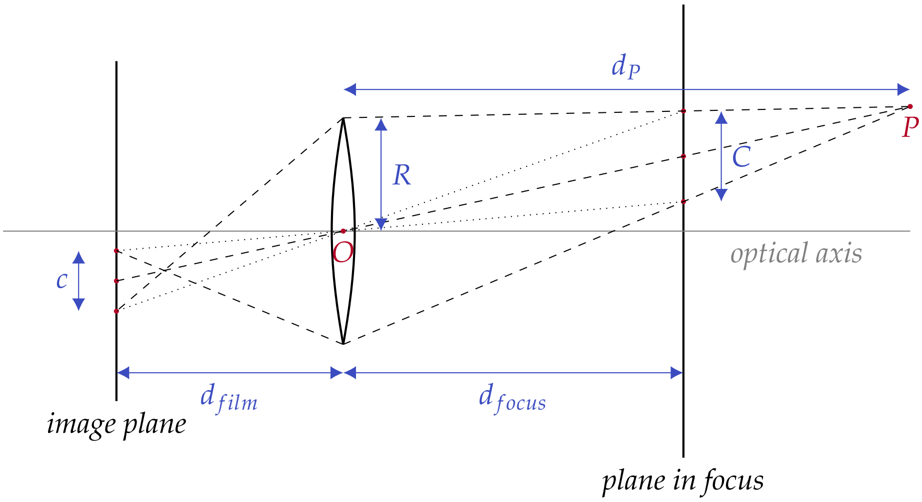


FIGURE C.1 – Geometric configuration

C.1 Importance sampling the circle of confusion

Given a scene point P visible from the camera lens, we want to distribute samples proportional to the size of the associated circle of confusion. A straightforward manner is to compare the pixel area and the circle area over the image plane. The number of samples required is given by the ratio between the pixel area and the circle of confusion area. Then a circle of confusion covering the area of ten pixel would result in ten contributions.

$$n = \frac{A_{circle}}{A_{pixel}} \tag{C.6}$$

Since the ratio can be arbitrary small or large, we clamp the value to a minimum of one, because we do not want less than one sample, and a maximum of a fixed user parameter n_{max} setting the maximum splatting limit for a given path.

C.2 Sampling the lens

Sampling a path requires sampling a point on the lens. The initial lens point is sampled in the unit square and then remapped on the disk. The concentric mapping by Shirley and Chiu [Shirley and Chiu 1997] maps a point (ξ_1, ξ_2) in the unit square to the unit disk while preserving the area. First the point is remapped in the [-1,1] square.

$$a = 2\xi_1 - 1 \qquad b = 2\xi_2 - 1 \tag{C.7}$$

Then the point (a, b) is mapped to the unit disk using the concentric mapping.

$$\begin{cases} u = a\cos\left(\frac{\pi b}{4a}\right) & v = a\sin\left(\frac{\pi b}{4a}\right) & \text{if } a^2 > b^2 \\ u = b\cos\left(\frac{\pi}{2} - \frac{\pi a}{4b}\right) & v = b\sin\left(\frac{\pi}{2} - \frac{\pi a}{4b}\right) & \text{otherwise} \end{cases} \tag{C.8}$$

C.3 Shifting the lens sample

We aim to reuse the path sample on other lens points such that the depth of field benefits from super sampling. We want to generate a set of n lens points following a known shift of the initial lens point (e.g. regular grid, QMC sequences, optimized pointsets). The set size n is proportional to the area of the circle of confusion. The supersampling is done by iterating over the shifted lens points and constructing the transformed path using. An illustrative example of shifting a sequence is given in Figure C.2 .

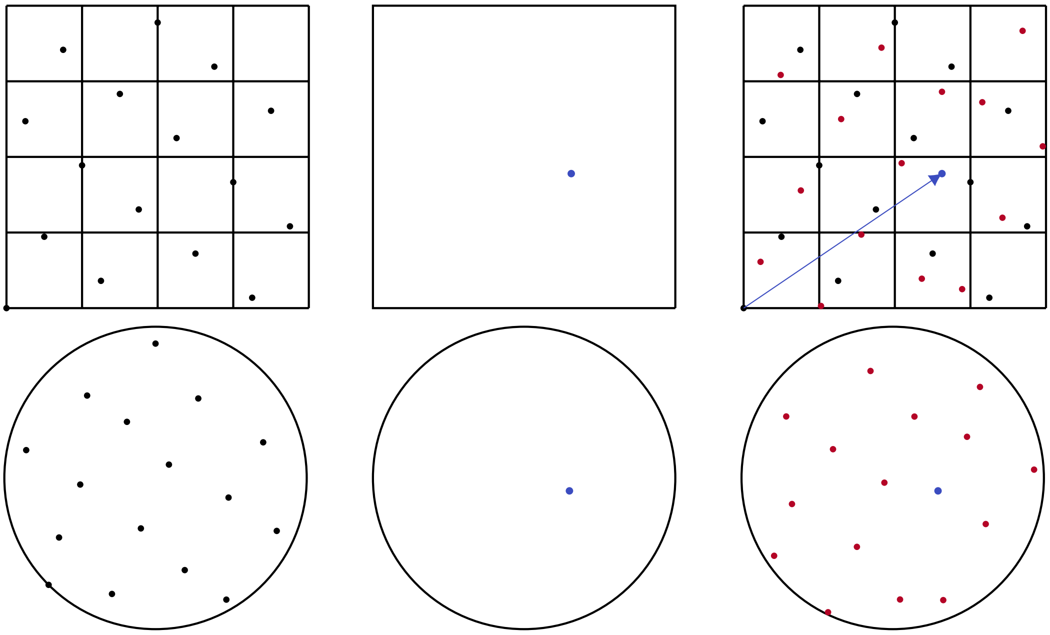


FIGURE C.2 – Shifted Fibonacci grid to unit disk mapping

Bibliography

Stephen J. Adelson and Larry F. Hodges. 1993. Stereoscopic ray-tracing. *The Visual Computer* 10, 3 (01 Mar 1993), 127–144.

Stephen J. Adelson and Larry F. Hodges. 1995. Generating exact ray-traced animation frames by reprojection. *IEEE Computer Graphics and Applications* 15 (1995), 43–52.

Attila T Áfra. 2018. Intel open image denoise (high-performance denoising library for ray tracing). https://www.openimagedenoise.org

Sai Bangaru, Tzu-Mao Li, and Frédo Durand. 2020. Unbiased Warped-Area Sampling for Differentiable Rendering. *ACM Trans. Graph.* 39, 6 (2020), 245:1–245:18.

Pablo Bauszat, Victor Petitjean, and Elmar Eisemann. 2017. Gradient-Domain Path Reusing. *ACM Trans. Graph. (Proc. of SIGGRAPH Asia)* 36, 6 (November 2017).

Philippe Bekaert, Mateu Sbert, and John Halton. 2002. Accelerating Path Tracing by Re-using Paths. In *Proceedings of the 13th Eurographics Workshop on Rendering* (Pisa, Italy) *(EGRW '02)*. Eurographics Association, 125–134.

Philippe Bekaert, Mateu Sbert, and Yves D. Willems. 2000. Weighted Importance Sampling Techniques for Monte Carlo Radiosity. In *Proceedings of the Eurographics Workshop on Rendering Techniques 2000*. Springer-Verlag, Berlin, Heidelberg, 35–46.

Laurent Belcour. 2018. Efficient Rendering of Layered Materials using an Atomic Decomposition with Statistical Operators. *ACM Transactions on Graphics* 37, 4 (2018), 1. https://doi.org/10.1145/3197517.3201289

N. Binder, Sascha Fricke, and A. Keller. 2019. Massively Parallel Path Space Filtering. *ArXiv* abs/1902.05942 (2019).

Benedikt Bitterli, Wenzel Jakob, Jan Novák, and Wojciech Jarosz. 2017. Reversible Jump Metropolis Light Transport Using Inverse Mappings. *ACM Transactions on Graphics* 37, 1 (Oct. 2017). https://doi.org/10.1145/3132704

Benedikt Bitterli, Srinath Ravichandran, Thomas Müller, Magnus Wrenninge, Jan Novák, Steve Marschner, and Wojciech Jarosz. 2018. A radiative transfer framework for non-exponential media. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 37, 6 (Nov. 2018), 225:1–225:17. https://doi.org/10/gfz2cm

Benedikt Bitterli, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. 2020. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 39, 4 (July 2020). https://doi.org/10/gg8xc7

Thomas E. Booth. 2007. Unbiased Monte Carlo Estimation of the Reciprocal of an Integral. *Nuclear Science and Engineering* 156, 3 (2007), 403–407. https://doi.org/10.13182/NSE07-A2707 arXiv:https://doi.org/10.13182/NSE07-A2707

Malik Boughida and Tamy Boubekeur. 2017. Bayesian Collaborative Denoising for Monte Carlo Rendering. *Computer Graphics Forum* 36, 4 (2017), 137–153.

Brent Burley, David Adler, Matt Jen-Yuan Chiang, Hank Driskill, Ralf Habel, Patrick Kelly, Peter Kutz, Yining Karl Li, and Daniel Teece. 2018. The design and evolution of disney's hyperion renderer. *ACM Transactions on Graphics (TOG)* 37, 3 (2018), 1–22.

Brent Burley and Walt Disney Animation Studios. 2012. Physically-based shading at disney. In *SIGGRAPH*, Vol. 2012. 1–7.

Subrahmanyan Chandrasekhar. 1960. *Radiative transfer*. Dover Publications, New York.

Min-Te Chao. 1982. A general purpose unequal probability sampling plan. *Biometrika* 69, 3 (1982), 653–656.

Per Christensen, Julian Fong, Jonathan Shade, Wayne Wooten, Brenden Schubert, Andrew Kensler, Stephen Friedman, Charlie Kilpatrick, Cliff Ramshaw, Marc Bannister, et al. 2018. Renderman: An advanced path-tracing architecture for movie rendering. *ACM Transactions on Graphics (TOG)* 37, 3 (2018), 1–21.

Per H. Christensen. 2015. An Approximate Reflectance Profile for Efficient Subsurface Scattering. In *ACM SIGGRAPH 2015 Talks* (Los Angeles, California) *(SIGGRAPH '15)*. Association for Computing Machinery, New York, NY, USA, Article 25, 1 pages. https://doi.org/10.1145/2775280.2792555

David Cline and Parris Egbert. 2005. A practical introduction to metropolis light transport. *Brigham Young University* (2005).

David Cline, Anshuman Razdan, and Peter Wonka. 2009. A comparison of tabular PDF inversion methods. In *Computer Graphics Forum*, Vol. 28. Wiley Online Library, 154–160.

David Cline, Justin Talbot, and Parris Egbert. 2005. Energy redistribution path tracing. *ACM Transactions on Graphics (TOG)* 24, 3 (2005), 1186–1195.

William G Cochran. 1977. *Sampling techniques*. John Wiley & Sons.

Michael F. Cohen, Shenchang Eric Chen, John R. Wallace, and Donald P. Greenberg. 1988. A Progressive Refinement Approach to Fast Radiosity Image Generation. *SIGGRAPH Comput. Graph.* 22, 4 (June 1988), 75–84. https://doi.org/10.1145/378456.378487

Michael F. Cohen and Donald P. Greenberg. 1985. The Hemi-Cube: A Radiosity Solution for Complex Environments. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '85)*. Association for Computing Machinery, New York, NY, USA, 31–40. https://doi.org/10.1145/325334.325171

Michael F Cohen, John R Wallace, and Pat Hanrahan. 1993. *Radiosity and realistic image synthesis*. Morgan Kaufmann.

Alejandro Conty and Christopher Kulla. 2017. Revisiting Physically Based Shading at Imageworks. In *ACM SIGGRAPH 2017 Courses* (Los Angeles, California) *(SIGGRAPH '17)*. Association for Computing Machinery, New York, NY, USA, Article 7, 8 pages. https://doi.org/10.1145/3084873.3084893

Alejandro Conty Estevez and Christopher Kulla. 2018. Importance sampling of many lights with adaptive tree splitting. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 1, 2 (2018), 1–17.

Robert L Cook. 1986. Stochastic sampling in computer graphics. *ACM Transactions on Graphics (TOG)* 5, 1 (1986), 51–72.

Robert L. Cook, Thomas Porter, and Loren Carpenter. 1984. Distributed Ray Tracing. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '84)*. Association for Computing Machinery, New York, NY, USA, 137–145. `https://doi.org/10.1145/800031.808590`

Robert L Cook and Kenneth E Torrance. 1982. A reflectance model for computer graphics. *ACM Transactions on Graphics (TOG)* 1, 1 (1982), 7–24.

Jean-Marie Cornuet, Jean-Michel Marin, Antonietta Mira, and Christian P Robert. 2012. Adaptive multiple importance sampling. *Scandinavian Journal of Statistics* 39, 4 (2012), 798–812.

Eugene d'Eon. 2016. *A Hitchhiker's Guide to Multiple Scattering*. Technical Report. Technical report.

Eugene d'Eon. 2021. An Analytic BRDF for Materials with Spherical Lambertian Scatterers. (2021). `https://research.nvidia.com/publication/2021-06_An-Analytic-BRDF`

Luc Devroye. 1986. *Non-Uniform Random Variate Generation*. Springer-Verlag New York. `https://doi.org/10.1007/978-1-4613-8643-8`

Glenn Evans and Michael D McCool. 1999. Stratified wavelength clusters for efficient spectral monte carlo rendering. In *Graphics Interface*, Vol. 99. Citeseer.

Luca Fascione, Johannes Hanika, Daniel Heckenberg, Christopher Kulla, Marc Droske, and Jorge Schwarzhaupt. 2019. Path Tracing in Production: Part 1: Modern Path Tracing. In *ACM SIGGRAPH 2019 Courses* (Los Angeles, California) *(SIGGRAPH '19)*. Association for Computing Machinery, New York, NY, USA, Article 19, 113 pages. `https://doi.org/10.1145/3305366.3328079`

Luca Fascione, Johannes Hanika, Mark Leone, Marc Droske, Jorge Schwarzhaupt, Tomáš Davidovič, Andrea Weidlich, and Johannes Meng. 2018. Manuka: A batch-shading architecture for spectral path tracing in movie production. *ACM Transactions on Graphics (TOG)* 37, 3 (2018), 1–18.

Basile Fraboni, Jean-Claude Iehl, Vincent Nivoliers, and Guillaume Bouchard. 2019. Adaptive Multi-view Path Tracing. In *Eurographics Symposium on Rendering - DL-only and Industry Track*, Tamy Boubekeur and Pradeep Sen (Eds.). The Eurographics Association. `https://doi.org/10.2312/sr.20191217`

Basile Fraboni, Antoine Webanck, Nicolas Bonneel, and Jean-Claude Iehl. 2022. Volumetric Multi-view Rendering. *submitted to Eurographics* (2022).

Luis E Gamboa, Adrien Gruson, and Derek Nowrouzezahrai. 2020. An Efficient Transport Estimator for Complex Layered Materials. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 363–371.

Iliyan Georgiev, Jaroslav Křivánek, Tomáš Davidovič, and Philipp Slusallek. 2012. Light Transport Simulation with Vertex Connection and Merging. *ACM Trans.*

Graph. 31, 6, Article 192 (Nov. 2012), 10 pages. https://doi.org/10.1145/2366145.2366211

Iliyan Georgiev, Zackary Misso, Toshiya Hachisuka, Derek Nowrouzezahrai, Jaroslav Křivánek, and Wojciech Jarosz. 2019. Integral Formulations of Volumetric Transmittance. *ACM Trans. Graph.* 38, 6, Article 154 (Nov. 2019), 17 pages. https://doi.org/10.1145/3355089.3356559

Ioannis Gkioulekas, Bei Xiao, Shuang Zhao, Edward H. Adelson, Todd Zickler, and Kavita Bala. 2013. Understanding the role of phase function in translucent appearance. *ACM Trans. Graph.* 32, 5, Article 147 (Oct. 2013), 19 pages. https://doi.org/10.1145/2516971.2516972

Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. 1984. Modeling the Interaction of Light between Diffuse Surfaces. *SIGGRAPH Comput. Graph.* 18, 3 (Jan. 1984), 213–222. https://doi.org/10.1145/964965.808601

Pascal Grittmann, Iliyan Georgiev, Philipp Slusallek, and Jaroslav Křivánek. 2019. Variance-Aware Multiple Importance Sampling. *ACM Trans. Graph. (SIGGRAPH Asia 2019)* 38, 6 (2019), 9. https://doi.org/10.1145/3355089.3356515

Adrien Gruson, Binh-Son Hua, Nicolas Vibert, Derek Nowrouzezahrai, and Toshiya Hachisuka. 2018. Gradient-domain volumetric photon density estimation. *ACM Transactions on Graphics (TOG)* 37 (2018). https://doi.org/10.1145/3197517.3201363

D. Guarnera, G. C. Guarnera, A. Ghosh, C. Denk, and M. Glencross. 2016. BRDF Representation and Acquisition. In *Proceedings of the 37th Annual Conference of the European Association for Computer Graphics: State of the Art Reports* (Lisbon, Portugal) *(EG '16)*. Eurographics Association, Goslar, DEU, 625–650.

Yu Guo, Miloš Hašan, and Shuang Zhao. 2018. Position-free monte carlo simulation for arbitrary layered bsdfs. *ACM Transactions on Graphics (ToG)* 37, 6 (2018), 1–14.

Toshiya Hachisuka. 2013. Five common misconceptions about bias in light transport simulation. (2013).

Eric Haines and Tomas Akenine-Möller (Eds.). 2019. *Ray Tracing Gems*. Apress. http://raytracinggems.com.

Johannes Hanika and Carsten Dachsbacher. 2014. Efficient Monte Carlo Rendering with Realistic Lenses. *Comput. Graph. Forum* 33, 2 (May 2014), 323–332. https://doi.org/10.1111/cgf.12301

Johannes Hanika, Marc Droske, and Luca Fascione. 2015. Manifold Next Event Estimation. *Computer Graphics Forum* (2015). https://doi.org/10.1111/cgf.12681

David Hart, Matt Pharr, Thomas Müller, Ward Lopes, Morgan McGuire, and Peter Shirley. 2020. Practical Product Sampling by Fitting and Composing Warps. In *Eurographics Symposium on Rendering (EGSR'20)*. https://casual-effects.com/research/Hart2020Sampling/index.html

Vlastimil Havran, Cyrille Damez, Karol Myszkowski, and Hans-Peter Seidel. 2003. An Efficient Spatio-temporal Architecture for Animation Rendering. In *ACM SIGGRAPH 2003 Sketches & Applications* (San Diego, California) *(SIGGRAPH '03)*. ACM, 1–1.

Eric Heitz. 2014. Understanding the Masking-Shadowing Function in Microfacet-Based BRDFs. *Journal of Computer Graphics Techniques* 3, 2 (June 2014), 32–91. https://hal.inria.fr/hal-01024289

Eric Heitz. 2018. Sampling the GGX Distribution of Visible Normals. *Journal of Computer Graphics Techniques (JCGT)* 7, 4 (30 November 2018), 1–13. http://jcgt.org/published/0007/04/01/

E. Heitz. 2020. Can't Invert the CDF? The Triangle-Cut Parameterization of the Region under the Curve. *Computer Graphics Forum* 39, 4 (2020), 121–132. https://doi.org/10.1111/cgf.14058 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14058

E. Heitz and E. d'Eon. 2014. Importance Sampling Microfacet-Based BSDFs using the Distribution of Visible Normals. *Computer Graphics Forum* 33, 4 (2014), 103–112. https://doi.org/10.1111/cgf.12417 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12417

Eric Heitz and Jonathan Dupuy. 2015. *Implementing a simple anisotropic rough diffuse material with stochastic evaluation*. Technical Report. Tech. rep., 2015. URL: https://eheitzresearch. wordpress. com/research/. 5, 9.

Eric Heitz, Johannes Hanika, Eugene d'Eon, and Carsten Dachsbacher. 2016. Multiple-Scattering Microfacet BSDFs with the Smith Model. *ACM Trans. Graph.* 35, 4, Article 58 (July 2016), 14 pages. https://doi.org/10.1145/2897824.2925943

Eric Heitz, Stephen Hill, and Morgan McGuire. 2018. Combining analytic direct illumination and stochastic shadows. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. ACM, 2.

Niklas Henrich, J Baerz, Thorsten Grosch, and S Müller. 2011. Accelerating path tracing by eye-path reprojection. In *International Congress on Graphics and Virtual Reality (GRVR)*.

L. G. Henyey and J. L. Greenstein. 1941. Diffuse radiation in the Galaxy. 93 (jan 1941), 70–83. https://doi.org/10.1086/144246

Stephen Hill, Stephen McAuley, Laurent Belcour, Will Earl, Niklas Harrysson, Sébastien Hillaire, Naty Hoffman, Lee Kerley, Jasmin Patry, Rob Pieké, Igor Skliar, Jonathan Stone, Pascal Barla, Mégane Bati, and Iliyan Georgiev. 2020. Physically Based Shading in Theory and Practice. In *ACM SIGGRAPH 2020 Courses* (Virtual Event, USA) *(SIGGRAPH '20)*. Association for Computing Machinery, New York, NY, USA, Article 11, 12 pages. https://doi.org/10.1145/3388769.3407523

Binh-Son Hua, Adrien Gruson, Derek Nowrouzezahrai, and Toshiya Hachisuka. 2017. Gradient-Domain Photon Density Estimation. *Eurographics* (2017).

Binh-Son Hua, Adrien Gruson, Victor Petitjean, Matthias Zwicker, Derek Nowrouzezahrai, Elmar Eisemann, and Toshiya Hachisuka. 2019. A Survey on Gradient-Domain Rendering. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 455–472.

Wenzel Jakob. 2010. Mitsuba renderer.

Wenzel Jakob, Eugene d'Eon, Otto Jakob, and Steve Marschner. 2014. A comprehensive framework for rendering layered materials. *ACM Transactions on Graphics (ToG)* 33, 4 (2014), 1–14.

Wenzel Jakob and Steve Marschner. 2012. Manifold exploration: a Markov Chain Monte Carlo technique for rendering scenes with difficult specular transport. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 58.

Wenzel Jakob, Andrea Weidlich, Andrew Beddini, Rob Pieké, Hanzhi Tang, Luca Fascione, and Johannes Hanika. 2019. Path Tracing in Production: Part 2: Making Movies. In *ACM SIGGRAPH 2019 Courses* (Los Angeles, California) *(SIGGRAPH '19)*. Association for Computing Machinery, New York, NY, USA, Article 20, 41 pages. https://doi.org/10.1145/3305366.3328085

Henrik Wann Jensen, Stephen R Marschner, Marc Levoy, and Pat Hanrahan. 2001. A practical model for subsurface light transport. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 511–518.

Herman Kahn. 1950a. Random sampling (Monte Carlo) techniques in neutron attenuation problems. I. *Nucleonics (US) Ceased publication* 6, See also NSA 3-990 (1950).

Herman Kahn. 1950b. Random Sampling (Monte Carlo) Techniques in Neutron Attenuation Problems. II. *Nucleonics (US) Ceased publication* 6, See also NSA 4-3795 (1950).

Herman Kahn and Andy W Marshall. 1953. Methods of reducing sample size in Monte Carlo computations. *Journal of the Operations Research Society of America* 1, 5 (1953), 263–278.

James T. Kajiya. 1986. The Rendering Equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '86)*. ACM, 143–150.

Ondřej Karlík, Martin Šik, Petr Vévoda, Tomáš Skřivan, and Jaroslav Křivánek. 2019. MIS Compensation: Optimizing Sampling Techniques in Multiple Importance Sampling. *ACM Trans. Graph. (SIGGRAPH Asia 2019)* 38, 6 (2019), 12. https://doi.org/10.1145/3355089.3356565

Csaba Kelemen, László Szirmay-Kalos, György Antal, and Ferenc Csonka. 2002. A simple and robust mutation strategy for the metropolis light transport algorithm. In *Computer Graphics Forum*, Vol. 21. Wiley Online Library, 531–540.

Alexander Keller. 1996. Quasi-Monte Carlo Radiosity. In *Eurographics Workshop on Rendering Techniques*. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.38.54&rep=rep1&type=pdf

Alexander Keller, Ken Dahm, and Nikolaus Binder. 2014. Path Space Filtering. In *ACM SIGGRAPH 2014 Talks* (Vancouver, Canada) *(SIGGRAPH '14)*. Association for Computing Machinery, New York, NY, USA, Article 68, 1 pages. https://doi.org/10.1145/2614106.2614149

Alexander Keller, Iliyan Georgiev, Abdalla Ahmed, Per Christensen, and Matt Pharr. 2019. My Favorite Samples. https://sites.google.com/view/myfavoritesamples. In *ACM SIGGRAPH 2019 Courses* (Los Angeles, California) *(SIGGRAPH '19)*. Association for Computing Machinery, New York, NY, USA, Article 15, 271 pages. https://doi.org/10.1145/3305366.3329901

Markus Kettunen, Eugene d'Eon, Jacopo Pantaleoni, and Jan Novak. 2021. An unbiased ray-marching transmittance estimator. (2021). arXiv:2102.10294 [cs.GR] https://research.nvidia.com/publication/2021-06_An-Unbiased-Ray-Marching

Markus Kettunen, Marco Manzi, Miika Aittala, Jaakko Lehtinen, Frédo Durand, and Matthias Zwicker. 2015. Gradient-domain Path Tracing. *ACM Trans. Graph.* 34, 4, Article 123 (July 2015), 13 pages.

Craig Kolb, Don Mitchell, and Pat Hanrahan. 1995. A Realistic Camera Model for Computer Graphics. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*. Association for Computing Machinery, New York, NY, USA, 317–324. https://doi.org/10.1145/218380.218463

Ivo Kondapaneni, Petr Vévoda, Pascal Grittmann, Tomáš Skřivan, Philipp Slusallek, and Jaroslav Křivánek. 2019. Optimal Multiple Importance Sampling. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2019)* 38, 4 (July 2019), 37:1–37:14. https://doi.org/10.1145/3306346.3323009

Christopher Kulla and Marcos Fajardo. 2012. Importance sampling techniques for path tracing in participating media. In *Computer graphics forum*, Vol. 31. Wiley Online Library, 1519–1528.

J.H. Lambert. 1760. *Photometria sive de mensura et gradibus luminis, colorum et umbrae*. sumptibus vidvae E. Klett, typis C.P. Detleffsen. https://books.google.fr/books?id=JdkTAAAAQAAJ

Jaakko Lehtinen, Tero Karras, Samuli Laine, Miika Aittala, Frédo Durand, and Timo Aila. 2013. Gradient-domain Metropolis Light Transport. *ACM Trans. Graph.* 32, 4, Article 95 (July 2013), 12 pages.

G Peter Lepage. 1978. A new algorithm for adaptive multidimensional integration. *J. Comput. Phys.* 27, 2 (1978), 192–203.

Louis Lorenz. 1890. *Lysbevægelsen i og uden for en af plane Lysbølger belyst Kugle*.

Guillaume Loubet, Tizian Zeltner, Nicolas Holzschuch, and Wenzel Jakob. 2020. Slope-space integrals for specular next event estimation. *ACM Transactions on Graphics* 39, 6 (Nov. 2020), 1–13. https://doi.org/10.1145/3414685.3417811

Marco Manzi, Markus Kettunen, Miika Aittala, Jaakko Lehtinen, Frédo Durand, and Matthias Zwicker. 2015. Gradient-Domain Bidirectional Path Tracing. In *Eurographics Symposium on Rendering 2015*. VK: Lehtinen, J.; RUIS; HICT.

Marco Manzi, Markus Kettunen, Frédo Durand, Matthias Zwicker, and Jaakko Lehtinen. 2016. Temporal Gradient-domain Path Tracing. *ACM Trans. Graph.* 35, 6, Article 246 (Nov. 2016), 9 pages.

Adam Marrs, Peter Shirley, , and Ingo Wald (Eds.). 2021. *Ray Tracing Gems II*. Apress. http://raytracinggems.com/rtg2.

William Ross McCluney. 2014. *Introduction to radiometry and photometry*. Artech House.

Morgan McGuire, Julie Dorsey, Eric Haines, John F. Hughes, Steve Marschner, Matt Pharr, and Peter Shirley. 2020. A Taxonomy of Bidirectional Scattering Distribution Function Lobes for Rendering Engineers. In *8th annual Workshop* *on Material Appearance Modeling*. 4. https://casual-effects.com/research/McGuire2020BSDF/

Don McLeish. 2011. A general method for debiasing a Monte Carlo estimator. *Monte Carlo Methods and Applications* 17, 4 (2011), 301–315. https://doi.org/doi:10.1515/mcma.2011.013

Àlex Méndez Feliu, Mateu Sbert, and László Szirmay-Kalos. 2006. Reusing frames in camera animation. *WSCG* 14, 1-3 (2006).

Nicholas Metropolis et al. 1987. The beginning of the Monte Carlo method. *Los Alamos Science* 15, 584 (1987), 125–130.

Nicholas Metropolis and Stanislaw Ulam. 1949. The monte carlo method. *Journal of the American statistical association* 44, 247 (1949), 335–341.

Gustav Mie. 1908. Beiträge zur Optik trüber Medien, speziell kolloidaler Metallösungen. *Annalen der physik* 330, 3 (1908), 377–445.

Bailey Miller, Iliyan Georgiev, and Wojciech Jarosz. 2019. A null-scattering path integral formulation of light transport. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 38, 4 (July 2019). https://doi.org/10/gf6rzb

Rosana Montes and Carlos Ureña. 2012. An overview of BRDF models. *University of Grenada, Technical Report LSI-2012-001* (2012).

Pierre Moreau and Petrik Clarberg. 2019. Importance sampling of many lights on the GPU. In *Ray tracing gems*. Springer, 255–283.

Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. 2019. Neural Importance Sampling. *ACM Trans. Graph.* 38, 5, Article 145 (Oct. 2019), 19 pages. https://doi.org/10.1145/3341156

Ken Museth. 2021. NanoVDB: A GPU-Friendly and Portable VDB Data Structure For Real-Time Rendering And Simulation. In *ACM SIGGRAPH 2021 Talks* (Virtual Event, USA) *(SIGGRAPH '21)*. Association for Computing Machinery, New York, NY, USA, Article 1, 2 pages. https://doi.org/10.1145/3450623.3464653

Ken Museth, Nick Avramoussis, and Dan Bailey. 2019. OpenVDB. In *ACM SIGGRAPH 2019 Courses* (Los Angeles, California) *(SIGGRAPH '19)*. Association for Computing Machinery, New York, NY, USA, Article 17, 56 pages. https://doi.org/10.1145/3305366.3328070

F Kenton Musgrave, Darwyn Peachey, Ken Perlin, and Steven Worley. 1994. *Texturing and modeling: a procedural approach*. Academic Press Professional.

Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. 2019. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–17.

Jan Novák, Iliyan Georgiev, Johannes Hanika, and Wojciech Jarosz. 2018. Monte Carlo methods for volumetric light transport simulation. *Computer Graphics Forum (Proceedings of Eurographics - State of the Art Reports)* 37, 2 (May 2018). https://doi.org/10/gd2jqq

Michael Oren and Shree K. Nayar. 1994. Generalization of Lambert's Reflectance Model. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '94)*. Association for Computing Machinery, New York, NY, USA, 239–246. https://doi.org/10.1145/192161.192213

Hisanari Otsu, Anton S. Kaplanyan, Johannes Hanika, Carsten Dachsbacher, and Toshiya Hachisuka. 2017. Fusing State Spaces for Markov Chain Monte Carlo Rendering. *ACM Transactions on Graphics (Proc. of SIGGRAPH)* 36, 4, Article 74 (2017), 74:1–74:10 pages.

Art B. Owen. 2013. *Monte Carlo theory, methods and examples*. https://statweb.stanford.edu/~owen/mc/

Jacopo Pantaleoni. 2017. Charted Metropolis Light Transport. *ACM Trans. Graph.* 36, 4, Article 75 (July 2017), 14 pages. https://doi.org/10.1145/3072959.3073677

Steven G Parker, James Bigler, Andreas Dietrich, Heiko Friedrich, Jared Hoberock, David Luebke, David McAllister, Morgan McGuire, Keith Morley, Austin Robison, et al. 2010. Optix: a general purpose ray tracing engine. *Acm transactions on graphics (tog)* 29, 4 (2010), 1–13.

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

Jordan Peck. 2020. FastNoise Lite: a portable open source noise generation library. (2020). https://github.com/Auburn/FastNoiseLite

Vincent Pegoraro. 2016. *Handbook of Digital Image Synthesis: Scientific Foundations of Rendering*. CRC Press.

Matt Pharr. 2018. Physically Based Rendering: From Theory to Implementation – Version 4. https://github.com/mmp/pbrt-v4

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

Michael JD Powell and J Swann. 1966. Weighted uniform sampling—a Monte Carlo technique for reducing variance. *IMA Journal of Applied Mathematics* 2, 3 (1966), 228–236.

Hao Qin, Xin Sun, Qiming Hou, Baining Guo, and Kun Zhou. 2015. Unbiased Photon Gathering for Light Transport Simulation. *ACM Trans. Graph.* 34, 6, Article 208 (Oct. 2015), 14 pages. https://doi.org/10.1145/2816795.2818119

John William Strutt Baron Rayleigh. 1871. *On the scattering of light by small particles*.

Jaume Rigau, Miquel Feixas, and Mateu Sbert. 2003. Refinement Criteria Based on F-divergences. In *Proceedings of the 14th Eurographics Workshop on Rendering* (Leuven, Belgium) *(EGRW '03)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 260–269.

Christian Robert and George Casella. 2013. *Monte Carlo statistical methods*. Springer Science & Business Media.

Reuven Y Rubinstein. 1997. Optimization of computer simulation models with rare events. *European Journal of Operational Research* 99, 1 (1997), 89–112.

Iman Sadeghi, Adolfo Munoz, Philip Laven, Wojciech Jarosz, Francisco Seron, Diego Gutierrez, and Henrik Wann Jensen. 2012. Physically-based Simulation of Rainbows. *ACM Transactions on Graphics (Presented at SIGGRAPH)* 31, 1 (Feb. 2012), 3:1–3:12. https://doi.org/10/gfzndf

Mateu Sbert, Vlastimil Havran, and Laszlo Szirmay-Kalos. 2018. Multiple importance sampling revisited: breaking the bounds. *EURASIP Journal on Advances in Signal Processing* 2018, 1 (27 Feb 2018), 15.

Jorge Schwarzhaupt. 2019. Path Tracing in Production: Part 1: Modern Path Tracing Chapter 4: Finding good paths. In *ACM SIGGRAPH 2019 Courses* (Los Angeles, California) *(SIGGRAPH '19)*. Association for Computing Machinery, New York, NY, USA, Article 19, 113 pages. https://doi.org/10.1145/3305366.3328079

Subodh Kumar Sharma. 2015. *A review of approximate analytic light-scattering phase functions*. Springer Berlin Heidelberg, Berlin, Heidelberg, 53–100. https://doi.org/10.1007/978-3-642-37985-7_2

Peter Shirley and Kenneth Chiu. 1997. A low distortion map between disk and square. *Journal of graphics tools* 2, 3 (1997), 45–52.

Peter Shirley and Changyaw Wang. 1994. Direct lighting calculation by monte carlo integration. In *Photorealistic Rendering in Computer Graphics*. Springer, 52–59.

Martin Šik and Jaroslav Křivánek. 2018. Survey of Markov chain Monte Carlo methods in light transport simulation. *IEEE transactions on visualization and computer graphics* 26, 4 (2018), 1821–1840.

François X Sillion and Claude Peuch. 1994. Radiosity & global illumination. (1994).

Jerome Spanier. 1979. A new family of estimators for random walk problems. *IMA Journal of Applied Mathematics* 23, 1 (1979), 1–31.

Jerome Spanier and Earl H Maize. 1994. Quasi-random methods for estimating integrals using relatively small samples. *SIAM review* 36, 1 (1994), 18–44.

Justin Talbot, David Cline, and Parris Egbert. 2005. Importance Resampling for Global Illumination. In *Eurographics Symposium on Rendering (2005)*, Kavita Bala and Philip Dutre (Eds.). The Eurographics Association. https://doi.org/10.2312/EGWR/EGSR05/139-146

Lorenzo Tessari, Johannes Hanika, and Carsten Dachsbacher. 2017. Local Quasi-Monte Carlo Exploration. In *Eurographics Symposium on Rendering - Experimental Ideas and Implementations*, Matthias Zwicker and Pedro Sander (Eds.). The Eurographics Association. https://doi.org/10.2312/sre.20171196

Yusuke Tokuyoshi and Takahiro Harada. 2016. Stochastic light culling. *Journal of Computer Graphics Techniques Vol* 5, 1 (2016).

Hale F. Trotter and John W. Tukey. 1956. Conditional Monte Carlo for normal samples. In *Proc. Symp. on Monte Carlo Methods*. John Wiley and Sons, 64–79. http://stanford.edu/~jugander/rare/TrotterTukey1954_ConditionalMonteCarlo.pdf

Emmanuel Turquin. 2017. *Practical multiple scattering compensation for microfacet models*. Technical Report. Industrial Light and Magic. https://blog.selfshadow.com/publications/turquin/ms_comp_final.pdf

Eric Veach. 1998. *Robust Monte Carlo Methods for Light Transport Simulation*. Ph.D. Dissertation. Stanford University, Stanford, CA, USA. Advisor(s) Guibas, Leonidas J. AAI9837162.

Eric Veach and Leonidas J. Guibas. 1995. Optimally Combining Sampling Techniques for Monte Carlo Rendering. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*. Association for Computing Machinery, New York, NY, USA, 419–428. https://doi.org/10.1145/218380.218498

Eric Veach and Leonidas J Guibas. 1997. Metropolis light transport. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM, 65–76.

Thijs Vogels, Fabrice Rousselle, Brian Mcwilliams, Gerhard Röthlin, Alex Harvill, David Adler, Mark Meyer, and Jan Novák. 2018. Denoising with Kernel Prediction and Asymmetric Loss Functions. *ACM Transactions on Graphics (TOG)* 37, 4 (2018).

Jiří Vorba, Johannes Hanika, Sebastian Herholz, Thomas Müller, Jaroslav Křivánek, and Alexander Keller. 2019. Path Guiding in Production. https://cgg.mff.cuni.cz/{~}jaroslav/papers/2019-path-guiding-course/2019-vorba-path-guiding-course-notes.pdf. In *ACM SIGGRAPH 2019 Courses* (Los Angeles, California) *(SIGGRAPH '19)*. ACM, New York, NY, USA, Article 18, 77 pages. https://doi.org/10.1145/3305366.3328091

Jiří Vorba and Jaroslav Křivánek. 2016. Adjoint-driven Russian Roulette and Splitting in Light Transport Simulation. *ACM Trans. Graph.* 35, 4, Article 42 (July 2016), 11 pages. https://doi.org/10.1145/2897824.2925912

Ingo Wald, Sven Woop, Carsten Benthin, Gregory S Johnson, and Manfred Ernst. 2014. Embree: a kernel framework for efficient CPU ray tracing. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–8.

Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. 2007. Microfacet Models for Refraction through Rough Surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques* (Grenoble, France) *(EGSR'07)*. Eurographics Association, Goslar, DEU, 195–206.

Pascal Weber, Johannes Hanika, and Carsten Dachsbacher. 2017. Multiple Vertex Next Event Estimation for Lighting in dense, forward-scattering Media. *Computer Graphics Forum (Proceedings of Eurographics)* (April 2017).

Rex West, Iliyan Georgiev, Adrien Gruson, and Toshiya Hachisuka. 2020. Continuous Multiple Importance Sampling. *ACM Transactions on Graphics* 39, 4 (2020), 12. https://doi.org/10.1145/3386569.3392436

Alexander Wilkie, Sehera Nawaz, Marc Droske, Andrea Weidlich, and Johannes Hanika. 2014. Hero wavelength spectral sampling. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 123–131.

Sven Woop, Attila T. Áfra, and Carsten Benthin. 2017. STBVH: A Spatial-temporal BVH for Efficient Multi-segment Motion Blur. In *Proceedings of High Performance Graphics* (Los Angeles, California) *(HPG '17)*. ACM, New York, NY, USA, Article 8, 8 pages.

Mengqi Xia, Bruce Walter, Christophe Hery, and Steve Marschner. 2020. Gaussian Product Sampling for Rendering Layered Materials. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 420–435.

Qing Xu and Mateu Sbert. 2007. A New Way to Re-using Paths. In *Computational Science and Its Applications (ICCSA 2007)*. 741–750.

Cem Yuksel. 2020. Stochastic Lightcuts for Sampling Many Lights. *IEEE Transactions on Visualization and Computer Graphics* (2020).

Tizian Zeltner, Iliyan Georgiev, and Wenzel Jakob. 2020. Specular Manifold Sampling for Rendering High-Frequency Caustics and Glints. *Transactions on Graphics (Proceedings of SIGGRAPH)* 39, 4 (July 2020). `https://doi.org/10.1145/3386569.3392408`

Henning Zimmer, Fabrice Rousselle, Wenzel Jakob, Oliver Wang, David Adler, Wojciech Jarosz, Olga Sorkine-Hornung, and Alexander Sorkine-Hornung. 2015. Path-space Motion Estimation and Decomposition for Robust Animation Filtering. *Computer Graphics Forum* 34, 4 (2015), 131–142.

Matthias Zwicker, Wojciech Jarosz, Jaakko Lehtinen, Bochang Moon, Ravi Ramamoorthi, Fabrice Rousselle, Pradeep Sen, Cyril Soler, and Sung-Eui Yoon. 2015. Recent Advances in Adaptive Sampling and Reconstruction for Monte Carlo Rendering. *Computer Graphics Forum (Proceedings of Eurographics - State of the Art Reports)* 34, 2 (may 2015), 667–681.

FOLIO ADMINISTRATIF

THÈSE DE L'UNIVERSITÉ DE LYON OPÉRÉE AU SEIN DE L'INSA LYON

NOM: FRABONI **DATE DE SOUTENANCE:** 15/12/2021
PRÉNOM: Basile

TITRE: Self-normalized estimators and joint path construction for efficient multi-view rendering

NATURE: Doctorat **NUMÉRO D'ORDRE:** 2021LYSEI103
ECOLE DOCTORALE: Ecole Doctorale EDA512 – Informatique et Mathématiques de Lyon
SPÉCIALITÉ: Informatique

RESUMÉ:

La simulations du transport de lumière physiquement réaliste est progressivement devenue l'approche standard dans l'industrie de la production d'images. L'algorithme de Path Tracing et ses variantes sont utilisés pour leur capacité à simuler des phénomènes d'éclairage complexe.

Cependant, ces simulations nécessitent d'explorer l'ensemble des chemins reliant une source lumineuse au capteur de la caméra. La construction de ces chemins est un processus séquentiel complexe et il est couvent nécéssaire de construire et évaluer un très grand nombre de chemins pour atteindre des niveaux de bruit acceptables dans les images. Ceci est encore plus problématique lorsque l'on ajoute des effets de production tels que le flou de mouvement, la profondeur de champ et le rendu volumétrique. Un autre aspect important est que nous devons souvent calculer plusieurs images d'une même scène, par exemple lors du rendu de paires stéréo; d'images lenticulaires, d'images de champs lumineux et de stéréogrammes holographiques pour visualiser plusieurs points de vue; de trajectoires de caméra animées pour les visites virtuelles ou le rendu de séquences animées.

Dans cette thèse, notre objectif est d'accélérer le rendu de plusieurs points de vue lors d'une même simulation en exploitant la cohérence entre les caméras. Ceci représente un défi car les méthodes existantes pour réutiliser les chemins entre plusieurs vues introduisent du bruit et du biais visible dans les images, et ne sont pas adaptées pour tous les effets de production, les matériaux, les surfaces et les volumes qu'une scène peut inclure. Nous développons un nouvel algorithme unidirectionnel pour rendre conjointement plusieurs images d'une meme scène. Nous introduisons de nouvelles méthodes pour transformer et réutiliser les chemins d'une caméra à l'autre en présence de milieux participants et pour générer des sous-chemins qui contribuent le mieux à un sous-ensemble d'observateur, ainsi qu'un un nouvel estimateur de Monte Carlo pour combine correctement les contributions de ces chemins. Nous démontrons sur plusieurs scènes comprenant de la géométrie complexe, des matériaux complexes, des milieux participants et des effets de production que cette méthode réduit efficacement le bruit par rapport aux calculs image par image à temps de calcul équivalent.

MOTS-CLÉS: Rendering – Path tracing – Path reusing – Multi-view rendering – Volume rendering – Monte Carlo techniques

LABORATOIRE DE RECHERCHE: Laboratoire d'InfoRmatique en Image et Systèmes d'information – UMR 5205 CNRS

DIRECTRICE DE THÈSE: Véronique Eglin
PRÉSIDENT DE JURY:
COMPOSITION DU JURY:

Tamy Boubekeur – George Drettakis – Nicolas Holzschuch
Mathias Paulin – Véronique Eglin – Jean-Claude Iehl