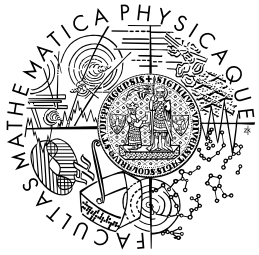


FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

DOCTORAL THESIS

Petr Vévoda

Advanced Monte Carlo methods in Image Synthesis

Department of Software and Computer Science Education

Supervisor of the doctoral thesis: doc. Dr. Alexander Wilkie

Study programme: Computer Science

Study branch: Visual Computing and Computer Games

Prague 2023

I declare that I carried out this doctoral thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ....

Author's signature

I would like to express my sincere gratitude to my advisor, Alexander Wilkie, for all his support and patience in the final years of my doctoral studies. I am also grateful to my former advisor Jaroslav Křivánek. He not only helped me with several of my publications but it was him who brought me to computer graphics research in the first place. His presence is dearly missed.

Next, I would like to thank Ivo Kondapaneni for the great collaboration we had that resulted in the two of the papers presented in this thesis. I also thank all my other colleagues from the computer graphics group at the Faculty of Mathematics and Physics as well as my colleagues from Chaos Czech for creating a great working environment and helping me with many of my publications.

Last but not least, I would like to thank my family that has always supported and encouraged me throughout my studies, and my girlfriend Tereza for her love and unlimited patience.

This work was supported by the Charles University Grant Agency projects GAUK 1172416 and 996218, by the grant SVV-2017-260452 and SVV-2017-260588, and by the Czech Science Foundation grants GACR 16-18964S, 19-07626S and 22-22875S. This work has also received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 642841 (DISTRO) and No 956585 (PRIME). Furthermore, this work was supported by Chaos Czech, through the Intel Graphics and Visualization Institute at Charles University and by The Czech Ministry of Education, Youth and Sports from the Large Infrastructures for Research, Experimental Development and Innovations project "IT4Innovations National Supercomputing Center – LM201507". The author also acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC resources that have contributed to the research results reported in this thesis.

To my father

Title: Advanced Monte Carlo methods in Image Synthesis

Author: Petr Vévoda

Department: Department of Software and Computer Science Education

Supervisor: doc. Dr. Alexander Wilkie, Department of Software and Computer Science Education

Abstract: Monte Carlo (MC) integration is an essential tool in many fields of science. In image synthesis, it has enabled photorealistic rendering results via physically based light transport simulation. However, an inherent problem of MC integration is variance causing noise in rendered images. This thesis presents three methods, each taking a different approach to variance reduction in rendering. The first approach focuses on improving the sampling technique. An adaptive solution is proposed for unbiased direct illumination sampling, employing Bayesian regression and a novel statistical model of direct illumination to achieve robustness. This method was integrated into a production renderer, demonstrating both its theoretical soundness and practical utility. The second approach explores the combination of multiple sampling techniques via multiple importance sampling (MIS). Optimal weighting functions are derived, proving to minimize the variance of MIS estimators. The new weights outperform all common heuristics and provide novel design considerations for selecting appropriate sampling techniques in integration problems. Finally, the third approach involves pre-computation to handle challenging scenarios effectively. Pre-computed reference images of a clear sky are used to create a high-quality fitted model, allowing any renderer to achieve realistic sky appearance without any atmospheric simulation overhead. An extension covering the full spectral range of terrestrial solar irradiance is also introduced, enabling usage of such pre-computed models for purposes other than renderings intended to mimic the perception of human observers, such as thermal analysis, and photovoltaic plant yield simulations. The three presented methods significantly improve rendering efficiency and quality, and contribute valuable insights to the field of MC integration in image synthesis.

Keywords: computer graphics, image synthesis, rendering, light transport simulation, Monte Carlo methods

Contents

Introduction

Monte Carlo (MC) integration is an essential tool in many fields of science and engineering [Kalos and Whitlock, 2008]. In the context of image synthesis, light transport simulation based on MC integration has become a standard approach to physically based rendering [Veach, 1997; Pharr et al., 2016]. This approach is conceptually simple, yet it is flexible and allows for photorealistic results. However, an inherent problem of MC integration is variance, which leads to noise in rendered images. The noise fades away with increasing render time, but for complex scenes it may take hours for the rendering to converge to a noise-free image. Therefore, many methods have been proposed to improve the performance in various situations. In this thesis, we present 3 such methods, each taking a different approach to decreasing the variance of MC integration in rendering.

Approach 1: A better sampling technique

For the first approach, we need to review the basics of MC integration. Let $F = \int_D f(x)\,\mathrm{d}x$ be the integral of a function $f : D \to \mathbb{R}$ over the domain D, and let there be a *sampling technique* for generating random samples from D following the probability density p such that $f(x) \neq 0 \Rightarrow p(x) \neq 0$. Then the importance sampling MC estimator $\langle F \rangle = f(X)/p(X)$, where the random variable X is distributed according to p, is unbiased, i.e., its expected value $\mathrm{E}[F]$ equals F. The shape of p has a dramatic impact on the estimator's variance $\mathrm{Var}[\langle F \rangle]$: the closer p is to being proportional to the integrand f, the lower the variance.

One way of decreasing the variance of MC integration is therefore obvious: finding a better sampling technique with its probability density as close to being proportional to the integrand as possible. This way has been explored in many research works but there is still room for improvement, one such area being direct illumination calculation. It is an important component of any physically based renderer with a substantial impact on the overall performance, yet it has received less attention in research than indirect illumination.

In Chapter 1 we present a novel adaptive solution for unbiased direct illumination sampling, based on online learning of the light selection probability distributions. We provide a formulation of the learning process as Bayesian regression [Bishop, 2006], based on a new, specifically designed statistical model of direct illumination. The net result is a set of regularization strategies that prevents overfitting and ensures robustness even in early stages of calculation, when the observed information is sparse. We make the method scalable by adopting a light clustering strategy from the Lightcuts method [Walter et al., 2005], and further reduce variance through the use of control variates [Kalos and Whitlock, 2008]. As a main design feature, the resulting algorithm is virtually free of any preprocessing, which enables its use for interactive progressive rendering. This was a hard constraint during the algorithm development as it was driven by practical needs of the established production renderer Corona [Chaos Czech a.s., 2023]. The method has been successfully used there to this day, which proves it to be not only theoretically sound, but useful in practice as well.

The content of Chapter 1 is an extended version of the publication **Bayesian**

online regression for adaptive direct illumination sampling by Vévoda et al. [2018]. The author shares the first authorship with Ivo Kondapaneni, the author's contribution was the initial design of the complete method and all the implementation work (see List of publications for more details).

Approach 2: A better combination of techniques

Finding a single sampling technique that would be a good match for the entire integrand is sometimes infeasible. In such a case multiple sampling techniques can be used, each of which could be a good match to a different feature of the integrand. This method is called multiple importance sampling (MIS). It was introduced by Veach and Guibas [1995] and became a key technique for achieving robustness of MC estimators in computer graphics and other fields. Samples are drawn from all the techniques and then combined using *weighting functions*. A set of weighting functions known as the balance heuristic was suggested as a de facto universal solution, as no other weights were claimed to yield substantially lower variance [Veach and Guibas, 1995]. However, a truly optimal set of weighting functions has not been known.

In Chapter 2 we derive optimal weighting functions for MIS that provably minimize the variance of an MIS estimator, given a set of sampling techniques. We show that the resulting variance reduction over the balance heuristic can be higher than predicted by the variance bounds derived by Veach and Guibas [1995], who assumed only non-negative weights in their proof. We theoretically analyse the variance of the optimal MIS weights and show the relation to the variance of the balance heuristic. Furthermore, we establish a connection between the new weighting functions and control variates, as previously applied to mixture sampling. We apply the new optimal weights to integration problems in light transport and show that they allow for new design considerations when choosing the appropriate sampling techniques for a given integration problem.

The content of Chapter 2 is an extended version of the publication **Optimal Multiple Importance Sampling** by Kondapaneni et al. [2019]. The author shares the first authorship with Ivo Kondapaneni, the author's contribution was the discovery of the limitation of the balance heuristic variance bounds, the design of applications of the optimal weights including new sampling techniques, and most of the implementation work.

Approach 3: Pre-computation

In cases when neither a good sampling technique nor their good combination is enough to produce desired results in reasonable time, the most practical approach might be to pre-compute the difficult parts. Imagine rendering an outdoor scene under a clear sky. If both precise control over the illumination as well as high accuracy of the depicted sky is needed, neither image based lightning nor existing analytical sky models could be used. A brute force simulation of light transport in the atmosphere had to be employed, which required not only a large amount of rendering time but also a specialized knowledge of the composition of the atmosphere and light propagation in it.

The situation changed when the Prague Sky Model [Wilkie et al., 2021] was published, which we review in Chapter 3. Using realistic scatterer distribution data from atmospheric measurements, Wilkie et al. pre-computed a large set of reference images of a clear sky for a wide range of parameters. These images were then compressed via tensor decomposition into a fitted model of sky dome radiance and attenuation. This model considerably improves on the visual realism of existing analytical clear sky models, as well as of interactive methods that are based on approximating atmospheric light transport. It also provides features not found in fitted models so far: radiance patterns for post-sunset conditions, in-scattered radiance and transmittance values for finite viewing distances, an observer altitude resolved model that includes downward-looking viewing directions, as well as polarisation information. At the same time, the model remains easy to use. It has been implemented and used in the Corona renderer demonstrating its practicality for industry.

An initial version of the Prague Sky Model was published in the doctoral thesis **Atmospheric Rendering** by Hošek [2019], the final version was then published in the paper **A Fitted Radiance and Attenuation Model for Realistic Atmospheres** by Wilkie et al. [2021]. The author shares the first authorship of the paper with Alexander Wilkie, the author's contribution is described in Chapter 3 alongside the model.

Since the Prague Sky Model was designed for traditional photorealistic rendering, it provides data only for the slightly extended visible range of wavelengths. In Chapter 3 we also present an extension of the Prague Sky Model that covers the entire spectral range of terrestrial solar irradiance, which enables a more specialized usage like accurate simulations of photovoltaic plant yield or thermal properties of buildings. This extension is based on the publication **A Wide Spectral Range Sky Radiance Model** by Vévoda et al. [2022] where the author was the primary investigator.

1. A better sampling technique

In this chapter, we describe our first approach to decreasing the variance of MC integration in rendering: finding a better sampling technique with its probability density as close to being proportional to the integrand as possible.

Traditionally, the *indirect* illumination component has been held responsible for the undesirable image noise produced by MC renderers, which is probably why the *direct* illumination component has received disproportionately less attention in research. However, many scenes in digital production feature complex lighting setups, and practical experience shows that it is often direct illumination that is responsible for the majority of image noise. Therefore, we tried to find a better sampling technique for direct illumination estimation.

Specifically, we address the problem of randomly choosing an appropriate light source for a given scene location, so that variance of the direct illumination estimator is minimized. This could be achieved by choosing lights with probability proportional to their respective contributions, but these are unknown at the outset, they are costly to evaluate and difficult to predict. This is true especially with regard visibility, since it can be discontinuous and its evaluation involves expensive ray casting. But ignoring the role visibility plays in the contribution of lights can have a large impact on the estimator variance as demonstrated in Figure 1.1. It shows a room lit by two light sources: one that is very strong but only illuminates a small part of the room (the sun), and one that is much weaker but covers most of the room (a ceiling light). Therefore, choosing lights proportionally to their unoccluded contribution (i.e., without taking their visibility into account) will strongly prefer the sun over the ceiling light even in sun's shadow. As a result, light sampling in most of the scene will be far from optimal and will produce strong noise as shown in left part of the figure.

One possible solution would involve constructing the light sampling distributions in a preprocessing step [Georgiev et al., 2012a]. However, long preprocessing disqualifies any form of *interactive rendering* – a crucial feature of any modern progressive renderer, a feature that we consider a hard constraint in our work motivated by practical needs of the Corona renderer [Chaos Czech a.s., 2023]. Such preprocessing can be avoided by learning from the observed samples during rendering, and our work follows this path. This is hardly a new idea in the general MC context and it has been used for direct illumination sampling [Donikian et al., 2006]. Unresolved challenges remain, though, such as how to ensure robustness, especially in the early stages of rendering, when the collected data are sparse.

The above concerns are common to most adaptive MC methods, and we address them through a systematic treatment based on Bayesian modelling. We formulate the learning process as maximum a posteriori (MAP) regression based on a new statistical model of direct illumination that explicitly models the effect of visibility. The prior distribution is modelled using estimates of lights' unoccluded contributions computed at a small cost. The net result of this formulation are regularization strategies that prevent overfitting and enable meaningful use of the collected samples even in early stages of rendering. Our regression model captures spatial variation of illumination, which enables aggregating statistics over relatively large spatial regions, and, in turn, ensures a fast learning rate.

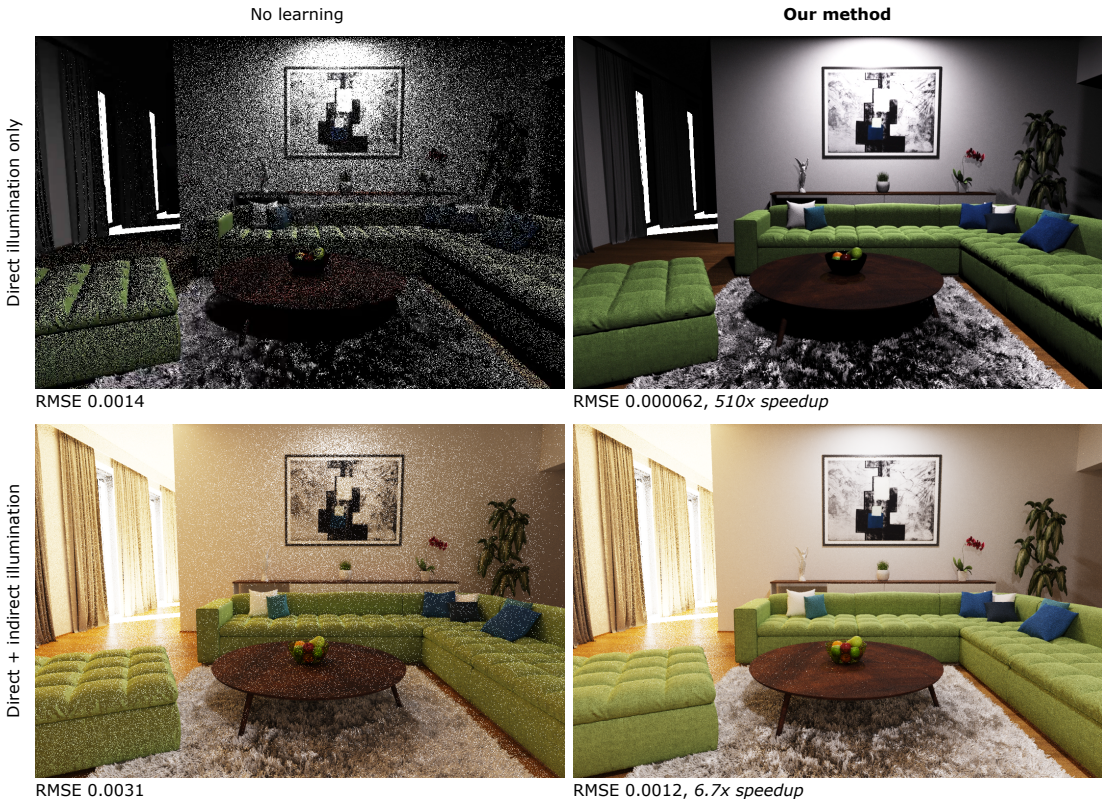


Figure 1.1: An equal-time comparison (60 s) of our proposed learning-based direct illumination sampling method (right column) and a baseline sampling method without learning (left column) on an image containing just direct illumination (top row) and both direct and indirect illumination (bottom row). While both methods start off by sampling lights proportionally to rough estimates of their unoccluded contribution, our method progressively incorporates information about their actual contributions, including visibility, dramatically reducing image noise. As a result, our method renders the direct illumination component 510× faster and achieves 6.7× speedup when rendering both direct and indirect illumination.

Furthermore, we show that sampling lights proportionately to their expected contribution can in fact be far from optimal. The reason is the additional variance due to computing illumination from each individual light source, once it has been selected. We derive the optimal sampling strategy for such *nested estimators* and apply it to the light selection problem.

Finally, to achieve a scalable solution we build upon the light clustering strategies from previous work [Wang and Akerlund, 2009; Walter et al., 2005], and we further reduce variance by using the gathered statistics to construct a control variate [Kalos and Whitlock, 2008]. The resulting algorithm is unbiased and virtually free of any preprocessing, which enables its use in an interactive progressive renderer, while the online learning enables superlinear convergence, especially in the early stages of rendering. Figure 1.1 shows an example of the algorithm performance.

1.1 Previous work

Direct illumination computation Different ways of improving the performance of direct illumination computation have been explored. One idea is to speed up evaluation of a single light contribution, the cost of which is often dominated by determining its visibility. This could be achieved by skipping visibility tests for lights that contribute weakly [Ward, 1994], clipping polygonal area lights [Hart et al., 1999], using a visibility oracle based on a photon map [Jensen and Christensen, 1995] or learning during rendering [Fernandez et al., 2002]. Wald and Benthin [2003] cull lights based on a path tracing prepass. Random skipping of visibility tests [Billen et al., 2013] or their caching [Popov et al., 2013] have been likewise explored.

Reducing the cost of a single light evaluation cannot reduce the linear complexity of direct illumination computation, which becomes a bottleneck when there is a large number of lights. Paquette et al. [1998] and Walter et al. [2005] propose to hierarchically cluster lights into a tree and then use adaptively constructed tree cuts to approximate direct illumination. Both methods scale well but this comes at the expense of some bias. As a follow-up, methods by Walter et al. [2006] and Bus et al. [2015] further reduce the number of scene shading points for which the direct illumination computation is carried out by additionally clustering the shading points.

We address random light selection in a MC renderer. In this context, Shirley et al. [1996] pioneered the idea of designing light selection probabilities based on expected lights' contributions, though they only used a rather crude classification into 'important' and 'unimportant' lights. Wang and Akerlund [2009] sample lights proportionally to the product of a contribution estimate and surface reflectance. The method handles many lights by clustering, an idea we use in our work and extend it with online optimization of sampling distributions. Sampling distributions can also be obtained in a preprocess [Georgiev et al., 2012a; Wu and Chuang, 2013], but this approach disqualifies any form of interactive rendering. Finally, Donikian et al. [2006] learn a sampling distribution from samples obtained during the rendering, just as we do. The method combines several distributions in an ad hoc manner, which limits its robustness and reliability, as we demonstrate in our results. We show that a theoretically funded Bayesian treatment of adaptive sampling yields substantial improvements in robustness and overall efficiency.

Bayesian modelling in rendering Bayesian modelling is a widespread methodology in computer vision and graphics, so we only review works closely related to MC rendering. Boughida and Boubekeur [2017] use Non-Local Bayes image denoising [Lebrun et al., 2013] in the context of MC simulation as a post-processing filter. Brouillat et al. [2009] and Marques et al. [2013] pioneered the use of Bayesian Monte Carlo (BMC) [Rasmussen and Ghahramani, 2002] in light transport simulation. The BMC methodology models the posterior probability of an integral given a set of integrand estimates and a prior distribution over the integral outcome. While theoretically sound, it comes with some important computational overhead. In contrast, we keep the efficient classic, frequentist MC approach and apply Bayesian modelling to optimize our sampling distributions.

This approach was also taken by Vorba et al. [2014], who employ a maximum a posteriori (MAP) formulation to regularize training of parametric mixture models for optimized indirect illumination sampling. Our work uses a MAP formulation of spatial regression so as to obtain robust direct illumination estimates across the scene.

Adaptive sampling Literature on adaptive sampling in both general MC [Kalos and Whitlock, 2008] and in rendering is wide and we only mention some more recent work. One impactful theoretical idea has been population Monte Carlo (PMC) [Cappé et al., 2004], which can, among other, be used to optimize sampling distributions represented by mixture models [Douc and Guillin, 2007; Cappé et al., 2008]. Adaptive multiple importance sampling (AMIS) [Cornuet et al., 2009] extends the adaptation idea to multiple importance sampling [Veach, 1997], whereas adaptive population importance sampling (APIS) [Martino et al., 2015] attempts to exploit the strong points of PMC or AMIS. PMC has been applied in rendering [Lai et al., 2007; Fan et al., 2007], but the benefits are not large. Our work differs from PMC by the lack of any resampling step which would require storing individual samples.

Path guiding Methods that build models of incoming illumination specific to one particular scene and use them for importance sampling have become known as *path guiding*. These methods perform either density estimation from particles obtained in a preprocessing step [Jensen, 1995; Hey and Purgathofer, 2002; Budge et al., 2008; Vorba et al., 2014] or they derive the importance density through regression modelling [Lafortune and Willems, 1995; Pegoraro et al., 2008; Müller et al., 2017]. Our method is orthogonal to guiding methods since it addresses sampling of *direct* illumination. In fact, it could be incorporated into existing guiding approaches based on regression, as we discuss in Section 1.7.

1.2 Overview

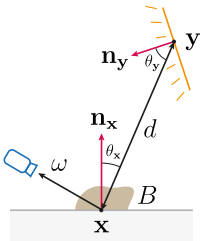
Direct illumination estimator Our goal is to compute the reflected radiance L due to direct illumination at a shading point $\mathbf{x}$ as seen from a direction ω. It is defined as an integral over all points $\mathbf{y}$ on the surface A of all scene light sources

$$L(\mathbf{x},\omega) = \int_A F(\mathbf{y}\to\mathbf{x}\to\omega)\,\mathrm{d}\mathbf{y}, \tag{1.1}$$

where the integrand equals

$$F(\mathbf{y}\to\mathbf{x}\to\omega) = L_\mathrm{e}(\mathbf{y}\to\mathbf{x})B(\mathbf{y}\to\mathbf{x}\to\omega)V(\mathbf{y}\leftrightarrow\mathbf{x})G(\mathbf{y}\leftrightarrow\mathbf{x}). \tag{1.2}$$

Here, $L_\mathrm{e}(\mathbf{y}\to\mathbf{x})$ is the radiance emitted from $\mathbf{y}$ towards $\mathbf{x}$, $B(\mathbf{y}\to\mathbf{x}\to\omega)$ is the BRDF describing the surface reflectance at $\mathbf{x}$, and $V(\mathbf{y}\leftrightarrow\mathbf{x})$ is the binary visibility function returning 1 if $\mathbf{y}$ is visible from $\mathbf{x}$ and 0 otherwise. The geometry factor $G(\mathbf{y}\leftrightarrow\mathbf{x})$ equals to $\frac{\cos\theta_\mathbf{y}\cos\theta_\mathbf{x}}{d^2(\mathbf{y},\mathbf{x})}$, where $\cos\theta_\mathbf{y} = \mathbf{n}_\mathbf{y}\cdot\frac{\mathbf{x}-\mathbf{y}}{d(\mathbf{y},\mathbf{x})}$, $\cos\theta_\mathbf{x} = \mathbf{n}_\mathbf{x}\cdot\frac{\mathbf{y}-\mathbf{x}}{d(\mathbf{y},\mathbf{x})}$ with $\mathbf{n}_\mathbf{y}$, $\mathbf{n}_\mathbf{x}$ being the unit surface normal at $\mathbf{y}$ and $\mathbf{x}$, respectively, and $d(\mathbf{y},\mathbf{x})$ is the Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$.



A Monte Carlo estimator for the integral (1.1) is given by

$$\langle L(\mathbf{x}, \omega) \rangle = \frac{F(\mathbf{y} \to \mathbf{x} \to \omega)}{p(\mathbf{y}|\mathbf{x}, \omega)}, \tag{1.3}$$

where $p(\mathbf{y}|\mathbf{x}, \omega)$ denotes the pdf of sampling the light point $\mathbf{y}$ from the shading point $\mathbf{x}$ given the viewing direction ω. The better the pdf approximates the integrand, the lower the variance, with the pdf directly proportional to the integrand yielding zero variance.

Light sampling. We seek a practical approximation to the ideal pdf described above. We follow a standard approach for generating a light sample $\mathbf{y}$, where one first selects a light source, and then samples a point on that light [Pharr et al., 2016]. To ensure good scalability with many lights, we additionally employ adaptive light clustering: each point $\mathbf{x}$ in the scene has an associated set $\mathcal{C}$ of light clusters c. In this setup, sampling the light point $\mathbf{y}$ in the estimator (1.3) breaks down into the following three steps:

1. Select a light cluster $c \in \mathcal{C}$ with the probability $P(c|\mathbf{x})$,[1]
2. Select a light $l \in c$ with the probability $P(l|c)$ proportional to its flux, i.e. $P(l|c) = \Phi_l / \sum_{l' \in c} \Phi_{l'}$,
3. Select a point $\mathbf{y} \in l$ with the pdf $p(\mathbf{y}|l, \omega)$ using standard techniques [Shirley et al., 1996; Pharr et al., 2016; Gamito, 2016].

The resulting pdf $p(\mathbf{y}|\mathbf{x}, \omega)$ is then obtained as $P(c|\mathbf{x})\, P(l|c)\, p(\mathbf{y}|l, \omega)$.

Adaptive cluster sampling. The main contribution of this chapter consists in a new adaptive method for constructing the cluster sampling distribution $P(c|\mathbf{x})$ used in Step 1. To this end, we first derive, in Section 1.3, the optimal distribution for cluster selection in presence of *variance due to nested MC estimation*, i.e. illumination evaluation within each cluster corresponding to Steps 2 and 3. Second, we devise a Bayesian methodology to learn such a distribution in a progressive manner (Section 1.4). For that purpose, we design a statistical MAP regression model of cluster contribution and visibility. The model is initialized by conservative cluster contribution estimates, which embody our prior knowledge. It is then updated on the fly during rendering using the calculated (observed) light contributions.

We do not use learning for sampling the point $\mathbf{y}$ on an individual light in Step 3, since techniques tailored to different kinds of light geometries provide close-to-optimal solutions [Shirley et al., 1996; Pharr et al., 2016; Gamito, 2016]. Furthermore, we design our cluster sampling distributions to be view independent: we omit the BRDF factor and we drop the dependency on the view direction ω in most equations. This is motivated by practical considerations of a production renderer, where reflectance can be defined by arbitrarily complex shaders, often given as a black-box. We discuss the above decisions in Section 1.7.

[1] Probabilities are denoted by the capital P while probability *densities* are lower-case p.

Light clustering and scene partitioning. Our light clustering approach is inspired by Lightcuts [Walter et al., 2005]. Similar to Wang and Akerlund [2009], we use the clusters for light selection, as opposed to using them directly as illumination estimates. As a result, the clustering affects the estimator variance, not a systematic image error, and hence it can be rather coarse.

In a preprocessing step, we first hierarchically cluster the lights into a binary *light tree* in a similar way to Lightcuts. The light tree is constructed in a bottom-up manner, starting with each light as one cluster and then repeatedly merging a pair of clusters with the lowest value of a metric described in Appendix 1.9.1. This is the only preprocessing step of our method, it is done once for a scene and its computational cost is negligible (especially in comparison to loading a regular production scene). Therefore, it does not limit interactive progressive rendering.

During rendering, the light tree serves for finding light clusterings $\mathcal{C}$, represented as a cut in the light tree. Unlike in the original Lightcuts algorithm, where lights are clustered for each shading point on-the-fly, we generate and cache light clusterings for entire scene regions. Such persistent clusterings are necessary to keep the statistics for updating the cluster sampling distributions. The scene is therefore divided into disjoint spatial regions, and each region has an associated light clustering, represented as a light cut. The light cut for a scene region is created on demand, the first time direct illumination calculation is carried out in that region. This saves a lot of computations since only a small percentage of all regions is usually used.

As in Lighcuts, the cut construction starts at the root and repeatedly replaces the cluster with the highest estimated contribution by its two children, until the estimated cluster contribution falls below ϵ-fraction of the estimated contribution of the entire cut (we use $\epsilon = 0.1$ and limit the cut size to 100 in all our results). Calculation of the cluster contribution estimates is described in Appendix 1.9.1. In scenes with a moderate light count, the clusters usually correspond to the individual lights, and our adaptive algorithm then samples the lights themselves. Figure 1.2 shows an example of the light clustering and scene partitioning.

Baseline scalable method. An algorithm based on the above light clustering, where cluster sampling probability $P(c|\mathbf{x})$ is proportional to the cluster contribution estimates (Appendix 1.9.1) and is *not* adapted during calculation, serves as a baseline for comparisons in Section 1.6. We call it the *Scalable* method.

1.3 What we learn: Optimal cluster selection

We now discuss the optimal cluster selection probabilities $P(c|\mathbf{x})$ in Step 1 of our three-step light sampling procedure (Section 1.2). The conventional way to shape $P(c|\mathbf{x})$ would be to select cluster c proportionally to its true expected contribution, denoted $L_c(\mathbf{x})$. However, as we show below, this choice would be optimal only if the cluster contributions could be evaluated with no variance. This is rarely the case in practice, since the *nested MC estimator* $\langle L_c(\mathbf{x}) \rangle$ of the cluster contribution is itself subject to additional variance (due to sampling of light areas and complex visibility). Intuitively, one would want to sample more frequently clusters that contribute more variance to the overall result, but the

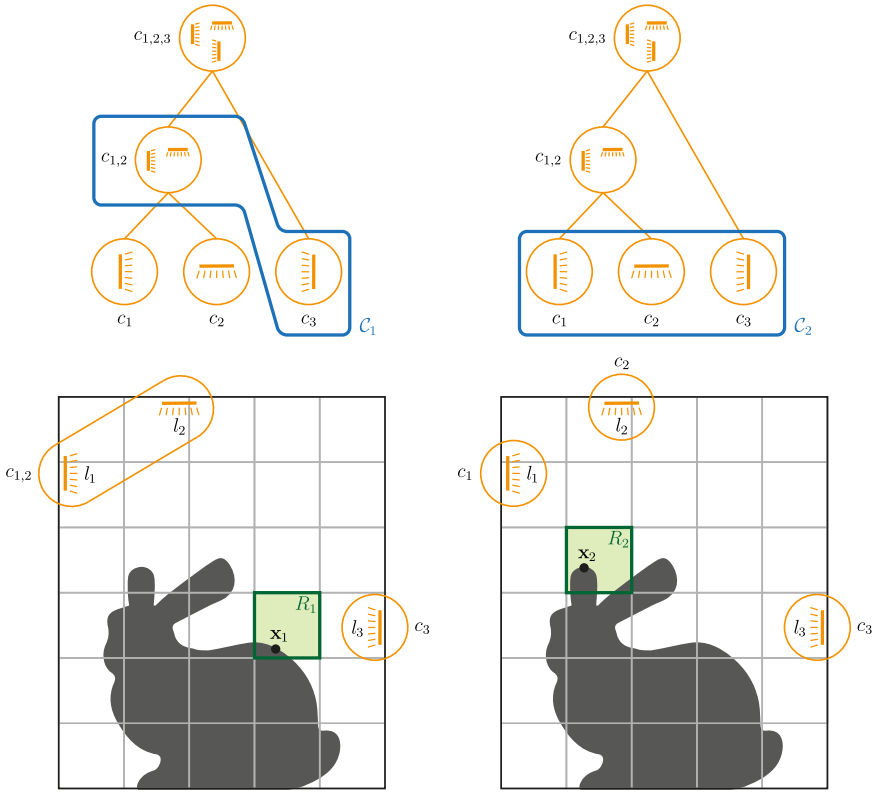


Figure 1.2: An example of clusterings of lights in a simple scene for two different scene regions (bottom row) and the respective cuts in the light tree (upper row). In the left example, contributions of lights l_1 and l_2 to region R_1 are weak and similar. Therefore, the descend in the light tree during the construction of cut $\mathcal{C}_1$ stops at cluster $c_{1,2}$ and c_3. As a result, any shading point $\mathbf{x}_1$ inside region R_1 will sample two clusters: one with lights l_1 and l_2, and one with light l_3. On the other hand, contributions of lights l_1 and l_2 to region R_2 in the right example are strong and different. Therefore, the cut construction does not stop until reaching the bottom of the light tree and any shading point $\mathbf{x}_2$ inside region R_2 will then sample the individual lights.

simple selection proportional to the contribution does not do this (Figure 1.3). We now derive the optimal cluster selection probabilities conforming to this intuition.

We seek optimal cluster sampling probabilities $P_{\text{opt}}(c|\mathbf{x})$ minimizing the overall variance of estimator (1.3). Given our three-step sampling, we have $p(\mathbf{y}|\mathbf{x}) = P(c|\mathbf{x})P(l|c)p(\mathbf{y}|l)$, and the variance can be written as:

$$\text{Var}[\langle L(\mathbf{x})\rangle] = -L(\mathbf{x})^2 + \sum_{c\in\mathcal{C}} \frac{1}{P(c|\mathbf{x})} \underbrace{\int_{A_c} \frac{(F(\mathbf{y}\to\mathbf{x}))^2}{P(l|c)p(\mathbf{y}|l)}\,\mathrm{d}y}_{m_{2,c}}. \tag{1.4}$$

Note that $m_{2,c}$ is the second moment of the *nested MC estimator* $\langle L_c(\mathbf{x})\rangle = \frac{F(\mathbf{y}\to\mathbf{x})}{P(l|c)p(\mathbf{y}|l)}$ of the cluster contribution.

We find $P_{\text{opt}}(c|\mathbf{x})$ as a solution to a constrained optimization problem, in which we minimize the variance (1.4) with respect to the cluster sampling probabilities $P(c|\mathbf{x})$, subject to $\sum_{c\in\mathcal{C}} P(c|\mathbf{x}) = 1$. Let us denote $w_c = P(c|\mathbf{x}), c \in \mathcal{C}$, where $\mathcal{C}$ is the set of clusters. We further define $\mathbf{w} = (w_{c_1}, \ldots, w_{c_{|\mathcal{C}|}})$ and $m_{2,c}$ as in (1.4).

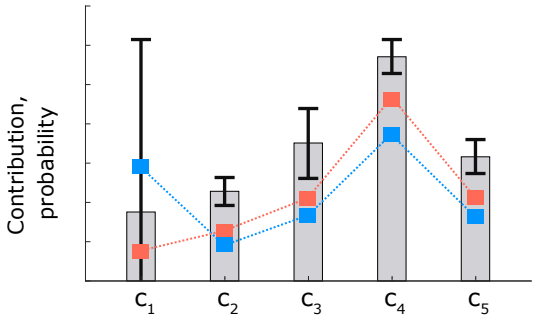


Figure 1.3: An illustration of optimal sampling probabilities on a synthetic dataset. The gray bars represent the expected cluster contributions and the error bars show standard deviation of the nested cluster contribution estimators. The orange distribution shows the conventional cluster selection probabilities directly proportional to the cluster contributions, while the blue one correspond to our provably optimal sampling probabilities promoting sampling of clusters that contribute more variance.

Next, we set up a lagrangian $\mathbb{L}(\mathbf{w}, \lambda)$

$$\mathbb{L}(\mathbf{w}, \lambda) = -L(\mathbf{x})^2 + \left(\sum_{c \in \mathcal{C}} \frac{1}{w_c} m_{2,c}\right) + \lambda \left(\sum_{c \in \mathcal{C}} w_c - 1\right), \tag{1.5}$$

where $\lambda \in \mathbb{R}$ and we seek a solution $\mathbf{w}, \lambda$ of the equation $\nabla \mathbb{L}|_{\mathbf{w},\lambda} = \mathbf{0}$, yielding the following set of equations:

$$\begin{aligned} \frac{d}{dw_c}\mathbb{L}(\mathbf{w}, \lambda) &= -\frac{1}{w_c^2} m_{2,c} + \lambda = 0, \\ \frac{d}{d\lambda}\mathbb{L}(\mathbf{w}, \lambda) &= \sum_{c \in \mathcal{C}} w_c - 1 = 0. \end{aligned} \tag{1.6}$$

The solution is $w_c = \sqrt{\frac{1}{\lambda} m_{2,c}}$ and $\lambda = \left(\sum_{c \in \mathcal{C}} \sqrt{m_{2,c}}\right)^2$, where λ serves as a normalization factor making the w_c sum up to one. In other words, the *optimal cluster sampling probability* $P_{opt}(c|\mathbf{x})$ *is proportional to the square root of the second moment* $m_{2,c}$. Given that $\mathrm{Var}[\langle L_c(\mathbf{x})\rangle] = m_{2,c} - L_c^2(\mathbf{x})$, we obtain the final result:

$$\boxed{P_{\mathrm{opt}}(c|\mathbf{x}) \propto \sqrt{L_c^2(\mathbf{x}) + \mathrm{Var}[\langle L_c(\mathbf{x})\rangle]}.} \tag{1.7}$$

Note that $P_{\mathrm{opt}}(c|\mathbf{x})$ is not proportional just to $L_c(\mathbf{x})$, but it also takes the variance of the nested estimator into account, i.e., variance due to sampling of light areas and complex visibility. This is crucial for the robustness of our method as it prevents excessive noise by focusing on problematic areas in the cases when the nested sampling according to the pdf $P(l|c)p(\mathbf{y}|l)$ is far from ideal (see Figure 1.6).

A derivation similar to ours appears in the work by Pantaleoni and Heitz [2017], but in a different context: seeking an optimal piecewise constant approximation to a given sampling probability density.

1.4 How we learn: Bayesian online regression

In the previous section we have shown that optimal cluster selection probability $P(c|\mathbf{x})$, given by (1.7), depends both on the expected cluster contribution $L_c(\mathbf{x})$ and the variance of the nested cluster contribution estimator $\mathrm{Var}[\langle L_c(\mathbf{x})\rangle]$. These quantities are, however, unknown up front, and have to be approximated.

We have two types of information available for that: a) Unbiased, but noisy MC direct illumination samples taken during rendering. b) Noise-free, but biased, estimates of unoccluded cluster contribution (see Appendix 1.9.1). Both are useful, but insufficient by themselves: The MC samples converge to the exact solution, but are extremely unreliable in early stages of computation. The contribution estimates are more reliable early on, but they do not get any more accurate over time and provide no information on visibility or the nested estimator's variance. A principled approach to *exploiting such uncertain information* and *fusing different sources of information* for adaptive MC sampling is the primary contribution of this chapter.

Intuitively, we understand the contribution estimates as our prior knowledge and the MC samples as observations. This view naturally leads to Bayesian modelling. While MC quadrature has traditionally served as a tool for Bayesian inference [Bishop, 2006], *we employ Bayesian inference as a tool for robust adaptive MC sampling.* The general idea of the Bayesian approach is to create a probability model describing the *likelihood* (occurrence probability density) of observed data, impose some *prior* probability over parameters of that model and then, infer the *posterior* probability of the model parameters after seeing the data. From the posterior, we can determine the quantity of interest. In our case, by modelling the likelihood of the MC samples and constructing the prior distribution using the contribution estimates, we can find the most probable approximations to $L_c(\mathbf{x})$ and $\mathrm{Var}[\langle L_c(\mathbf{x})\rangle]$ given both these sources of information.

1.4.1 Model

We start with a standard statistical learning setup. First, we define our training data $\mathcal{D}$ based on the MC samples observed during rendering. Second, we derive a model $p(\mathcal{D}|\theta)$ describing the likelihood of the data given parameters θ. Mean and variance of this model provide the $L_c(\mathbf{x})$ and $\mathrm{Var}[\langle L_c(\mathbf{x})\rangle]$ we are looking for.

These statistics depend on the parameters θ that are initially unknown. We could find them by direct maximization of $p(\mathcal{D}|\theta)$, i.e., use the *maximum likelihood* (ML) estimate. However, ML is prone to overfitting when data is scarce and provides poor approximations in early stages of rendering as shown in Figure 1.6. Since robustness is a major concern in adaptive MC, we employ the Bayesian treatment: Impose *prior* probability $p(\theta)$, and infer the *posterior* probability $p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta)$ after seeing the data. By its maximization we get a robust *maximum a posteriori* (MAP) estimate of the parameters.

Data. Each scene region is associated with a set of light clusters (the light cut). We collect the data and learn the model *independently for each region–cluster pair.* Consider one such pair. Sampling of lights in the cluster yields MC illumination samples, $\langle L_c(\mathbf{x})\rangle = \frac{F(\mathbf{y}\to\mathbf{x})}{P(l|c)p(\mathbf{y}|l)}$, where $\mathbf{y}$ is a sampled point on light l,

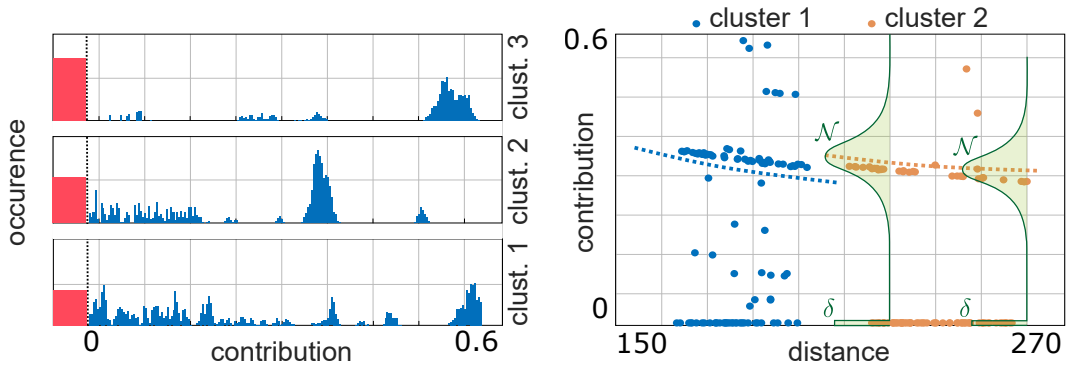


Figure 1.4: **Left:** A histogram of direct illumination samples for three region-cluster pairs. The area of each column corresponds to the overall occurrence in the dataset. Note that zeros (in red) are frequent due to complex occlusion. **Right:** A scatter plot of sample contribution $\hat{e}_{\mathbf{x}}$ vs sample distance $\hat{d}$ for two clusters distinguished by colours. Note the inverse-squared-distance falloff. The green overlay shows our regression model (1.10) for two different distances.

and $\mathbf{x}$ is a shading point inside the region. Our goal is to use the MC samples collected for the region-cluster pair to build a model that accurately predicts $L_c(\mathbf{x})$ and $\mathrm{Var}[\langle L_c(\mathbf{x})\rangle]$ over the different positions $\mathbf{x}$ in the region.

A major cause of spatial variations of illumination is the cosine term $\cos\theta_{\mathbf{x}}$ changing due to varying surface normal. While this effect would be difficult to capture by statistical modelling, it is trivial to compute when needed, so we drop it from our model. We therefore define two quantities

$$\hat{e} = \frac{L_{\mathrm{e}}(\mathbf{y}\to\mathbf{x})V(\mathbf{y}\leftrightarrow\mathbf{x})\cos\theta_{\mathbf{y}}/d^2(\mathbf{x},\mathbf{y})}{P(l|c)\,p(\mathbf{y}|l)} \quad \text{and} \quad \hat{e}_{\mathbf{x}} = \hat{e}\,\overline{\cos}\theta_{\mathbf{x}}. \tag{1.8}$$

The former quantity, $\hat{e}$, represents the MC sample of the cluster contribution, $\langle L_c(\mathbf{x})\rangle = \frac{F(\mathbf{y}\to\mathbf{x})}{P(l|c)p(\mathbf{y}|l)}$, with the surface cosine term $\cos\theta_{\mathbf{x}}$ dropped. In the latter quantity, $\hat{e}_{\mathbf{x}}$, we replace the cosine term by its upper bound over the entire cluster $\overline{\cos}\theta_{\mathbf{x}}$. Our region statistics are based on $\hat{e}$, while $\hat{e}_{\mathbf{x}}$ is used at a specific shading point $\mathbf{x}$ to inject surface normal dependency into our model.

After the surface normal, the second important factor in illumination variation across a region is the inverse-squared falloff with the distance $\hat{d} = d(\mathbf{x},\mathbf{y})$, as confirmed by the empirical data (Figure 1.4). To capture this dependency, we explicitly model the relation between illumination samples and the distance $\hat{d}$ using a regression model. Therefore, our training data $\mathcal{D}$ consists of tuples $(\hat{e}_{\mathbf{x},i}, \hat{d}_i)$.

Model and its parameters. The next step is to define a statistical regression model $p(\mathcal{D}|\theta)$ expressing the data likelihood, i.e., *probability of MC samples of direct illumination*. The general form of the likelihood used to model the relation between $\hat{d}$ and $\hat{e}_{\mathbf{x}}$ is:

$$p(\mathcal{D}|\theta) = \prod_i^N p(\hat{e}_{\mathbf{x},i}|\hat{d}_i,\theta). \tag{1.9}$$

where $p(\hat{e}_{\mathbf{x}}|\hat{d},\theta)$ represents a regression model, N is the total number of samples (for a region-cluster pair), and the model parameters θ are discussed below.

Our regression model of direct illumination has the following two important features:

1. Approximation of the inverse-squared-distance falloff.
2. Explicit modelling of occluded contributions.

A motivation for using the first feature was given above, and follows naturally from the form of the sample contribution $\hat{e}_{\mathbf{x}}$ (1.8). The second feature arises from the all-or-nothing nature of the visibility function, which is difficult to model by any common smooth distribution (Figure 1.4). We, therefore, design our regression model as a mixture of a delta function δ (describing zero, i.e., occluded contributions) and a Gaussian $\mathcal{N}$ with mean and variance decreasing with the second and fourth power of the distance term (describing non-zero, i.e., visible contributions):

$$p(\hat{e}_{\mathbf{x}}|\hat{d},\theta) = \delta(\hat{e}_{\mathbf{x}})p_o + (1-p_o)\mathcal{N}\left(\hat{e}_{\mathbf{x}} \;\middle|\; \frac{k}{\hat{d}^2}, \frac{h}{\hat{d}^4}\right). \tag{1.10}$$

See Figure 1.4 for an illustration. The model parameters $\theta = (p_o, k, h)$ are respectively the probability of occlusion, mean visible contribution coming from a cluster omitting the distance, and the variance of this contribution. As each sample $\hat{e}_{\mathbf{x},i}$ shows inverse-squared-distance falloff of its mean, sample's variance changes as well, but with $1/\hat{d}^4$. The benefit of approximation of the inverse-squared-distance falloff and explicit visibility modelling is illustrated in Figure 1.6.

Prior distribution To make the inference step tractable, we seek a *conjugate prior*, i.e., prior distribution which yields a posterior of the same function type. The conjugate prior for our model, derived in Appendix 1.9.2, has p_o distributed according to the beta distribution B and the pair (k, h) according to the normal-inverse-gamma distribution $\mathcal{N}$-Γ^{-1}. Our prior distribution for parameters θ is then:

$$p(\theta) = \mathrm{B}(p_o|\hat{N}_o, \hat{N}_v)\ \mathcal{N}\text{-}\Gamma^{-1}(k, h \;\big|\; \mu_0, \hat{N}, \hat{N}_\alpha, \beta). \tag{1.11}$$

The various hyperparameters in the above equation can be understood as statistics of hypothetical prior observations before the first actual sample has been taken. $\hat{N}_o$ and $\hat{N}_v$ denote the number of occluded and visible prior observations, μ_0 is the mean of $\hat{N}$ prior visible observations and β is the sum of squares of $2\hat{N}_\alpha$ prior visible observations. Note that these hyperparameters do not necessarily describe a consistent set of virtual prior observations (i.e., in general $\hat{N} \neq \hat{N}_v$ and $2\hat{N}_\alpha \neq \hat{N}$). Intuitively, $\hat{N}_o$, $\hat{N}_v$, $\hat{N}$ and $\hat{N}_\alpha$ express the strength of the priors and larger values will cause slower, but potentially more robust learning.

To obtain the hyperparameter μ_0, we use our *second source of information*, the unoccluded cluster contribution estimate $\tilde{L}_c(\mathbf{x})$ (Appendix 1.9.1). To make the prior more robust to occasional gross errors in these estimates, we blend the $\tilde{L}_c(\mathbf{x})$-proportional distribution with a defensive uniform distribution over the clusters. Finally, $\tilde{L}_c(\mathbf{x})$ contains a division by the squared-distance $d^2(\mathrm{ctr}(c), \mathbf{x})$ to the cluster center $\mathrm{ctr}(c)$. But μ_0 is a prior on the parameter k, which gets divided by the distance in our model (1.10). We counter double division by the

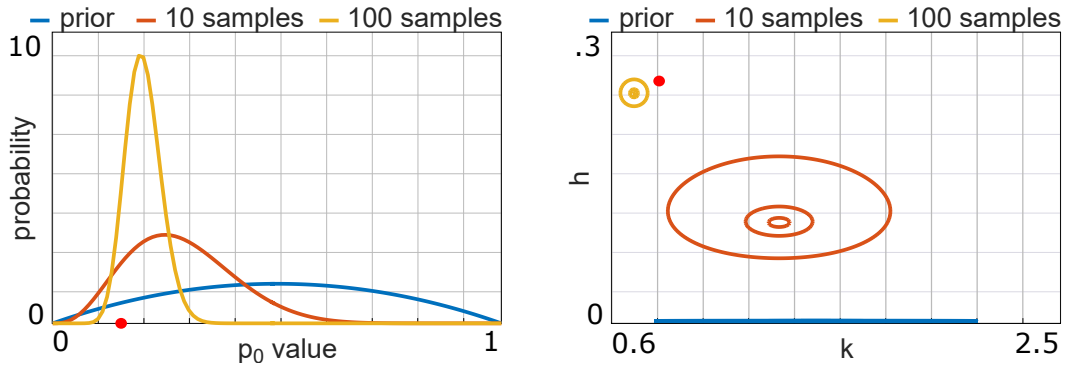


Figure 1.5: Evolution of the posterior distribution for the parameters p_o (beta distribution, left), and k and h (normal-inverse-gamma, right) after seeing 0 (prior), 10 and 100 (synthetic) samples, with hyperparameters set as described in the main text, and the hyperparameter μ_0 set to 1.5. The samples' true occlusion probability was 0.15, the visible samples' true mean and variance was 0.8 and 0.25, respectively, and are marked by the red dot. Note that the $\mathcal{N}$-Γ^{-1} prior is zero almost everywhere except near the x-axis, due to $\beta = 1\text{e}{-6}$.

distance by pre-multiplying by $d^2(\text{ctr}(c), \mathbf{x})$. In summary, our informed prior mean reads

$$\mu_0 = \frac{1}{2}\left(\tilde{L}_c(\mathbf{x}) + \frac{\sum_{c' \in \mathcal{C}} \tilde{L}_{c'}(\mathbf{x})}{|\mathcal{C}|}\right) d^2(\text{ctr}(c), \mathbf{x}). \tag{1.12}$$

Good values of the other hyperparameters should strike a good trade-off between the learning rate and robustness to noisy samples. We found the following values to work robustly across all our tests: $\hat{N}_o = 2, \hat{N}_v = 2, \hat{N} = 1, \hat{N}_\alpha = 1, \beta = 1\text{e}{-6}$. Refer to Section 1.6 for a discussion of this choice.

Figure 1.5 shows how increasing number of observed samples shapes the posterior distribution of parameters θ from this prior distribution towards the true value of the parameters.

1.4.2 Inference

With both the likelihood and prior defined, we now infer the most probable parameter values after seeing the data. We maximize the logarithm of the posterior distribution with respect to the parameters to obtain the MAP point estimate for θ. That boils down to finding the solution to $\nabla_\theta \log(\, p(\mathcal{D}|\theta)p(\theta)\,) = 0$, which expands to:

$$\left(\sum_i^N \frac{\nabla_\theta p(\hat{e}_{\mathbf{x},i}|\hat{d}_i, \theta)}{p(\hat{e}_{\mathbf{x},i}|\hat{d}_i, \theta)}\right) + \frac{\nabla_\theta p(\theta)}{p(\theta)} = 0. \tag{1.13}$$

Plugging our model equations (1.10) and (1.11) into (1.13) we get the following system of equations:

$$\begin{aligned}
\frac{(\hat{N}_o - 1)(1-p_o) - p_o(\hat{N}_v - 1)}{p_o(1-p_o)} - \frac{N_o}{p_o} - \frac{N_v}{1-p_o} &= 0 \\
\frac{s_{1,\mathbf{x}} - k(\hat{N} + N_v) + \hat{N}\mu_0}{h} &= 0 \\
\frac{-2ks_{1,\mathbf{x}} + s_{2,\mathbf{x}} - 2\hat{N}_\alpha h + 2\beta + \hat{N}(\mu_0 - k)^2 + N_v(k^2 - h) + h}{h} &= 0.
\end{aligned}$$

The solution to this system gives us the MAP estimate of the θ parameters:

$$p_o = \frac{-1 + \hat{N}_o + N_o}{-2 + \hat{N}_o + \hat{N}_v + N}, \tag{1.14}$$

$$k = \frac{s_{1,\mathbf{x}} + \hat{N}\mu_0}{\hat{N} + N_v}, \tag{1.15}$$

$$h = \frac{-2\hat{N}\mu_0 s_{1,\mathbf{x}} - s_{1,\mathbf{x}}^2 + (s_{2,\mathbf{x}} + 2\beta)(\hat{N} + N_v) + \hat{N}N_v\mu_0^2}{(2\hat{N}_\alpha + N_v - 1)(\hat{N} + N_v)} \tag{1.16}$$

where $s_1 = \sum_i^{N_v} \hat{d}_i^2 \hat{e}_i$, $s_{1,\mathbf{x}} = s_1 \overline{\cos}\theta_{\mathbf{x}}$ and $s_2 = \sum_i^{N_v} \hat{d}_i^4 \hat{e}_i^2$, $s_{2,\mathbf{x}} = s_2 \overline{\cos}\theta_{\mathbf{x}}^2$ represent statistics over *visible* samples, N_o and N_v are the number of occluded and visible samples, and $N = N_o + N_v$ is the overall number of samples (for the considered region-cluster pair).

With these parameters, the expectation and variance of our model in (1.10), approximating $L_c(\mathbf{x})$ and $\mathrm{Var}[\langle L_c(\mathbf{x})\rangle]$, respectively, are:

$$L_c(\mathbf{x}) \approx (1-p_o)k/\hat{d}^2, \tag{1.17}$$

$$\mathrm{Var}[\langle L_c(\mathbf{x})\rangle] \approx (1-p_o)(p_o k^2 + h)/\hat{d}^4. \tag{1.18}$$

We set $\hat{d} = d(\mathrm{ctr}(c), \mathbf{x})$ to approximate the not yet known distance for $\mathbf{x}$, where $\mathrm{ctr}(c)$ denotes the cluster center.

1.4.3 Summary

Let us now summarize the steps involved in direct illumination computation at a shading point $\mathbf{x}$. We take the cut $\mathcal{C}$ stored in region R containing $\mathbf{x}$ and for each of its clusters c we compute the unoccluded contribution estimates $\tilde{L}_c(\mathbf{x})$ and $\overline{\cos}\theta_{\mathbf{x}}$ (Appendix 1.9.1), and we set $\hat{d} = d(\mathrm{ctr}(c), \mathbf{x})$. We cull clusters with $\tilde{L}_c(\mathbf{x}) = 0$, i.e., which have provably zero contribution to $\mathbf{x}$, from any further processing.

For the remaining clusters, we compute μ_0 using (1.12), retrieve the region-cluster statistics s_1, s_2, N_o, N_v, and compute the MAP parameters (p_o, k, h) using (1.14), (1.15) and (1.16). Finally, we get the sampling probability $P^*(c|\mathbf{x})$ by plugging (1.17) and (1.18) into (1.7):

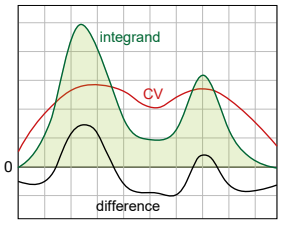
$$P^*(c|\mathbf{x}) \propto \frac{1}{\hat{d}^2}\sqrt{(1-p_o)^2k^2 + (1-p_o)(p_o k^2 + h)}. \tag{1.19}$$

Using these probabilities we select a cluster c^*, then we select a light $l^* \in c^*$ with probability $P(l^*|c^*) = \Phi_{l^*} / \sum_{l' \in c^*} \Phi_{l'}$ and finally, sample a point $\mathbf{y}^* \in A_{l^*}$ using the standard techniques [Shirley et al., 1996; Pharr et al., 2016; Gamito, 2016]. Contribution of this sample is then used to update statistics s_1, s_2, N_o, N_v stored for the cluster c^* in the region R.

1.5 Control variates

Inspired by the successes of control variates documented in previous work [Owen and Zhou, 2000; Clarberg and Akenine-Möller, 2008; Pegoraro et al., 2008; Rousselle et al., 2016], we exploit our accumulated statistics as a control variate for further variance reduction.

The idea of control variates is to subtract a function – a control variate (CV) – with a known expected value from the integrand, to estimate the integral of the difference using MC, and then add the expected value of the CV. Intuitively, the closer the CV is to the integrand, the smaller the difference is and so is the variance of the MC estimation. See Section 2.5.1 in the next chapter for more formal introduction into control variates.



We keep the nested MC estimator $\langle L_c(\mathbf{x}) \rangle$ of cluster contribution as before (i.e. Steps 2 and 3 in Section 1.2), and apply the CV to the MC estimator of the sum over clusters:

$$\langle L(\mathbf{x}) \rangle_{\mathrm{CV}} = \frac{\langle L_c(\mathbf{x}) \rangle - H(c, \mathbf{x})}{P(c|\mathbf{x})} + \sum_{c' \in \mathcal{C}} H(c', \mathbf{x}). \tag{1.20}$$

The better the control variate $H(c, \mathbf{x})$ approximates the true cluster contribution $L_c(\mathbf{x})$, the more the variance is reduced. Since this is precisely the purpose of our Bayesian model (see (1.17)), it would seem natural to also use it directly as the CV. However, while we strongly prefer overestimation to underestimation for the sampling distribution, this is not the case for the CV. We, therefore, omit the conservative prior in its definition, and the CV reads

$$H(c, \mathbf{x}) = \frac{1}{N} \frac{s_{1,\mathbf{x}}}{d^2(\mathrm{ctr}(c), \mathbf{x})}. \tag{1.21}$$

Despite the CV acting as a mere empirical improvement over the theory presented so far, it yields noticeable variance reduction at a negligible cost (Figure 1.6).

1.6 Results

Implementation We implemented our method in the path tracer of the Corona renderer and deployed it among users. Our path tracer combines light sampling and BRDF importance sampling using MIS [Veach, 1997] alleviating the fact that our sampling distributions do not take BRDF into account (as we discussed in Section 1.2). When used in this setting, the direct illumination samples $\hat{e}_i$, which we use for training, are pre-multiplied by MIS weights. This heuristic approach works well in practice (see Figure 1.13), and a more principled analysis is left for future work.

Test setup We show results of our tests on three different scenes: Living room, City and Door (see Figure 1.1 and 1.10). Living room is a typical scene in the architectural visualization featuring a living room lit by the sun and a few area lights on the ceiling. In contrast, the City scene shows a street at night and contains more than 5000 light sources. Finally, Door is a rather simple scene featuring complex shadowing.

In addition to these three main scenes, we use two another scenes for specific comparisons: Wedge and Hall (see Figure 1.9 and 1.13). Wedge is a simple synthetic scene illuminated by three area lights and an environment map. Hall features complex glossy materials illuminated by the sun, an environment map and tens of area lights of various sizes.

Exact light counts along with other statistics are summarized in Table 1.1. All scenes were rendered at the resolution 1080×720 on a single machine with the Intel Core i7-5820K CPU (6 cores, 12 threads) and 32 GB of RAM.

Method components We first demonstrate the effect of the individual components of our method in the City scene in Figure 1.6. We start by sampling proportionally to an estimate of each light's unoccluded contribution (a). At every shading point, this method estimates the contribution of *all* scene lights (using $\tilde{L}_c(\mathbf{x})$ from Appendix 1.9.1), and uses these estimates to construct the sampling distribution. This procedure becomes prohibitively expensive for the many lights as in this scene.

By subdividing the scene into regions and sampling proportionally to the unoccluded contribution of light clusters in the associated cuts, we obtain the

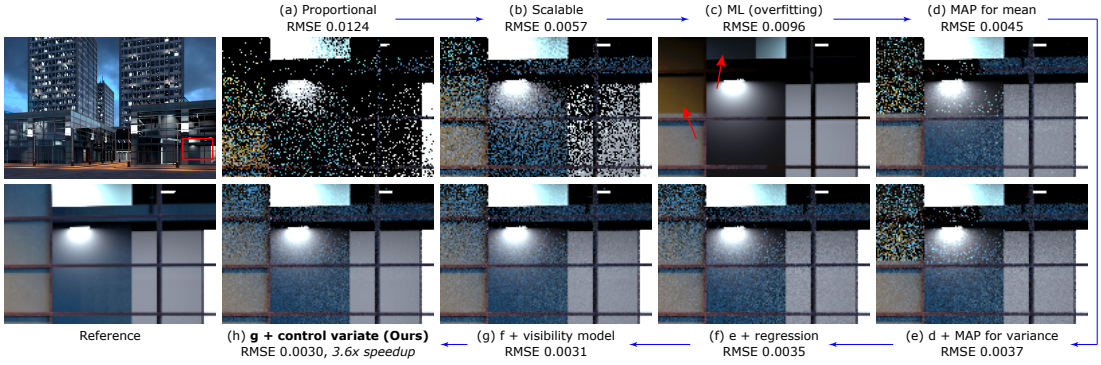


Figure 1.6: An equal-time comparison (60 s) of different components of our direct illumination sampling method in a scene with more than 5000 lights and high occlusion. We compare sampling proportional to (a) unoccluded light contribution computed separately for each shading point and light, (b) unoccluded light cluster contribution incorporating our scalable solution, (c) maximum likelihood (ML) estimate of the mean cluster contribution (dark artefacts are a consequence of overfitting), (d) maximum a posteriori (MAP) estimate of the mean cluster contribution. The remaining variants gradually add the following components: (e) MAP estimate for variance, (f) regression to model the distance falloff, (g) explicit modelling of occluded samples, (h) control variate. The last result corresponds to our final solution. The numbers below the method names denote the RMSE to a reference solution. The speedup is with respect to (b).

Scalable method (b) which scales much better with the number of lights but still neglects visibility.

Learning light sampling probabilities using a simple maximum likelihood (ML) estimate, i.e., the mean of MC samples, (c) can easily lead to bias: If the first observed sample is occluded (zero), the cluster will not receive any further samples, yielding dark artefacts highlighted by the red arrows in the figure.

Such artefacts can be avoided by using a MAP estimate of the mean (d). However, as we show in Section 1.3, optimal cluster sampling distribution should take into account the variance of sampling inside each cluster. Indeed, adding a MAP estimate for this nested estimator's variance significantly reduces noise (e). Incorporating regression modelling of the distance falloff (f) eliminates noise most noticeable near region boundaries. Finally, explicit modelling of occluded samples and the use of control variates further reduces noise. This is the complete method we use in all our further tests, and we denote it Ours. Version (b), denoted Scalable, serves as a baseline for the comparisons. In this scene, Ours is 3.6× faster than Scalable.

Robustness and DI-only performance We now demonstrate superior robustness of our method over the work by Donikian et al. [2006] (details of our reimplementation are given in Appendix 1.9.3). While Donikian et al.'s method also relies on learning, it is based on heuristics that eventually fail to deliver a robust solution. The method gathers statistics in image space and cannot be easily integrated in a global illumination solution. For this reason, we compare on direct illumination (DI), and take this opportunity to provide a DI-only comparison to the Scalable method, see Figure 1.7.

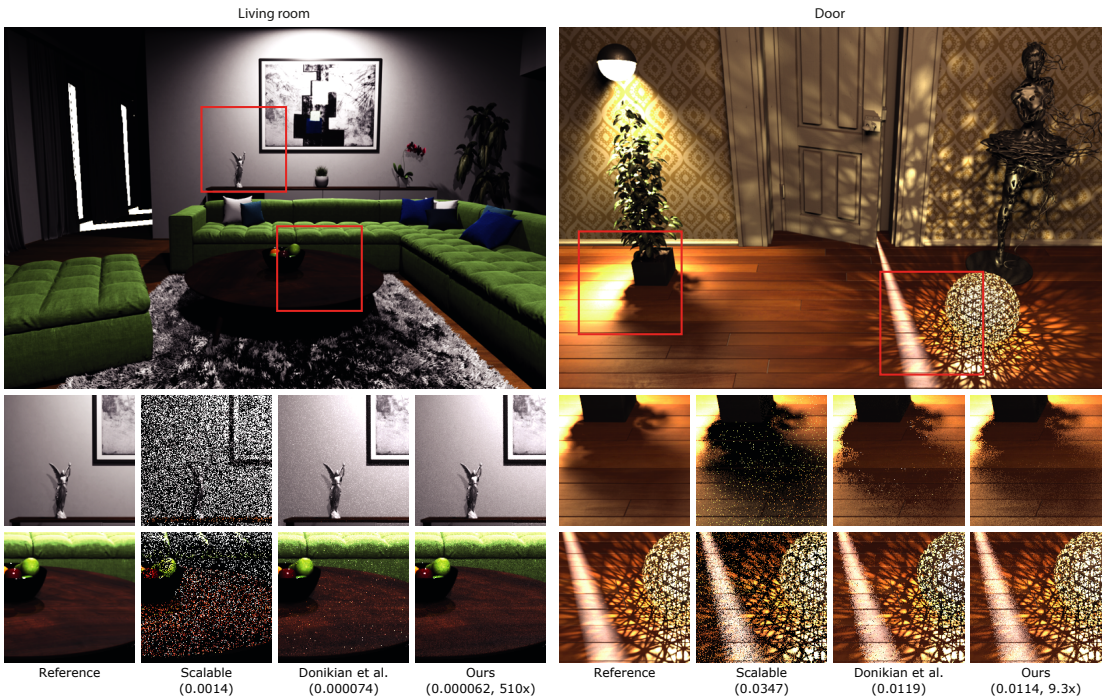


Figure 1.7: Equal-time comparison (60 s) of our method against Scalable and Donikian et al.'s methods in a direct illumination setting. See the main text for details.

The sun in the Living room scene is significantly stronger than other lights. Since the Scalable method has no notion of visibility, it prefers sampling the sun while undersampling the other lights, even in sun's shadow. Our method quickly learns the sun occlusion and avoids the excessive noise of Scalable. It converges more evenly and more than 500× faster. Donikian et al.'s method also shows improvement over Scalable but struggles with sampling a ceiling area light covered by a shade letting only a small portion of the light through. The method overfits and introduces spiky noise.

The Door scene aims at testing robustness with complex shadow and light patterns. While Scalable struggles in shadows as before, Donikian et al.'s method learns light occlusion quickly and it may even outperform our method in uniformly lit areas. However, this aggressive adaptation comes at the cost of overfitting, which is then manifested as spiky noise and artefacts around shadow boundaries. Notice the square holes in the penumbra of the plant in the first inset and at intersections of the net of shadows in the second one. Our method robustly handles all these situations while being more than 9× faster than Scalable.

We compared Scalable and our method in the City scene (Figure 1.6) but we had to omit Donikian et al.'s because of its vague description of dealing with many lights (we lack information of how to accumulate block statistics coming from possibly very distant shading points and thus having different light clusterings). RMSE evolution plots in Figure 1.8 show that in the City scene our method maintains a stable speedup over Scalable, while in the other two we can observe a higher empirical convergence rate.

We want to underline that our improvement over Donikian et al. lies mainly in the robustness, not the speed. In fact, their method can outperform ours in uniformly lit areas, but introduces unacceptable artefacts at shadow boundaries (Figure 1.7 and 1.9). This lack of robustness is an inherent property of their static strategy to prevent overfitting (weighting distributions based on the iteration step) and cannot be avoided by any parameter tweaking. Addressing this deficiency is the very purpose of our Bayesian approach.

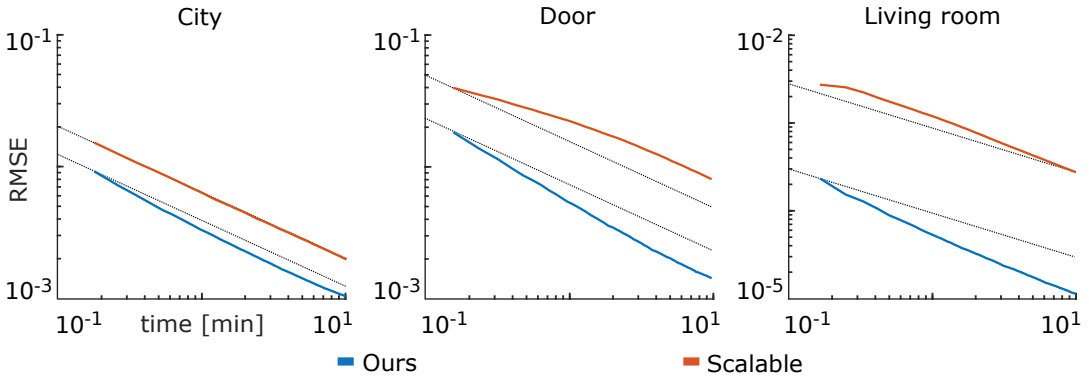


Figure 1.8: RMSE evolution (10 min) for the direct illumination only. Our method is compared against the Scalable method. The plots start at 10 seconds to ensure all pixels were sampled at least once.

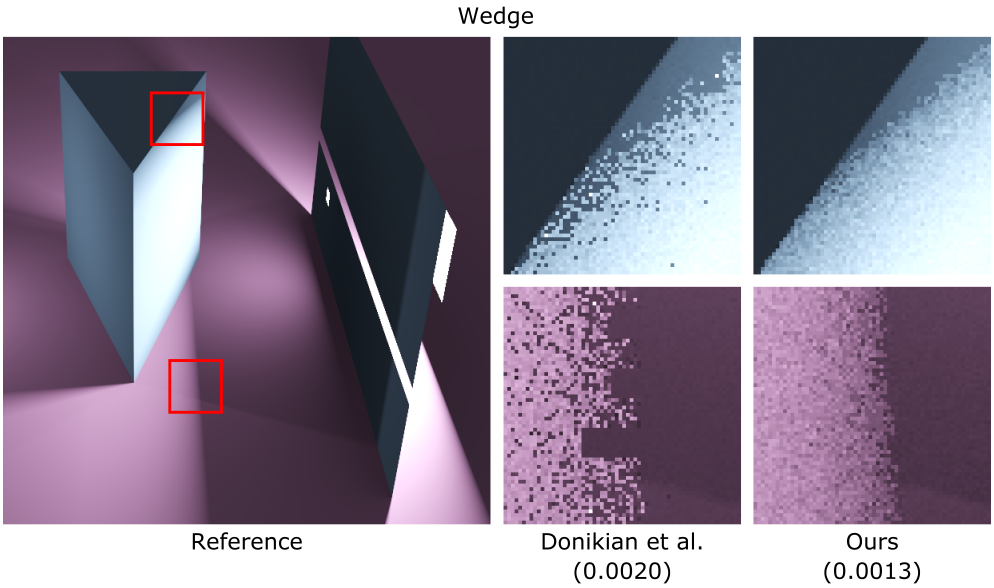


Figure 1.9: An equal-time time comparison (10 s) of our method against Donikian et al.'s method in a direct illumination setting. The large area light on the right that illuminates the scene only through a narrow gap presents a difficult situation for the Donikian et al.'s method. Many of its samples are blocked, which increases the danger of overfitting, manifested as block artefacts along shadow boundaries where the algorithm incorrectly decided to stop sampling the light.

Discussion of other competing work. The method of Wang and Akerlund [2009] is similar to the Scalable method. Unlike Wang and Akerlund, Scalable omits the BRDF from light sampling distribution, but that does not introduce any disadvantage on diffuse surfaces. Furthermore, Scalable achieves some performance gain by caching of light cuts for scene regions. As a result, comparison against the Scalable baseline can serve as a fairly good approximation to a comparison against Wang and Akerlund.

We do not compare against methods that involve substantial preprocessing [Georgiev et al., 2012a; Wu and Chuang, 2013] since these methods address a different use case than ours. In a typical commercial rendering workflow a vast majority of renders are in fact short tests, not the final images. In this context, a preprocessing step is an obstacle that would prevent the method from being used in the pipeline that our users rely on in their daily work.

Global illumination integration When integrated in a global illumination (GI) solution, the relative performance improvement of our method naturally depends on the variance contribution due to the direct and indirect components. While our DI method yields an almost noise-free GI result in all three scenes, in the City and Door scenes (Figure 1.10) roughly half of the speedup of the DI-only solution is retained (speedup 3.6× from Figure 1.6 and 9.3× from Figure 1.7 decreases to 2.0× and 4.3× respectively). On the other hand, our 510× speedup in the DI-only comparison in the Living room scene (Figure 1.7) reduces to 6.7× in GI (Figure 1.1). This indicates that variance contribution of the direct component in this scene is small in comparison to the total illumination.

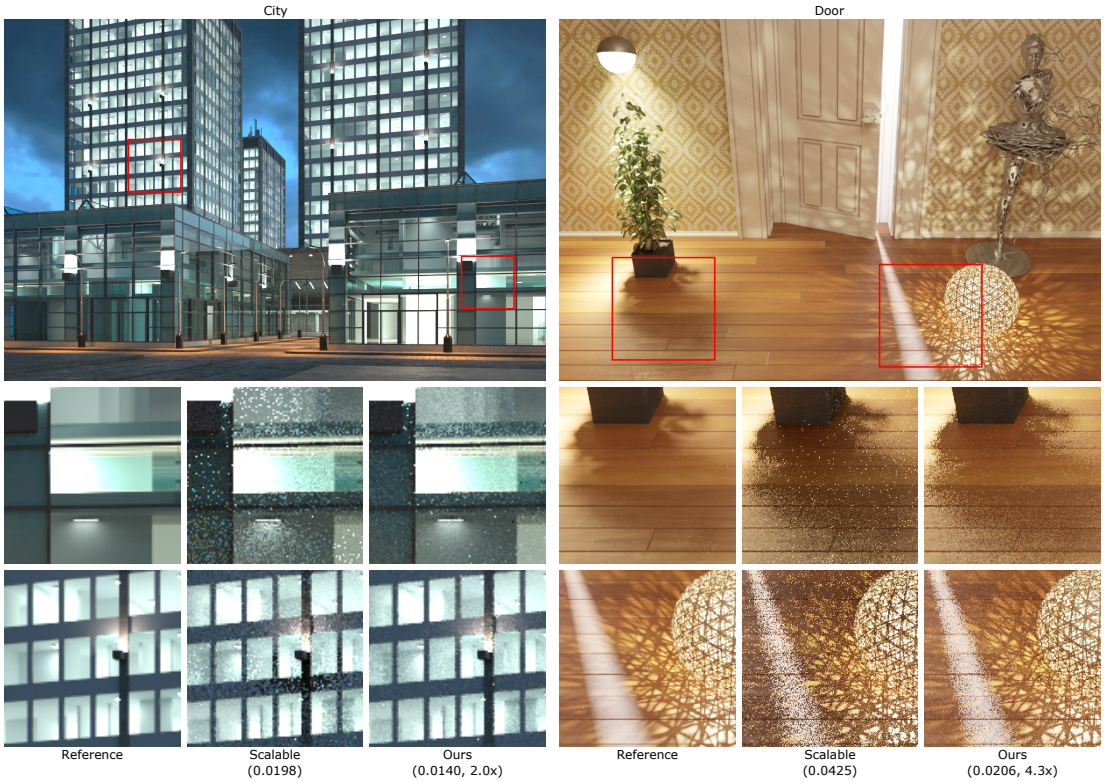


Figure 1.10: Equal-time time comparison (60 s) of our method against Scalable in a global illumination setting. See the main text for more details.

Grid resolution Our spatial regression model makes the performance of our algorithm rather insensitive to the division of scene into regions. As shown in Figure 1.11, a trade-off exists between the model accuracy (the smaller the regions, the more accurate the models) and the learning rate (the larger the regions the more samples are available) in the City scene (though the dependence is weak) while almost no difference is visible in the other scenes. For this reason, all our results use a fixed-resolution uniform grid with cubical regions with 64 regions along the shortest scene dimension.

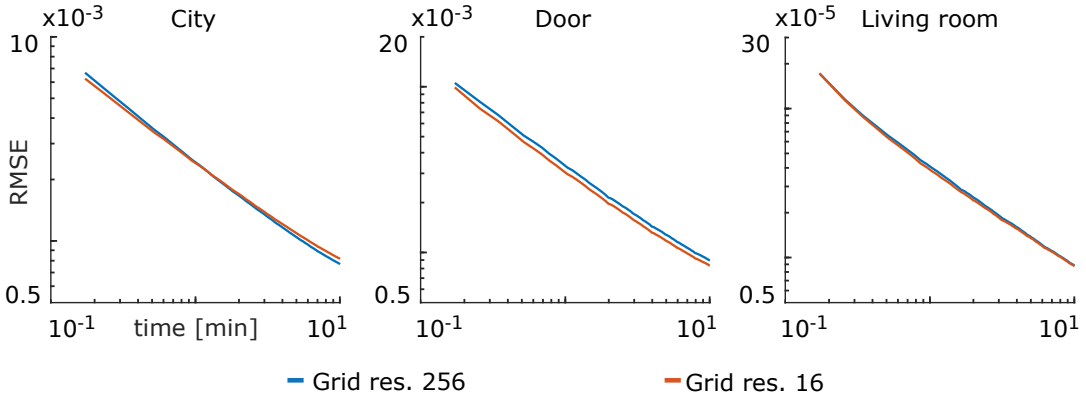


Figure 1.11: RMSE evolution (10 min) for different grid resolutions. With a finer resolution our model might learn more slowly but achieve better accuracy (and thus lower RMSE). Nonetheless, the differences are very small.

	Light count	Non-empty regions	Average cut size	Memory (MB)	Over-head
City	5022	39666 (4.1%)	33	101	7.2%
Door	5	24526 (1.1%)	5	97	3.6%
Living room	5	57304 (2.3%)	5	113	7.9%
Hall	78	31304 (6.6%)	39	78	9.8%
Wedge	4	10871 (0.4%)	4	101	9.0%

Table 1.1: Statistics gathered after 120 s of rendering of our test scenes with global illumination. The number in parentheses is the percentage of total region count (scene dependent). The average cut size (i.e., number of clusters per region) is taken over non-empty regions only. Total memory consumed by the regions and clusters is reported. The overhead expresses relative decrease of pixel samples per second with respect to Scalable.

Memory consumption and overhead At our grid resolution, memory consumed by the stored light cuts and model statistics is moderate, as we show in Table 1.1. These numbers are for a GI setting and less memory is consumed when computing only DI. An empty scene region occupies 40 B of memory. Every cluster inside a region consumes additional 48 B in order to store: 2× 64-bit double for statistics s_1, s_2; 2× 32-bit integer for statistics N_o, N_v; 64-bit pointer to cluster tree node; 32-bit integer for flags; 3× 32-bit float for RGB channels of s_1 for the control variate.

Regarding computation overhead, the number of pixel samples per second decreased in our method in comparison to Scalable by no more than 10% in all our test scenes (see Table 1.1). The learning compensates for this by better sampling, which yields a much improved overall result.

Unbiasedness Although we use past samples to update sampling distributions, we do not modify sample values based on the past observations and our method is therefore unbiased. In Figure 1.12 we empirically demonstrate a steady convergence of our method to the result of the (non-adaptive) Scalable method.

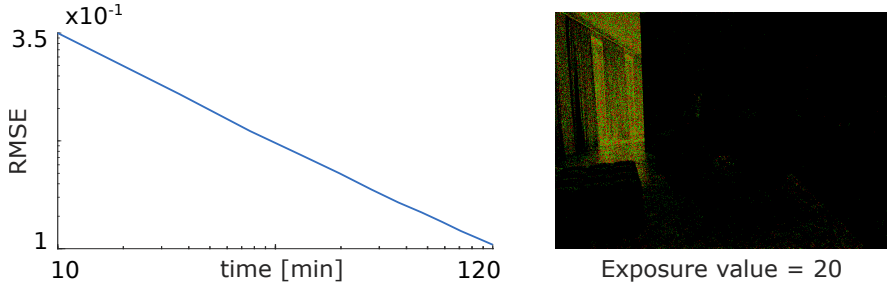


Figure 1.12: Steady convergence of our method (RMSE plot, left) to reference solution in the Living room scene suggests that our adaptive method accumulates no bias. A 2^{20}× amplified colour-coded difference image (right), taken at the end of the measurement, shows that any remaining differences are due to a random noise (red=positive and green=negative difference).

MIS combination We tested our method both with and without MIS combination with BRDF sampling. While there is almost no difference in the Living room, City and Door scenes, in scenes with large area lights and glossy materials, the MIS combination proves beneficial as shown in the Hall scene in Figure 1.13. Even in this scene containing complex illumination and glossy materials, our method performs well even though our light sampling distribution neither takes the BRDF into account, nor addresses sampling of individual lights.

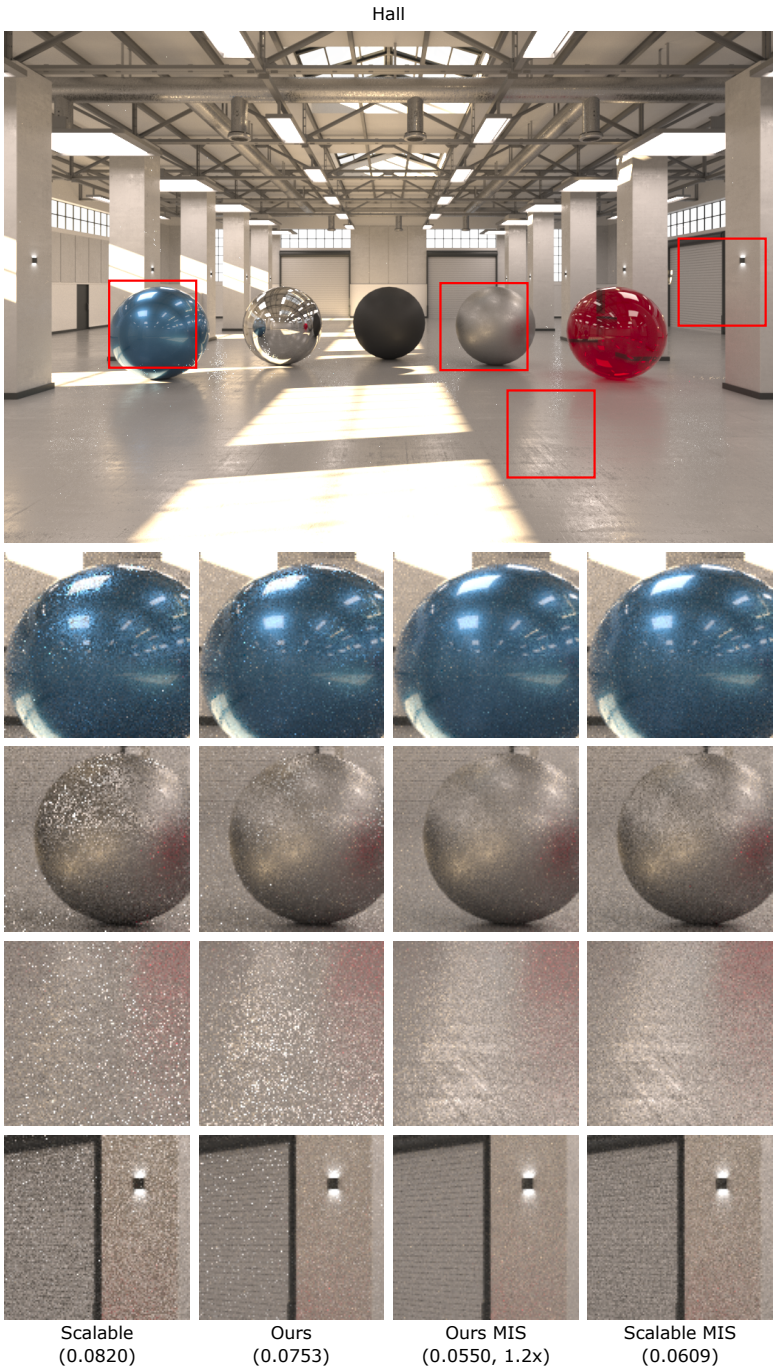


Figure 1.13: An equal-time time comparison (60 s) of our method against Scalable with and without MIS in a global illumination setting.

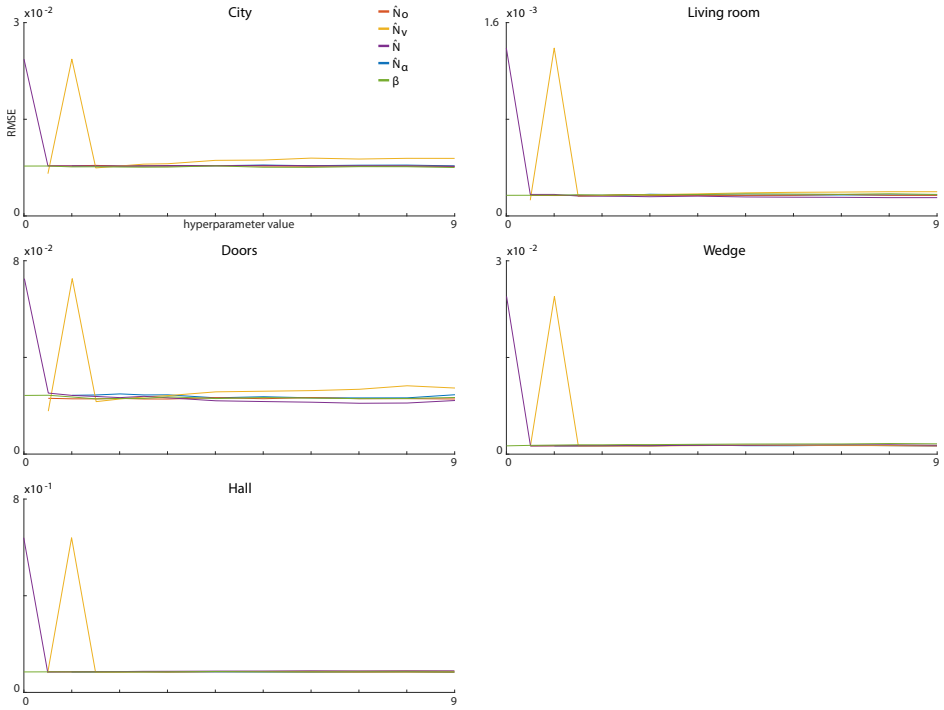


Figure 1.14: Plots of RMSE (after 10 s) with respect to different values of the hyperparameters in a direct illumination setting.

Hyperparameters Our default choice of the hyperparameter values yields an uninformed prior distribution over the model parameters, and works robustly across all our tests. In particular, we use $\hat{N}_o = 2, \hat{N}_v = 2, \hat{N} = 1, \hat{N}_\alpha = 1, \beta =$ 1e−6. We tried to individually vary each of these values but we did not see any significant change in the resulting image quality (see Figure 1.14). Only the setting $\hat{N}_v = 1$ or $\hat{N} = 0$ causes a sudden increase of image noise, since our method with these values essentially degenerates into the maximum likelihood solution.

Prior accuracy To better understand the importance of the prior of our model and its accuracy, we tested our method with a less precise prior. In particular, we replaced the upper bound $\overline{\cos}\theta_{\mathbf{x}}$ on the surface cosine in the $\tilde{L}_c(\mathbf{x})$ estimate (1.23) with a trivial bound of 1. This modification had only a minor effect in most of the scenes except in the City scene, where the trivial bound noticeably increased the image noise (see Figure 1.15). This observation is in line with our expectation that the prior is important but our method is not too sensitive to its exact value as it quickly learns the actual light contributions.

Clustering In Figure 1.16, we analyse the effect of light clustering on the performance of our method, in particular the effect of ϵ, the fraction of the estimated contribution of the entire cut, used as a threshold for stopping the cut refinement. With higher values the cuts are smaller and faster to compute, the maximum value of 1 would cluster all lights into a single cluster. With lower values the cuts are more accurate, the minimum value of 0 either clusters each light in its own cluster (less than 100 lights) or into a maximum cut of 100 clusters (more than 100 lights).

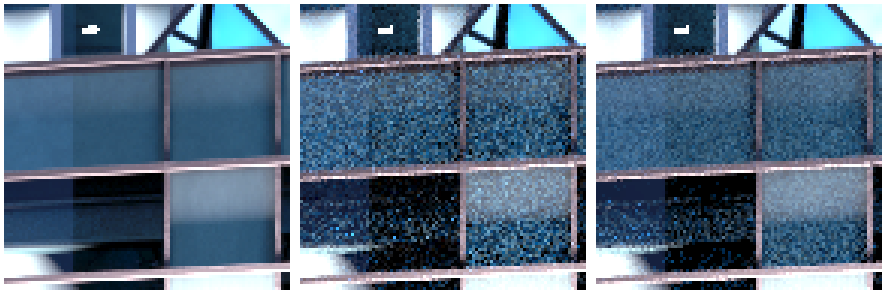


Figure 1.15: An equal-time time comparison (60 s) of using a trivial bound on the surface cosine for the model prior against using the upper bound $\overline{\cos}\theta_{\mathbf{x}}$ in a direct illumination setting.

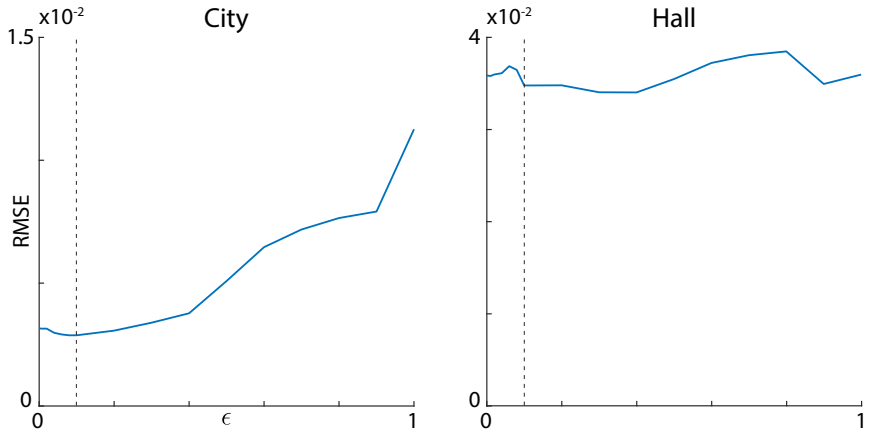


Figure 1.16: Plots of RMSE (after 60 s) with respect to the clustering precision ϵ in a direct illumination setting. The dashed line denotes $\epsilon = 0.1$, the value we used in all our tests.

As expected, the clustering has an important impact on the performance in the City scene, which contains more than 5000 lights (the optimum values yields more than 20× speedup in comparison with the least suitable value). On the other hand, the clustering has much smaller effect in the Hall scene with less than 100 lights (the speedup is only 1.3×). We used $\epsilon = 0.1$ in all our tests, which is optimal in the City scene and close to optimal in the Hall scene.

1.7 Limitations and future work

Multiple Importance Sampling (MIS) We have discussed in Section 1.6 the heuristic nature of the integration of our method with MIS. While our approach works well in practice and successfully handles large area light sources and complex materials (Figure 1.13), a more in-depth analysis could yield further improvements.

BRDF Our method does not consider the BRDF factor in learning the sampling distributions. This makes the learning more tractable (a less detailed function to learn) and practical in a production setting (the BRDF can be a black box). But

it limits the adaptability of the sampling. Though this has not been an issue in practice thanks to the MIS combination with BRDF sampling, incorporating the BRDF in the learning process could still be beneficial.

Scene subdivision Another interesting point is the trade-off between model accuracy and learning rate due to the scene division. The graphs in Figure 1.11 suggest such a trade-off exists, although the differences are small. However, the graphs show aggregate statistics over the entire scene, which can obscure the fact that adaptive scene subdivision could still have an important positive *local* impact.

Hyperparameters While we discussed different hyperparameter values in Section 1.6, we see a more rigorous approach for hyperparameter selection as yet another area of research. Our default choice yields an uninformed prior distribution over the parameters, which fits all scenes, but it might deliver suboptimal performance. Full Bayesian treatment (i.e., marginalizing the hyperparameters out) could yield further performance gains.

Sampling of individual lights Our method focuses on light selection and leaves sampling of the final point on the light unaddressed. This is motivated by the fact that the light selection is usually responsible for most of the variance in direct illumination. But this may not always be the case, especially when the individual lights are large (e.g., environment maps). This is partially alleviated by the integration with MIS (Figure 1.13) but there is certainly some potential for improvement.

Overhead Probably the thorniest practical issue, shared with the Scalable method, is the overhead associated with constructing the sampling distribution at each shading point. This is amortized in our implementation by a relatively large splitting factor (16 samples taken from one distribution) but it could be an issue in a simple path tracer without splitting.

Relation to path guiding As mentioned in Section 1.1, path guiding and our method share the idea of sampling according to a priori unknown illumination estimates. But while path guiding usually focuses on indirect illumination, we address specifically light source selection for direct illumination computation. In fact, our work is a component that could be integrated into a path guiding solution. We believe that any path guiding algorithm working in a forward manner from camera toward light could benefit from incorporating our approach and even implementation should be relatively straightforward as most of the algorithms already use some space partitioning schemes.

1.8 Conclusion

In this chapter, we presented our approach to decreasing the variance of MC integration in rendering by finding a better sampling technique. We focused on direct illumination calculation and proposed an unbiased adaptive direct illumination

algorithm with online learning of a light sampling distribution. The distribution is continually improved to better match the integrand based on the contribution of the direct illumination samples taken during rendering, including the visibility factor. As in any other adaptive MC sampling scheme, issues associated with limited reliability of the available information threaten the robustness of the resulting algorithm. As the main contribution of this chapter, we propose a Bayesian treatment of the learning process based on a statistical model developed specifically for the direct illumination sampling process. This treatment results in a robust and efficient algorithm, which has been successfully used in the Corona renderer to this day. We hope that the presented methodology will find its use in other adaptive MC schemes both in image synthesis and other application domains.

1.9 Appendix

1.9.1 Contribution estimates and clustering metric

Our scalable method differs from Lightcuts mainly in the way the cluster contribution estimates are calculated and in the clustering metric used when building the light tree. We use two kinds of estimates. First, $\tilde{L}_c(\mathbf{x})$ denotes an estimate of the contribution of cluster c to a particular shading point $\mathbf{x}$. It is used as a prior distribution in our Bayesian learning model. Second, since we construct one cut per entire scene region, the cut construction needs an estimate $\tilde{L}_c(R)$ valid for all points in the respective region R.

Cluster-to-point estimate. We first discuss the *point estimate* $\tilde{L}_c(\mathbf{x})$. Unlike Lightcuts, we do not desire an upper bound, since it often drastically overestimates the actual contribution. Instead, we use less conservative estimates, so that our prior better matches actual contributions. We seek to estimate the radiance due to direct illumination from cluster c:

$$L_c(\mathbf{x}) = \int_{A_c} \frac{L_\mathrm{e}(\mathbf{y}\to\mathbf{x})V(\mathbf{y}\leftrightarrow\mathbf{x})\cos\theta_\mathbf{y}\cos\theta_\mathbf{x}}{d^2(\mathbf{y},\mathbf{x})}\,\mathrm{d}\mathbf{y}. \tag{1.22}$$

As in Lightcuts we use the same trivial bound for visibility $V = 1$, upper bound $\overline{\cos}\theta_\mathbf{x}$ on the cosine at surface and upper bound $\overline{\cos}\theta_c$ on the cosine at the light cluster. $\overline{\cos}\theta_\mathbf{x}$ is computed as the maximum cosine between the surface normal at $\mathbf{x}$ and the direction from $\mathbf{x}$ to any point inside the cluster's bounding box. $\overline{\cos}\theta_c$ is computed as the maximum cosine between any normal in the cluster's normal cone and the direction from any point inside the cluster's bounding box to $\mathbf{x}$. See Figure 1.17 for an illustration and the Lightcuts publication [Walter et al., 2005] for computation details.

Unlike Lightcuts, we use $\overline{\cos}\theta_c$ only if the cluster center is further than 1.5 times the cluster diameter. For nearby clusters this bound would become too conservative and yield poor priors, so we average it with the cosine at the cluster center $\mathrm{ctr}(c)$, i.e., the cosine between the direction $\mathbf{x} - \mathrm{ctr}(c)$ and the axis of the cluster's normal cone. We denote the resulting cosine estimate as $\overline{\cos}\theta'_c$. For the distance factor, we use a distance to the cluster center $d(\mathrm{ctr}(c), \mathbf{x})$. And finally for each light $l \in c$ we conservatively estimate radiance L_e it can emit to $\mathbf{x}$ and

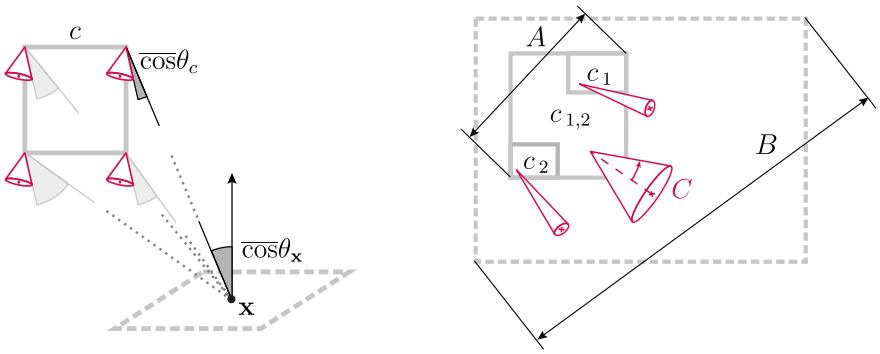


Figure 1.17: **Left:** An example of upper bound $\overline{\cos}\theta_{\mathbf{x}}$ on the cosine at surface and upper bound $\overline{\cos}\theta_c$ on the cosine at a cluster. Red cones represent the cluster's normal cone, i.e., the bounding cone of all light normals in the cluster. **Right:** The creation of the parent cluster $c_{1,2}$ from children c_1 and c_2. The bounding cone of the contained normals encompasses both child bounding cones.

denote it $\overline{L}_{\mathrm{e},l}$. For instance, for cosine lights with emission defined as $I_0(\cos\theta_{\mathbf{y}})^\alpha$ this estimate can be obtained as $I_0(\overline{\cos}\theta'_c)^\alpha$. Together we have:

$$\tilde{L}_c(\mathbf{x}) = \frac{\overline{\cos}\theta'_c \overline{\cos}\theta_{\mathbf{x}}}{d^2(\mathrm{ctr}(c), \mathbf{x})} \sum_{l \in c} |A_l| \overline{L}_{\mathrm{e},l}. \tag{1.23}$$

See Section 1.6 for a discussion of importance of $\tilde{L}_c(\mathbf{x})$ accuracy.

Cluster-to-region estimate. On the other hand, the *region-wide estimate* $\tilde{L}_c(R)$ is more conservative so as to produce better cuts (it is less prone to a premature stop of the cut construction because of underestimating parent clusters). We construct it as an upper bound of $\tilde{L}_c(\mathbf{x})$ over all points in region R by bounding its individual factors. A trivial bound is used for the cosine at surface since the surface normal in the region may be arbitrary. To bound the cluster cosine with respect to the entire region, we enlarge the cluster bounding box by the region box [Walter et al., 2006] and denote this bound as $\overline{\cos}\theta_c^R$. The distance between the cluster center and the region is bounded from below and denoted as $\underline{d}(\mathrm{ctr}(c), R)$. Finally, emitted radiance is bounded using maximum radiance a cluster light can contribute to any point in the region (similarly as in $\tilde{L}_c(\mathbf{x})$ but using the region-wide bound on the cluster cosine, i.e., $I_0(\overline{\cos}\theta_c^R)^\alpha$ for cosine lights). We denote it $\overline{L}_{\mathrm{e},l}^R$. Together we have:

$$\tilde{L}_c(R) = \frac{\overline{\cos}\theta_c^R}{\underline{d}^2(\mathrm{ctr}(c), R)} \sum_{l \in c} |A_l| \overline{L}_{\mathrm{e},l}^R. \tag{1.24}$$

Clustering metric. The light tree is constructed in a bottom-up manner, starting with each light as one cluster and then repeatedly merging a pair of clusters with the lowest value of the metric d_{tree} which expresses similarity of two clusters. For any two disjoint light clusters c_1, c_2 it is defined as

$$d_{\mathrm{tree}}(c_1, c_2) = (\Phi_{c_1} + \Phi_{c_2})(A^2 + B^2(1 - \cos C)^2) \tag{1.25}$$

where A is the length of the diagonal in the bounding box of the two clusters and C is the half-angle of the bounding cone of their normals. The relative weight of

the spatial and directional similarity is controlled by B which is set to length of the diagonal in the scene bounding box. See Figure 1.17 for an illustration. Φ_c is an approximation of the flux of the cluster c computed as

$$\Phi_c = \sum_{l \in c} \Phi_l, \quad \Phi_l = \int_{A_l} \max_{\omega} L_{\mathrm{e}}(\mathbf{y} \to \omega) \, \mathrm{d}\mathbf{y} \tag{1.26}$$

where l is a light inside cluster c, A_l is its surface and $\max_\omega L_{\mathrm{e}}(\mathbf{y} \to \omega)$ is its maximum radiance emitted from the point $\mathbf{y}$ to any direction ω on a sphere (e.g., $\Phi_l = |A_l| I_0$ for cosine lights).

1.9.2 Conjugate priors for our model

Setting $p(\theta) = p(p_o)p(k,h)$ in the relation $p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta)$, the posterior $p(\theta|\mathcal{D})$ will be proportional to:

$$p(p_o)p(k,h) \left(\prod_i^{N_o} \delta(\hat{e}_{\mathbf{x},i}) p_o \right) \left(\prod_i^{N_v} (1-p_o) \mathcal{N}\left(\hat{e}_{\mathbf{x},i} \;\middle|\; \frac{k}{\hat{d}_i^2}, \frac{h}{\hat{d}_i^4} \right) \right). \tag{1.27}$$

Beta prior To get the posterior distribution of p_o, we need to divide the above expression (1.27) by the marginal distribution $p(\mathcal{D}, k, h)$ which we get by integrating out p_o from (1.27). By doing so we get the posterior in the form:

$$p(p_o|\mathcal{D}, k, h) = K \; p(p_o) \; (1-p_o)^{N_v} p_o^{N_o}, \tag{1.28}$$

where K is some normalization factor depending only on the data $\mathcal{D}$ and our choice of the prior $p(p_o)$. We see that $(1-p_o)^{N_v} p_o^{N_o}$ is of the same form as the beta distribution. Therefore by setting $p(p_o) = \mathrm{B}(p_o|\hat{N}_o, \hat{N}_v)$ we are now able to evaluate K from (1.28) and we get the posterior distribution

$$p(p_o|\mathcal{D}, k, h) = \mathrm{B}(p_o|\hat{N}_o + N_o, \hat{N}_v + N_v). \tag{1.29}$$

We see that the beta distribution is indeed a conjugate prior of our model from (1.10).

Normal-inverse-gamma prior To find a conjugate prior for the k and h parameters, we proceed similarly as before with p_o. We get the posterior distribution of the form:

$$p(k,h|\mathcal{D}, p_o) = K \; \hat{d}_i^2 \; p(k,h) \prod_i^{N_v} \mathcal{N}\left(\hat{e}_{\mathbf{x},i} \hat{d}_i^2 | k, h \right), \tag{1.30}$$

where we used the relation $\mathcal{N}(\hat{e}_{\mathbf{x},i}|k/\hat{d}_i^2, h/\hat{d}_i^4) = \hat{d}_i^2 \mathcal{N}(\hat{e}_{\mathbf{x},i}\hat{d}_i^2|k,h)$ and K is again some normalization constant. The normal-inverse-gamma $\mathcal{N}$-Γ^{-1} distribution is a conjugate prior for such a case [Bishop, 2006]. Therefore it is a conjugate prior for our model (1.10).

1.9.3 Our implementation of Donikian et al. [2006]

Donikian et al. [2006] divide the image into blocks and process them one by one. For each block, they first fix one shading point for each pixel and then process the block pixels in iterations until convergence. In each iteration, they sample direct illumination at the shading points a fixed number of times (1.5× the light count), and subsequently update light contribution estimates at the block and pixel levels, respectively. The next iteration then uses a sampling distribution which mixes distributions at the block and pixel levels with the uniform distribution. The mixing weights are oblivious to the observed samples and depend solely on the iteration count. They change during the first 10 iterations only and remain fixed after that. This process is repeated until a convergence criterion is met for all pixels in the block; then a new block is started.

To make this method more compatible with ours, we made it progressive by computing all blocks at once. Furthermore, we find a new shading point for every pixel sample. One iteration then corresponds to taking one sample from all image pixels. The rendering time in our tests is set long enough for this method to complete enough iterations to learn (i.e., at least 10). Finally, we set our method in these tests to sample direct illumination at each shading point the same number of times (i.e., 1.5× the light count instead of the default 16×).

2. A better combination of techniques

In the previous chapter, we focused on finding a single sampling technique that would be a good match for the entire integrand. In this chapter, we investigate a different approach to decreasing the variance of MC integration: combining multiple sampling techniques, each of which could be a good match to a different feature of the integrand. In particular, we focus on *multiple importance sampling* (MIS), a robust way of combining sampling techniques proposed by Veach and Guibas [1995].

In the context of light transport simulation, MIS has served as a cornerstone for robust bidirectional path sampling [Veach and Guibas, 1995; Georgiev et al., 2012b; Hachisuka et al., 2012; Křivánek et al., 2014; Popov et al., 2015], Markov chain Monte Carlo light transport [Hachisuka et al., 2014; Šik et al., 2016; Gruson et al., 2016], adaptive path sampling (path guiding) [Vorba et al., 2014; Herholz et al., 2016; Müller et al., 2017], or in isolated integration problems such as direct illumination estimation [Veach and Guibas, 1995; Georgiev et al., 2012a]. Recall that we used MIS in the previous chapter in Section 1.6 to improve robustness of our direct illumination algorithm in the presence of glossy surfaces and large area light sources.

The key to the efficiency of MIS are the *weighting functions* used to combine samples from different sampling techniques. A set of weighting functions known as the balance heuristic has been suggested as a de facto universal solution, as no other weights can yield substantially lower variance [Veach and Guibas, 1995] (we show that this claim does not generally hold). Since the balance heuristic variance bounds can be fairly loose, alternative weights have been proposed to address shortcomings in some specific cases. The power, cutoff, or maximum heuristics can reduce variance for low-variance problems, but this comes at the expense of an overall variance increase [Veach and Guibas, 1995]. The α-max heuristic incorporates prior assumptions to avoid assigning too high weights to poorly performing sampling techniques [Georgiev et al., 2012a]. However, the performance of different weighting heuristics is problem-specific and the existing work fails to provide a clear answer as to which weighting functions to use in which situation.

Our work focuses on weighting functions for MIS. We derive a set of weighting functions that *provably minimize the variance of the MIS estimator* for a given set of sampling techniques and a fixed number of samples. The resulting optimal weights *may be negative*, and this additional flexibility enables substantial variance reduction over the existing weighting heuristics. In fact, we show that the optimal weights can result in *variance lower than the balance heuristic bounds* derived by Veach and Guibas [1995], as non-negativity of the weights was a silent assumption made in their derivation.

We provide further theoretical insights into the new optimal weights: we establish a connection between MIS with our optimal weights and another common variance reduction scheme – control variates. Specifically, we show the equivalence of the optimal weights to control variates applied to mixture sampling

[Owen and Zhou, 2000]. Moreover, we relate the variance of the optimal weights and the balance heuristic. The derivation of the optimal MIS weights and their analysis comprise the main theoretical contribution of this chapter.

The practical contribution of this chapter consists in proof-of-concept applications of the optimal weighting scheme in light transport, specifically in direct illumination calculation. Figure 2.1 demonstrates this application on a combination of two sampling techniques for light selection in a scene illuminated by two light sources. The first technique – *Trained* – was trained from samples to select lights based on their unoccluded contribution. It performs very well on surfaces illuminated by both lights or where one of the lights cannot contribute because

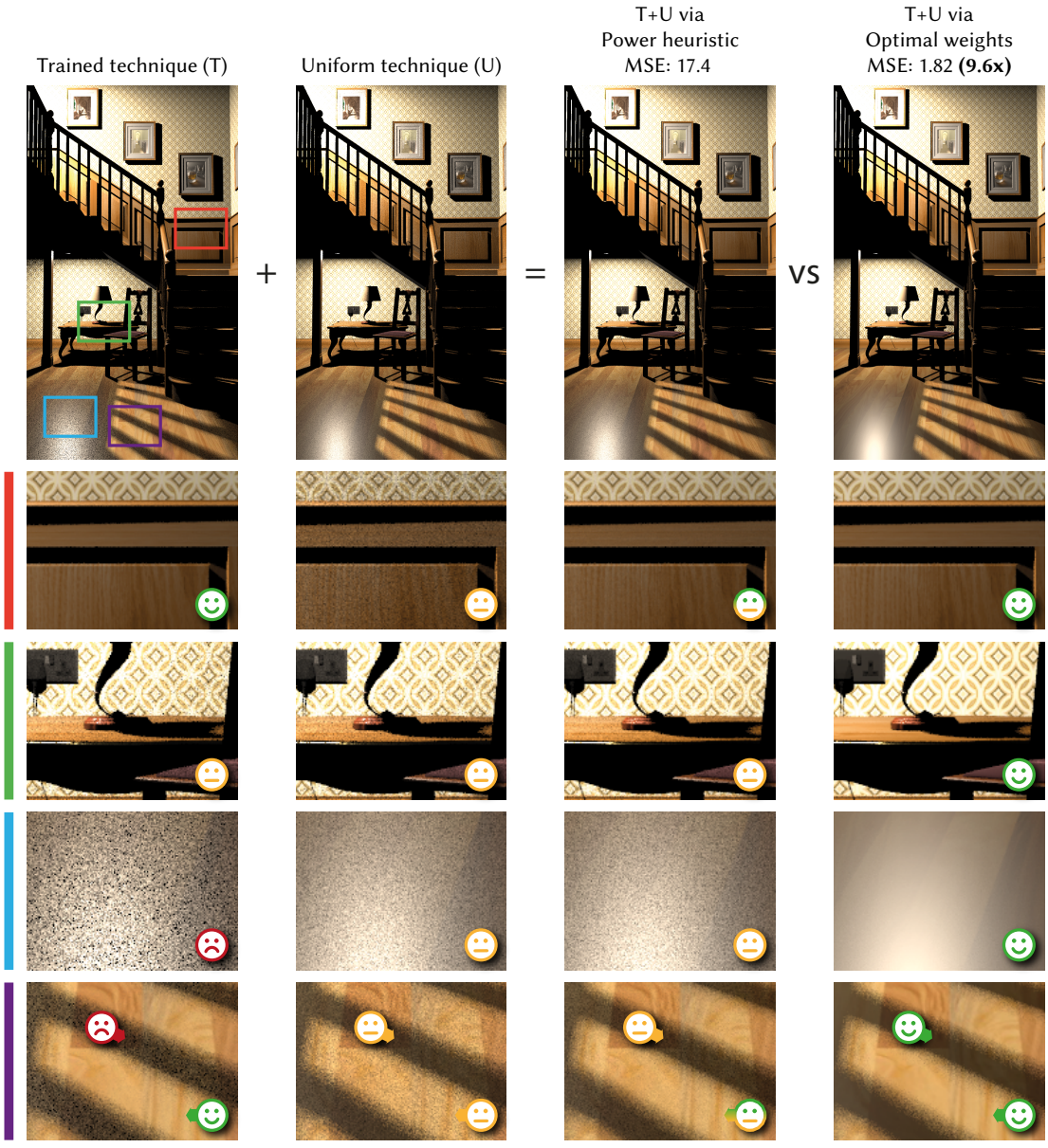


Figure 2.1: An equal-sample comparison (20 samples per technique per pixel) of direct illumination estimated by two sampling techniques for light selection *Trained* and *Uniform* and their MIS combination with the *Power heuristic* and our *Optimal weights*. All MSE values are $\times 10^{-10}$. See the main text for more information.

of its orientation (green smileys) but fails in shadows (red smileys) where it produces excessive noise. Therefore, it cannot be used alone and has to be combined with a second, defensive technique. An example of such a technique is *Uniform* which selects lights with equal probabilities. It does not perform particularly well anywhere in the scene (yellow smileys) but it also does not fail. By using MIS to combine these two techniques, the fail cases of the *Trained* technique are avoided (i.e., no more red smileys). However, when the traditional MIS weights are used (e.g., the power heuristic), the areas where *Trained* originally excelled are also affected and the performance there is decreased (i.e., no more fully green smileys). By using the proposed optimal weights, not only the good performance of the *Trained* technique is fully retained, but even the places where none of the two techniques performed well are improved (i.e., green smileys everywhere). This is enabled by the optimal weights being allowed to take negative values and Figure 2.2 shows that they are indeed negative in this scene. Altogether, the optimal weights lead to an overall 9.6 times lower error per sample taken than the power heuristic in this scene.

Apart from the variance reduction afforded by using the optimal weights in an existing sampling setup, we show that the optimal weights allow for an additional flexibility in designing the sampling techniques themselves. More specifically, variance properties of the optimal weights directly motivate new sampling techniques that – while performing poorly with balance and power heuristics – provide a substantial speedup with our optimal weights.

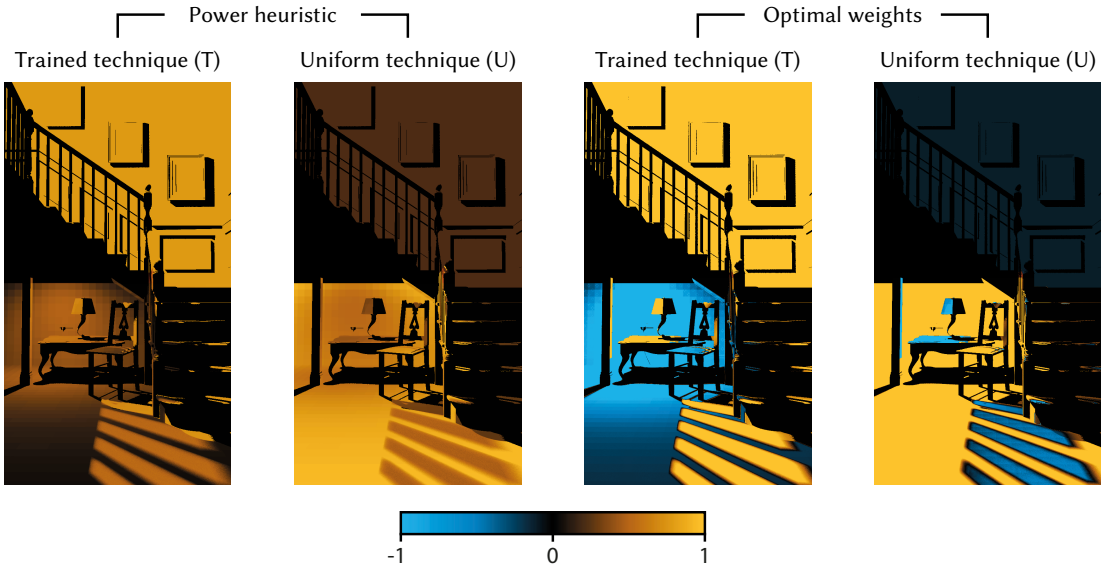


Figure 2.2: False colour images showing per-pixel average MIS weight values for each of the two techniques from Figure 2.1 as determined by the two weighting strategies. Note that the value range of the optimal weights in the two right images is clamped to $[-1, 1]$ for easier comparison (therefore, the two images do not sum up to 1).

2.1 Previous work

MIS in light transport Multiple importance sampling (MIS) [Veach and Guibas, 1995] offers a flexible way to combine a set of MC integral estimators, so as to achieve reasonable performance in a wide range of scenarios – a property referred to as *robustness*. It has been one of the keys behind the success of physically based light transport in VFX and computer animation [Keller et al., 2015]. MIS is typically used to combine a set of sampling techniques, each of which matches different features of the integrand, but none of which is a particularly good match across the entire domain. A prime example is direct illumination estimation [Veach and Guibas, 1995], where MIS is used to mix BRDF and light sampling techniques. Likewise, bidirectional path tracing [Veach and Guibas, 1995] and algorithms built upon it [Georgiev et al., 2012b; Hachisuka et al., 2012; Křivánek et al., 2014; Popov et al., 2015] rely on MIS to combine different techniques to sample entire light transport paths. In Markov chain Monte Carlo methods, MIS has been used to combine contributions from different chains [Kelemen et al., 2002; Šik et al., 2016] and to mix different target functions [Gruson et al., 2016].

Another important use case for MIS is defensive sampling: an adaptively trained sampling distribution is combined with a defensive strategy to ensure robustness to overfitting. In path guiding, adaptively constructed guiding distributions are typically mixed with BRDF sampling [Vorba et al., 2014; Herholz et al., 2016; Müller et al., 2017]. Similarly, in adaptive direct illumination sampling, MIS is used to combine learned light selection distributions with other, more defensive strategies, like in the work of Georgiev et al. [2012a], our work in the previous chapter or in the example in Figure 2.1.

MIS estimator design MIS represents a wide family of estimators parametrized by the combined sampling techniques, number of samples taken from each technique, and the weighting functions used to combine the samples. The choice of sampling techniques is application-dependent and we are not aware of any work addressing the sampling technique design specifically in the context of MIS. Another degree of freedom is the sample allocation. While Veach [1997] argues that "no strategy is much better than that of simply setting all [sample counts] equal", the fixed sample allocation has its shortcomings. For instance, if one technique is particularly good, samples from other techniques only serve to incur overhead and increase variance. To determine the sample allocation among BSDF, light, and photon map-based sampling, Pajot et al. [2011] introduce the notion of "representativity" – a measure of how well each technique samples a given integrand. Similarly, Lu et al. [2013] optimize sample allocation among BSDF and environment-map sampling by approximately minimizing the MIS estimator variance in terms of the sample counts. Havran and Sbert [2014] and Sbert et al. [2016] show that the optimal sample allocation must equalize the second moment of the weighted estimates corresponding to the individual sampling techniques. Sbert and Havran [2017] use the above result to design an approximate sample allocation solution and Sbert et al. [2018] introduce new balance heuristic estimators better than the balance heuristic with equal sample count per technique. Finally, Cappé et al. [2008] apply population Monte Carlo to optimize

sampling from mixture densities.

Alternative weighting heuristics In our work, we assume the sample counts to be given and we focus on designing the optimal MIS *weighting functions* – a problem setup shared with several previous works. In the context of many-light sampling, Georgiev et al. [2012a] point out that the balance, power, and maximum heuristics perform poorly, and they introduce the α-max heuristic with the aim to achieve better stratification among the sampling techniques. Popov et al. [2015] introduce a new weighting heuristic accounting for correlations between paths in bidirectional path tracing obtained by minimizing an upper bound of the variance of a correlated MIS estimator. Elvira et al. [2015, 2016] propose clustering of sampling techniques to cut the overhead introduced by evaluating the balance heuristic when the number of sampling techniques is high. While these works design new weighting heuristic for some specific cases, our goal is more ambitious: the provably optimal MIS weighting functions (for a given set of sampling techniques and fixed sample allocation).

Control variates and mixture sampling We show in Section 2.5 that our optimal weights are equivalent to optimal control variates (CV) [Lavenberg et al., 1982; Rubinstein and Marcus, 1985; Venkatraman and Wilson, 1986]. These were also studied by Owen and Zhou [2000], who realize CV by a mixture of sampling densities, and approximate the optimal CV coefficients through multiple linear regression over a set of observed estimates. We discuss the relation to their work in more detail in Section 2.6 and in Appendix 2.10.4. Fan et al. [2006] then applied Owen and Zhou's approach in rendering, and we compare to their approach in Section 2.7.5. In the follow-up work [He and Owen, 2014], the authors jointly optimize the CV coefficients and the sample allocation. They show that the MIS estimator variance is jointly convex in the above quantities and these can be found by convex optimization.

2.2 Multiple importance sampling

In this section, we review multiple importance sampling (MIS), as first described by Veach and Guibas [1995]. But let us first repeat the notation we use for the basics of MC integration.

Monte Carlo integration Let $F = \int_D f(x)\,\mathrm{d}x$ be the integral of a function $f : D \to \mathbb{R}$ over the domain D, and let there be a *sampling technique* for generating random samples from D following the probability density p such that $f(x) \neq 0 \Rightarrow p(x) \neq 0$. Then the importance sampling estimator $\langle F \rangle = f(X)/p(X)$, where the random variable X is distributed according to p, is unbiased, i.e., its expected value $\mathrm{E}[F]$ equals to F. The shape of p has a dramatic impact on the estimator's variance $\mathrm{Var}[\langle F \rangle]$: the closer p is to being proportional to the integrand f, the lower the variance.

Multiple importance sampling The idea of MIS is to improve the robustness of MC integration by incorporating N sampling techniques with probability

densities $p_i, i = 1, \ldots, N$, each of which could be a good match to a different feature of the integrand. An MIS estimator of the integral F is then defined as:

$$\langle F \rangle^* = \sum_i^N \sum_{j=1}^{n_i} \frac{w_i(X_{ij}) f(X_{ij})}{n_i p_i(X_{ij})}, \tag{2.1}$$

where $X_{ij} \in D$ is a random variable representing the j-th sample out of n_i samples generated by the i-th sampling technique, and $w_i(x)$ are *weighting functions*. All X_{ij} are independent. To keep the MIS estimator (2.1) unbiased, the weighting functions must satisfy:

$$f(x) \neq 0 \Rightarrow \textstyle\sum_{i=1}^N w_i(x) = 1, \tag{2.2}$$

$$p_i(x) = 0 \Rightarrow w_i(x) = 0, \tag{2.3}$$

i.e., they must sum up to 1 whenever $f(x)$ is nonzero, and each weight $w_i(x)$ must be zero whenever $p_i(x)$ is zero. A particular set of weighting functions is referred to as a *combination strategy*.

The above formulation of MIS, where a pre-determined number of samples are taken from each sampling technique, is known as the *multi-sample model*. On the other hand, the *one-sample model*

$$\langle F \rangle^{*1} = \frac{w_i(X_i) f(X_i)}{c_i p_i(X_i)}, \tag{2.4}$$

is evaluated by first selecting one sampling technique p_i at random with probability c_i, and then generating a sample X_i from it.

Balance and power heuristics All combination strategies yield unbiased estimators, but they can differ in their variance. The two most commonly used combination strategies are the *balance* and *power* heuristics, sharing the common form

$$w_i^p(x) = \frac{[n_i p_i(x)]^\beta}{\sum_{k=1}^N [n_k p_k(x)]^\beta}. \tag{2.5}$$

For the *balance heuristic*, we have $\beta = 1$. Veach and Guibas [1995] showed that no other combination strategy can have significantly lower variance than the balance heuristic; we revisit this near-optimality claim below. The *power heuristic*, for $\beta > 1$, is a strategy better suited for low-variance problems, i.e., those where one p_i closely matches the integrand [Veach and Guibas, 1995, Sec. 4.1]. We set $\beta = 2$, the choice that Veach and Guibas considered the best.

The same authors have additionally proposed the *cutoff* and *maximum* heuristics, but these are used less frequently in practice and we do not consider them here further (they throw away samples, are more complex to evaluate and usually inferior to the power heuristic).

2.3 Revisiting balance and power heuristics

In this section, we first illustrate sub-optimal performance of the balance and power heuristics, we then revisit the balance heuristic variance bounds, and show that allowing for negative weights may yield far lower variance than predicted by the bounds.

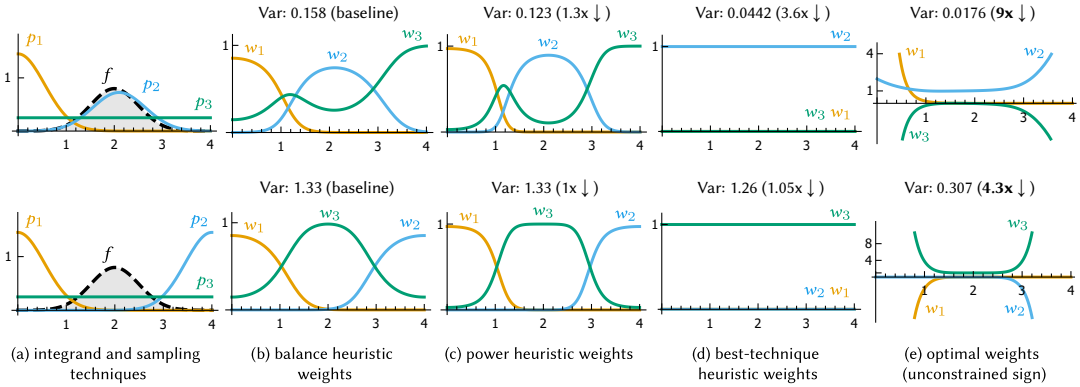


Figure 2.3: (a) The integrand f along with three sampling techniques p_1, p_2, and p_3. (b)–(f) The weighting functions associated with the balance, power, and best-technique heuristic, respectively. (e) Optimal weights (unconstrained sign). The two rows differ by the sampling technique p_2. See Appendix 2.10.6 for additional results for the cutoff and maximum heuristics (slightly worse than the balance heuristic) and a Mathematica notebook used to produce this figure.

2.3.1 Motivation

The balance and power heuristics enable combining sampling techniques in a robust way, so that the presence of a bad technique does not ruin the combined estimator's performance. 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.

Let us illustrate this observation on a simple 1D example shown in Figure 2.3. Column a) depicts an integration problem where the integral of a function f is estimated via MIS. Three sampling techniques, p_1, p_2, and p_3, are used, and one sample is taken from each. The two rows differ solely in the sampling technique p_2: while p_2 closely matches f in the first row, in the second row it is fairly different. Columns b) and c) plot, respectively, the balance and the power heuristic weights. We additionally define the *best-technique heuristic*, depicted in column d), as the combination strategy assigning unit weight to the single technique yielding the lowest variance and zero to the others. We can now compare the variance of the balance, power, and best-technique heuristics.

While in the second row the variance of all the three strategies is similar, there is a significant difference in the first row. The power heuristic achieves somewhat lower variance ($\sim$0.123) than the balance heuristic ($\sim$0.158), as this case is an instance of the low-variance problem due to p_2 being a good match to the integrand. Nonetheless, the best-technique heuristic has by far the lowest variance ($\sim$0.0442), almost 3x lower than the power heuristic. This is an inherent problem of the balance and power heuristics; they are not optimal and sometimes much worse than using the best technique alone.

2.3.2 Balance heuristic variance bounds: Are they valid?

The balance heuristic is widely used for its robustness and because it is provably good: Veach [1997] has shown that a) for the multi-sample model, no other

combination strategy can improve the variance beyond certain bounds, and b) it is optimal for the one-sample model. While the optimality proof for the one-sample model is valid in general, the proof of the variance bounds for the multi-sample model assumes non-negative weights – and this assumption results in an entire class of combination strategies being omitted.

We now revisit the proof for the multi-sample model and point out that allowing negative weights (affine combinations rather than convex) can improve the variance beyond the bounds derived by Veach. To simplify the notation, we denote the inner product of two functions a and b defined over the domain D as $\langle a, b\rangle = \int_D a(x)b(x)\,\mathrm{d}x$.

According to Veach, the variance of a multi-sample MIS estimator utilizing the balance heuristic is no larger than the variance of *any* other MIS estimator plus some fraction of F^2, more precisely:

$$\mathrm{Var}[\langle F\rangle^{b}] - \mathrm{Var}[\langle F\rangle^{*}] \leq \left(\frac{1}{\min_i n_i} - \frac{1}{\sum_{i=1}^{N} n_i}\right) F^2. \tag{2.6}$$

In the proof [Veach, 1997, p. 288], the variance of an MIS estimator

$$\mathrm{Var}[\langle F\rangle^{*}] = \underbrace{\sum_{i}^{N} \int_D \frac{w_i(x)^2 f(x)^2}{n_i p_i(x)}\,\mathrm{d}x}_{\text{first term}} - \underbrace{\sum_{i}^{N} \frac{1}{n_i}\langle w_i, f\rangle^2}_{\text{second term}} \tag{2.7}$$

was inspected. While the balance heuristic was the result of the minimization of the first term (giving the optimum for the one-sample model), the variance bound $({}^{1}/_{\min_i n_i} - {}^{1}/_{\sum_{i=1}^{N} n_i})F^2$ was established as the difference of the upper and the lower bound of the second term in (2.7). The lower bound derivation did not rely on any specific assumption, but in the upper bound derivation:

$$\begin{aligned}\sum_{i}^{N} \frac{1}{n_i}\langle w_i, f\rangle^2 &\leq \frac{1}{\min_i n_i}\sum_{i}^{N}\langle w_i, f\rangle^2 \\ &\overset{\star}{\leq} \frac{1}{\min_i n_i}\left(\sum_{i}^{N}\langle w_i, f\rangle\right)^2 = \frac{1}{\min_i n_i}F^2,\end{aligned} \tag{2.8}$$

the second inequality $\star$ holds only if

$$\langle w_i, f\rangle \geq 0, \tag{2.9}$$

that is, in the context of rendering where the integrand is non-negative, only when $w_i(x) \geq 0$.[1] For $\langle w_i, f\rangle < 0$ the upper bound on the variance of the balance heuristic can in fact be *larger* than what Veach's result suggests. See Figure 2.4 for an illustration.

To the best of our knowledge, this fact has not been previously recognized; the weighting functions are usually designed to be non-negative everywhere and for such the bounds are valid.

In what follows, we show that the non-negativity assumption is not necessary for an MIS estimator to remain unbiased. In fact, there are many cases where a combination strategy with $\langle w_i, f\rangle < 0$ produces an MIS estimator with variance lower than predicted by the bounds, and it can be significantly better than any other combination strategy considered by Veach [1997].

[1] To be precise, the condition is slightly weaker, because a weighting function w_i negative in a part of the domain may still yield $\langle w_i, f\rangle \geq 0$.

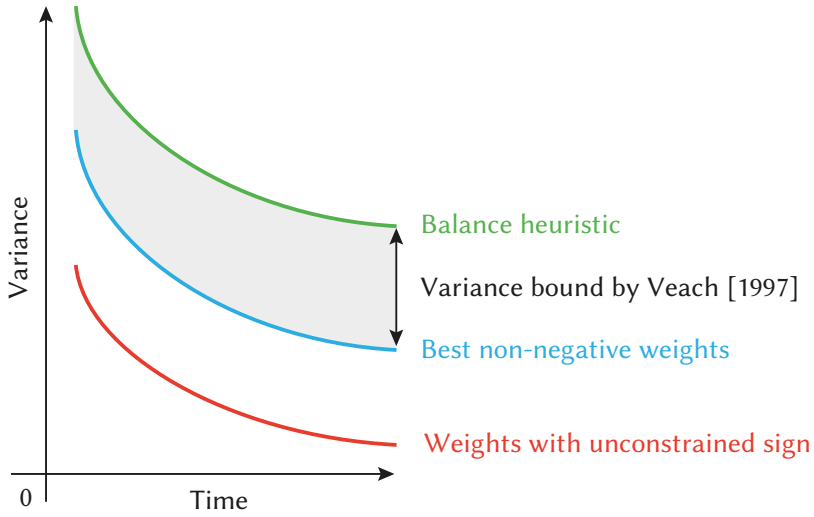


Figure 2.4: The upper bound on the variance of the balance heuristic given by Veach [1997] is valid only with respect to the best possible non-negative weights. For weights with unconstrained sign the difference in the variance can be much larger. Note that this figure is for an illustration purpose only. See Section 2.3.3 for an actual example breaking the bound.

2.3.3 Weights with unconstrained sign: An example

Suppose we define weights allowing negative values for our integration problems from Section 2.3.1. One example of such weights is shown in Figure 2.3e), along with the variance of the resulting estimators. They yield estimators with far lower variance than estimators utilizing any of the three heuristics discussed in Section 2.3.1.

For the integration problem in the second row, the MIS estimator using these weights has variance even *lower* than dictated by the variance bounds for the balance heuristic: the balance heuristic variance is $\sim$1.3 and the bounds are $\sim$0.5, meaning that any other MIS estimator $\langle F \rangle^*$ with only positive weights should have variance above 0.8 (according to (2.6)). But the MIS estimator with the negative weights has variance $\sim$0.3, which is well below this threshold.

In the next section, we derive weighting functions that *provably minimize the variance of the MIS estimator*, should there be no constraint on the weights' sign. In fact, the weights used in Figure 2.3e) resulted from that derivation.

2.4 Optimal MIS weights

We now derive optimal weights for MIS by directly minimizing the variance $\mathrm{Var}[\langle F \rangle^*]$ of the combined estimator (2.1), without imposing any restrictions other than those necessary to obtain an unbiased estimator. More formally:

Problem 1. Given the MIS estimator (2.1), minimize the functional $V[w_1, \ldots, w_N] = \mathrm{Var}[\langle F \rangle^*]$ in terms of weights w_i, while maintaining the constraints $\sum_{i=1}^{N} w_i(x) = 1$ and $p_i(x) = 0 \Rightarrow w_i(x) = 0$, and keeping the number of samples n_i and probability densities p_i fixed.

To describe the solution let us first define some terms:

Definition 1. *Let $f : D \to \mathbb{R}$ be a function to integrate, $p_i(x), i = 1, \ldots, N$ be a set of probability densities on D, and let n_i denote the number of samples taken from p_i. We define the* technique matrix *$\mathbf{A} = (a_{ik})$ as a symmetric $N \times N$ matrix with elements given by*

$$a_{ik} = \left\langle p_i, p_k / (\textstyle\sum_{j=1}^{N} n_j p_j) \right\rangle, \tag{2.10}$$

and the contribution vector *$\mathbf{b} = (b_1, \ldots, b_N)^\intercal$ as a column vector of length N composed of*

$$b_i = \left\langle f, p_i / (\textstyle\sum_{j=1}^{N} n_j p_j) \right\rangle. \tag{2.11}$$

The technique matrix is independent of the integrand f and it is composed of the inner products between all the probability densities normalised by the factor $(\sum_{i=1}^{N} n_i p_i)^{-1}$. Elements of the contribution vector represent contributions to the final $F = \int_D f(x)\,\mathrm{d}x$, because the dot product $(n_1, \ldots, n_N) \cdot \mathbf{b}$ equals to the integral F.

The solution to Problem 1 can now be summarized as follows:

Theorem 1. *Let the column vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)^\intercal$ satisfy the system of linear equations*

$$\mathbf{A}\boldsymbol{\alpha} = \mathbf{b}, \tag{2.12}$$

where $\mathbf{A}$ and $\mathbf{b}$ are the technique matrix and the contribution vector, respectively. Then the weighting functions

$$w_i^o(x) = \alpha_i \frac{p_i(x)}{f(x)} + \frac{n_i p_i(x)}{\sum_{j=1}^{N} n_j p_j(x)} \left(1 - \frac{\sum_{j=1}^{N} \alpha_j p_j(x)}{f(x)}\right) \tag{2.13}$$

minimize the functional $V[w_1, \ldots, w_N]$.

An MIS estimator using the weights $w_i^o(x)$ will be denoted $\langle F \rangle^o$. The proof of Theorem 1, given below, employs the calculus of variations (Appendix 2.10.1) to directly minimize the variance functional. It does not rely on any other assumptions than those necessary to ensure unbiasedness, and therefore the solution is indeed optimal in the MIS estimator family, i.e., *no other MIS combination strategy can result in a lower variance.*[2]

Please note that the weights in (2.13) satisfy the constraints $\sum_{i=1}^{N} w_i(x) = 1$ and $p_i(x) = 0 \Rightarrow w_i(x) = 0$ for any value of $\boldsymbol{\alpha}$. Therefore, any value of $\boldsymbol{\alpha}$ produces an unbiased estimator and the difference from the true $\boldsymbol{\alpha}$ only introduces extra variance. For $\boldsymbol{\alpha} = 0$, (2.13) becomes the balance heuristic.

Also note that due to the negative term in (2.13), the weights can be *negative*; the example in Figure 2.3e) shows that this indeed happens in practice.

[2]Applies to combination strategies in the MIS framework (2.1) as defined by Veach and Guibas [1995]. Other ways of combining samples, e.g., non-linear ones, may still perform better, but these do not belong to the MIS family.

2.4.1 Proof of Theorem 1

We prove THEOREM 1 by construction. To do that, we seek weighting functions $w_i, i = 1, \ldots, N$ that minimize the variance functional $V[w_1, \ldots, w_N]$, given by (2.7), constrained by $\sum_{i=1}^{N} w_i(x) = 1$ and $p_i(x) = 0 \Rightarrow w_i(x) = 0$. To simplify the derivation, we leave out the latter constraint, and verify it at the end. Dropping the function arguments, the solution is given by the minimum of the Lagrangian

$$\mathbf{L} = V[w_1, \ldots, w_N] - \int_D \lambda \left(\textstyle\sum_{i=1}^{N} w_i \; - 1 \right) \mathrm{d}x, \tag{2.14}$$

in terms of the weights w_i and the Lagrange multiplier $\lambda : D \to \mathbb{R}$. To find the minimum, we set all the partial functional derivatives $\partial \mathbf{L}/\partial w_i$ and $\partial \mathbf{L}/\partial \lambda$ to zero. Using the relation (2.32), we find $\partial \mathbf{L}/\partial w_i$ as

$$\begin{aligned} \frac{\partial}{\partial \varepsilon}\Big|_{\varepsilon=0} \mathbf{L}(\ldots, w_i + \varepsilon\delta_i, \ldots) = \Big|_{\varepsilon=0} \Bigg[\frac{2}{n_i} \int_D \frac{(w_i + \varepsilon\delta_i) f^2 \delta_i}{p_i} \, \mathrm{d}x - \\ \frac{2}{n_i} \int_D (w_i + \varepsilon\delta_i) f \, \mathrm{d}x \int_D \delta_i f \, \mathrm{d}x - \int_D \lambda \delta_i \, \mathrm{d}x \Bigg] \\ = \int_D \Big(\underbrace{\frac{2 w_i f^2}{p_i n_i} - \frac{2f}{n_i} \int_D w_i f \, \mathrm{d}x - \lambda}_{\partial \mathbf{L}/\partial w_i} \Big) \, \delta_i \, \mathrm{d}x. \end{aligned} \tag{2.15}$$

We proceed in a similar way to find $\partial \mathbf{L}/\partial \lambda$. This gives us a set of equations for w_i and λ:

$$w_i - \frac{p_i}{f} \int_D w_i f \, \mathrm{d}x = \frac{n_i}{2} \lambda \frac{p_i}{f^2}, \qquad \textstyle\sum_{i=1}^{N} w_i = 1 \tag{2.16}$$

The equation on the left can be rewritten as

$$w_i = \alpha_i \frac{p_i}{f} + \frac{n_i}{2} \lambda \frac{p_i}{f^2}, \quad \text{with} \quad \alpha_i = \int_D w_i f \, \mathrm{d}x. \tag{2.17}$$

Plugging the above equation for w_i into the constraint $\sum_{i=1}^{N} w_i = 1$, (i.e., $\partial \mathbf{L}/\partial \lambda = 0$), we can solve for the multiplier λ:

$$\lambda = 2 \, \frac{f^2 - f \sum_i^N \alpha_i p_i}{\sum_i^N n_i p_i}. \tag{2.18}$$

The final form of the optimal weights $w_i^o(x)$, given by (2.13), is now obtained by plugging (2.18) back into (2.17), left.

Our next step is to find the $\alpha_i, i = 1, \ldots, N$. Plugging the optimal weights (2.13) into (2.17), right, we obtain a set of equations for α_j

$$\int_D n_i p_i \frac{f - \sum_{j=1}^{N} \alpha_j p_j}{\sum_{k=1}^{N} n_k p_k} \, \mathrm{d}x = 0, \quad i = 1 \ldots N, \tag{2.19}$$

which can be rearranged into

$$\sum_{j=1}^{N} \alpha_j \int_D \frac{p_i p_j}{\sum_{k=1}^{N} n_k p_k} \, \mathrm{d}x = \int_D \frac{p_i f}{\sum_{k=1}^{N} n_k p_k} \, \mathrm{d}x. \tag{2.20}$$

This can be written in a matrix form as $\mathbf{A}\boldsymbol{\alpha} = \mathbf{b}$, where $\mathbf{A}$ and $\mathbf{b}$ are the technique matrix and contribution vector from *Definition 1* and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)^\intercal$.

From (2.17) we can see that whenever $p_i(x) = 0$, we get $w_i(x) = 0$, which validates our second constraint. This completes the proof.

2.4.2 Solution existence and uniqueness

Here we discuss the existence and uniqueness of the optimal weights from THEOREM 1, and show that there are infinitely many values of $\boldsymbol{\alpha}$ yielding the same optimal weights.

Existence and uniqueness The optimal weights exist whenever the linear system (2.12) is consistent. To prove the consistency, we would need to show that if two rows i, j of $\mathbf{A}$ are the same, then also $b_i = b_j$, which we have not yet been able to do.

Nonetheless, it holds that whenever one sampling strategy is a convex combination of other strategies, i.e., $p_i = \sum_{j \neq i} c_j p_j$, then the i-th row of $\mathbf{A}$ becomes the same linear combination of the other rows, and $b_i = \sum_{j \neq i} c_j b_j$. In such cases the linear system becomes singular (but remains consistent) and there are infinitely many solutions for $\boldsymbol{\alpha}$, each yielding possibly *different* MIS weights, but producing an MIS estimator *with the same variance*. This is because $\boldsymbol{\alpha} \in \{\mathbf{u} + \mathbf{v} | \mathbf{A}\mathbf{u} = \mathbf{b} \wedge \mathbf{v} \in \text{Null}(\mathbf{A})\}$, and (2.42) is the same for all such $\boldsymbol{\alpha}$. If the linear system is non-singular, the $\boldsymbol{\alpha}$ vector and the resulting weights are unique.

Full solution for $\boldsymbol{\alpha}$ Adding a term $s\mathbf{n}$, where $s \in \mathbb{R}$ and $\mathbf{n} = (n_1, \ldots, n_N)^\intercal$, to $\boldsymbol{\alpha}$ produces the same weights, despite the modified vector $\boldsymbol{\alpha}$ not being a solution to the system (2.12). This is because the offset $s\mathbf{n}$ cancels out when the modified $\boldsymbol{\alpha}$ is plugged into the weights (2.13). Therefore all $\widetilde{\boldsymbol{\alpha}} = \mathbf{A}^{-1}\mathbf{b} + s\mathbf{n}$ yield the same optimal weights and we refer to $\widetilde{\boldsymbol{\alpha}}$ as to the *full solution* for $\boldsymbol{\alpha}$.

2.5 Optimal weights as control variates

In this section, we show that the optimal weights from THEOREM 1 can be interpreted as control variates [Glasserman, 2003]. Based on that we provide some intuition on the integration problems for which the optimal weights will yield the highest variance reduction.

2.5.1 Background: Control variates

Consider an MC estimator $\langle F \rangle$ for the integral $F = \int f(x)\mathrm{d}x$. Take a set of K other estimators $\langle G_i \rangle$ with expected values G_i, $i = 1, \ldots, K$, called control variates. Rewriting the original estimator $\langle F \rangle$ as

$$\begin{aligned} \langle F \rangle^{\text{CV}} &= \langle F \rangle + \textstyle\sum_{i=1}^{K} \gamma_i (G_i - \langle G_i \rangle) \\ &= \textstyle\sum_{i=1}^{K} \gamma_i G_i + \langle F \rangle - \sum_{i=1}^{K} \gamma_i \langle G_i \rangle \end{aligned} \tag{2.21}$$

can reduce variance when some of the $\langle G_i \rangle$ are correlated with $\langle F \rangle$ and $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_K)^\intercal$ is chosen appropriately. Variance is minimized for $\boldsymbol{\gamma}$ solving the system $\boldsymbol{\Sigma}\boldsymbol{\gamma} = \boldsymbol{\sigma}$, where $\boldsymbol{\Sigma} = (\sigma_{ik})$ is a $K \times K$ covariance matrix, and $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_K)^\intercal$ is a covariance vector, with their elements defined as

$$\sigma_{ik} = \text{Cov}[\langle G_i \rangle, \langle G_k \rangle], \qquad \sigma_i = \text{Cov}[\langle G_i \rangle, \langle F \rangle]. \tag{2.22}$$

This is a well-known form [Lavenberg et al., 1982; Rubinstein and Marcus, 1985; Venkatraman and Wilson, 1986]. In the case of a single control variate ($K = 1$), variance is minimized for $\gamma_1 = \mathrm{Cov}[\langle G_1\rangle, \langle F\rangle]/\mathrm{Var}[\langle G_1\rangle]$. For $\gamma_1 = 1$, $\langle F\rangle = f(X)/p(X)$, and $\langle G\rangle = g(X)/p(X)$ the estimator $\langle F\rangle^{\mathrm{CV}}$ can be rewritten as $\frac{f(X)}{p(X)} - \frac{g(X)}{p(X)} + G$, which is the form we used in the previous chapter in Section 1.5.

2.5.2 Optimal weights as control variates

Let us plug the optimal weights from (2.13) into the multi-sample MIS estimator in (2.1). Denoting $M = \sum_i^N n_i$, $c_i = n_i/M$, and $p_{\mathbf{c}}(x) = \sum_i^N c_i p_i(x)$, we obtain the optimal MIS estimator $\langle F\rangle^o$ in the form

$$\langle F\rangle^o = \sum_{k=1}^{N} \alpha_k + \frac{1}{M} \sum_{i}^{N} \sum_{j=1}^{n_i} \left(\frac{f(X_{ij})}{p_{\mathbf{c}}(X_{ij})} - \frac{\sum_{k=1}^N \alpha_k p_k(X_{ij})}{p_{\mathbf{c}}(X_{ij})} \right). \tag{2.23}$$

The above form can be interpreted as the control variate estimator (2.21) utilizing either one or N control variates. Here, for the purpose of further analysis, we interpret it as the former: Using $g(x) = \sum_{k=1}^N \alpha_k p_k(x)$, the above form is equivalent to (2.21) with $K = 1$, where

$$\langle F\rangle = \frac{1}{M} \sum_{i}^{N} \sum_{j=1}^{n_i} \frac{f(X_{ij})}{p_{\mathbf{c}}(X_{ij})}, \qquad \langle G_1\rangle = \frac{1}{M} \sum_{i}^{N} \sum_{j=1}^{n_i} \frac{g(X_{ij})}{p_{\mathbf{c}}(X_{ij})}, \tag{2.24}$$

the expected value $G_1 = \int \sum_{k=1}^N \alpha_k p_k(x)\,\mathrm{d}x = \sum_{k=1}^N \alpha_k$, and the parameter $\gamma_1 = 1$. The estimator $\langle F\rangle$ above is a multi-sample MIS estimator of F utilizing the balance heuristic, further denoted $\langle F\rangle^b$. Similarly, the $\langle G_1\rangle$ estimator above is an MIS estimator of $\int_D g(x)\,\mathrm{d}x$, and we denote it $\langle G\rangle^b$.

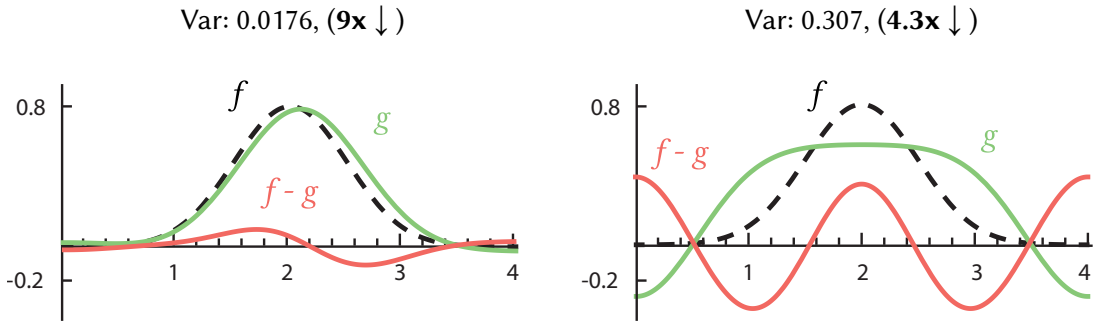
In other words, the optimal weights are equivalent to the balance heuristic combined with a control variate of the form $\sum_{k=1}^N \alpha_k p_k(x)$. And since (2.13) are valid MIS weights for any value of $\boldsymbol{\alpha}$ (as we discussed in Section 2.4), any combination of the balance heuristic with some mixture of sampling pdfs as a control variate is equivalent to some MIS weights.

2.5.3 Variance considerations

The $\boldsymbol{\alpha}$ vector from Theorem 1 yields an optimal control variate of the general form (2.23), minimizing its variance.[3] The variance is then equal to the variance of the balance heuristic MIS estimator of $\int_D f(x) - g(x)\,\mathrm{d}x$, and as such it depends on the magnitude of $f - g$ as well as its proportionality to $p_{\mathbf{c}}$. Intuitively, the "closer" the function g is in its shape to the integrand f, the higher the variance reduction due to the optimal weights compared to the balance heuristic. Moreover, the variance of $\langle F\rangle^o$ becomes zero for $f = g$, that is, whenever the integrand f can be written as a linear combination of the sampling pdfs p_k.

In Figure 2.5 we plot the difference $f - g$ for the two integration problems from Section 2.3.1, where g is computed using the vector $\boldsymbol{\alpha}$ for the respective optimal weights. The overall amplitude of the difference is smaller for the first example

[3] If it was not the optimum, then other weights better than $w_i^o(x)$ would exist, which is a contradiction.



(a) first row (b) second row

Figure 2.5: Illustration of the difference $f - g$ for the first (a) and second (b) row of the integration problem from Figure 2.3a along with the variance of MIS using the optimal weights, and the variance reduction with respect to MIS using the balance heuristic. Note, that the flatter the difference the higher the variance reduction.

and larger for the second, which is in line with the higher variance reduction for the former case. We build on these observations in Section 2.7.4 to design new sampling techniques specifically aiming at variance reduction with the optimal weights.

Relation to the balance heuristic The optimal estimator $\langle F \rangle^o$ is given by the sum $\sum_{k=1}^{N} \alpha_k$ (no variance) plus the difference of two correlated MIS estimators $\langle F \rangle^b$ and $\langle G \rangle^b$, given by (2.24). The variance of $\langle F \rangle^o$ is therefore equal to the variance of that difference, i.e., $\mathrm{Var}[\langle F \rangle^o] = \mathrm{Var}[\langle F \rangle^b - \langle G \rangle^b]$. In Appendix 2.10.2 we prove that

$$\mathrm{Var}[\langle F \rangle^o] = \mathrm{Var}[\langle F \rangle^b] - \mathrm{Var}[\langle G \rangle^b]. \tag{2.25}$$

First, this result confirms the expected: the optimal estimator variance is less than or equal to the balance heuristic variance. More importantly, it shows that the balance heuristic is optimal whenever $\mathrm{Var}[\langle G \rangle^b] = 0$. This occurs when $\boldsymbol{\alpha}$ is collinear with the vector $\mathbf{n} = (n_1, \ldots, n_N)^\intercal$, that is, when the elements of the vector $\boldsymbol{\alpha}$ are proportional to the number of samples from the individual sampling techniques. This result can be used to detect the achievable variance improvement over the balance heuristic.

Covariance vector and matrices Interpreting (2.23) as a form utilizing N control variates

$$\langle G_k \rangle = \frac{1}{M} \sum_{i}^{N} \sum_{j=1}^{n_i} \frac{p_k(X_{ij})}{p_{\mathbf{c}}(X_{ij})}, \quad k = 1, \ldots, N, \tag{2.26}$$

with expected values $G_k = 1$, we can verify that $\boldsymbol{\alpha}$ indeed represents the optimal parameters $\boldsymbol{\gamma}$. The technique matrix $\mathbf{A}$ and contribution vector $\mathbf{b}$ in Theorem 1 are related to their covariance counterparts (defined by (2.22)) by

$$\boldsymbol{\Sigma} = (\mathbf{I} - \mathbf{AN})\mathbf{A}, \qquad \boldsymbol{\sigma} = (\mathbf{I} - \mathbf{AN})\mathbf{b}, \tag{2.27}$$

where $\mathbf{N}$ is a diagonal $N \times N$ matrix with the sample count n_i along the diagonal. The above relation emerges if we obtain the covariances σ_{ik} and σ_i in a similar way we obtained the covariance (2.38) in Appendix 2.10.2. It follows that the full solution for alphas from Section 2.4.2 solves the system $\mathbf{\Sigma}\boldsymbol{\gamma} = \boldsymbol{\sigma}$.

2.6 Optimal weights in practice

An MIS estimator with the optimal weights (2.13) cannot be evaluated directly since the inner products forming the technique matrix $\mathbf{A}$ and contribution vector $\mathbf{b}$ from *Definition 1* generally do not have a closed-form solution. Our implementation therefore follows three steps: 1) estimation of the technique matrix $\mathbf{A}$ and contribution vector $\mathbf{b}$; 2) estimation of the vector $\boldsymbol{\alpha}$ using the estimated $\mathbf{A}$ and $\mathbf{b}$; and 3) realization of an approximate optimal estimator $\langle F \rangle^o$ using the estimated $\boldsymbol{\alpha}$. We now elaborate on the individual steps.

2.6.1 Estimating technique matrix and contribution vector

The elements of the technique matrix $\mathbf{A}$ and the contribution vector $\mathbf{b}$ are given by the integrals (2.10) and (2.11), respectively. We estimate these integrals using MIS with the balance heuristic, and denote the result $\langle \mathbf{A} \rangle$ and $\langle \mathbf{b} \rangle$.[4] In the matrix form, the estimators $\langle \mathbf{A} \rangle$ and $\langle \mathbf{b} \rangle$ can be expressed as follows:

$$\langle \mathbf{A} \rangle = \sum_{i}^{N} \sum_{j=1}^{n_i} \mathbf{W}_{ij} \mathbf{W}_{ij}^{\intercal}, \quad \langle \mathbf{b} \rangle = \sum_{i}^{N} \sum_{j=1}^{n_i} f(X_{ij})\ S_{ij}\ \mathbf{W}_{ij}, \tag{2.28}$$

where $S_{ij} = \left(\sum_{k=1}^{N} n_k p_k(X_{ij}) \right)^{-1}$ and $\mathbf{W}_{ij}$ is the column vector of all sampling techniques evaluated at X_{ij} and scaled by S_{ij},

$$\mathbf{W}_{ij} = S_{ij} \left(p_1(X_{ij}), \ldots, p_N(X_{ij}) \right)^{\intercal}. \tag{2.29}$$

Recall from (2.1) that X_{ij} denotes the j-th sample from p_i.

2.6.2 Estimating the vector alpha

The vector $\boldsymbol{\alpha}$ is given by the linear system (2.12). We estimate $\langle \boldsymbol{\alpha} \rangle$ by least squares minimization, because the *estimated* system $\langle \mathbf{A} \rangle \langle \boldsymbol{\alpha} \rangle = \langle \mathbf{b} \rangle$ may be (close to) singular, especially when the estimates $\langle \mathbf{A} \rangle$ and $\langle \mathbf{b} \rangle$ are based on just a few samples. While the $\langle \mathbf{A} \rangle$ and $\langle \mathbf{b} \rangle$ estimates are unbiased, the estimate $\langle \boldsymbol{\alpha} \rangle$ is generally biased, because the matrix inversion involved in solving the linear system does not preserve expectation, i.e. $(\mathrm{E}[\langle \mathbf{A} \rangle] = \mathbf{A}) \not\Rightarrow (\mathrm{E}[\langle \mathbf{A} \rangle^{-1}] = \mathbf{A}^{-1})$. Nonetheless, as we explained in Section 2.4, the resulting MIS estimator will be unbiased for any value of $\boldsymbol{\alpha}$ and the difference between the true $\boldsymbol{\alpha}$ and its particular estimate $\langle \boldsymbol{\alpha} \rangle$ only introduces extra variance in the final estimator $\langle F \rangle^o$. The extra variance diminishes thanks to the $\langle \boldsymbol{\alpha} \rangle$ estimate being *consistent*; this follows from $\langle \mathbf{A} \rangle^{-1}$ approaching $\mathbf{A}^{-1}$ with the increasing sample count in the $\langle \mathbf{A} \rangle$ estimate.

[4] The power heuristic is less appropriate, as the integrals (2.10) and (2.11) are not low-variance, i.e., no sampling strategy is a particularly good match for any of the integrands.

2.6.3 Approximate optimal estimator

We have various options to approximate the optimal estimator $\langle F \rangle^o$. For instance, we could estimate $\langle \boldsymbol{\alpha} \rangle$ from an initial batch of samples, hold it fixed, and use it to evaluate the optimal weights (2.13) for all subsequent samples. This approach would be suboptimal, however, as the estimated alphas would not evolve over time.

Progressive estimator The computation is performed in iterations. In each iteration, we first draw n_i samples from each sampling technique p_i, $i = 1, \ldots, N$. We then compute $\langle \boldsymbol{\alpha} \rangle$ based on the $\langle \mathbf{A} \rangle$ and $\langle \mathbf{b} \rangle$ estimates from the previous iterations. We plug it in formula (2.23) of the MIS estimator $\langle F \rangle^o$ to compute the integral estimate from the current samples and accumulate it. Note that for the first iteration we set $\langle \boldsymbol{\alpha} \rangle$ to zero which is equivalent to evaluating an MIS estimator with the balance heuristic. Finally, we update $\langle \mathbf{A} \rangle$ and $\langle \mathbf{b} \rangle$ using the current samples according to (2.28) and proceed to the next iteration. This procedure yields an unbiased MIS estimator, the efficiency of which improves over time as the estimates $\langle \boldsymbol{\alpha} \rangle$ converge to the true value. Since recomputing $\langle \boldsymbol{\alpha} \rangle$ every iteration may be time-consuming, we also allow for performing the recomputation only after every U updates to $\langle \mathbf{A} \rangle$ and $\langle \mathbf{b} \rangle$. We will call U the update step. See Algorithm 1 in Figure 2.6 for a pseudocode.

An important note: Despite the division by the integrand $f(x)$ in the optimal weights (2.13), the MIS estimator $\langle F \rangle^o$ in the form (2.23) exists even for $f(x) = 0$. In contrast to previous MIS weighting heuristics, samples X with $f(X) = 0$ *must not be discarded*, because they generally have a non-zero contribution to the estimator.

ALGORITHM 1: Progressive

```
1  ⟨A⟩ ← 0^{N×N}; ⟨b⟩ ← 0^{N×1};
2  ⟨α⟩ ← 0^{N×1}; result ← 0;
3  for iteration ← 0 to maxIterations − 1 do
4      for i ← 1 to N do
5          {X_ij}_{j=1}^{n_i} ← draw n_i samples from
             technique p_i;
6      end
7      if (iteration ≥ 1) and (iteration
         mod U) = 0 then
8          ⟨α⟩ ← solve linear system ⟨A⟩⟨α⟩ = ⟨b⟩;
9      end
10     estimate ← evaluate ⟨F⟩^o using ⟨α⟩; // (2.23)
11     result ← result + estimate;
12     ⟨A⟩ ← ⟨A⟩ + Σ_i^N Σ_{j=1}^{n_i} W_ij W_ij^T; // (2.28)
13     ⟨b⟩ ← ⟨b⟩ + Σ_i^N Σ_{j=1}^{n_i} f(X_ij) S_ij W_ij;
         // (2.28)
14 end
15
16 return result/maxIterations
```

ALGORITHM 2: Direct

```
1  ⟨A⟩ ← 0^{N×N}; ⟨b⟩ ← 0^{N×1};
2
3  for iteration ← 0 to maxIterations − 1 do
4      for i ← 1 to N do
5          {X_ij}_{j=1}^{n_i} ← draw n_i samples from
             technique p_i;
6      end
7
8
9
10
11
12     ⟨A⟩ ← ⟨A⟩ + Σ_i^N Σ_{j=1}^{n_i} W_ij W_ij^T; // (2.28)
13     ⟨b⟩ ← ⟨b⟩ + Σ_i^N Σ_{j=1}^{n_i} f(X_ij) S_ij W_ij;
         // (2.28)
14 end
15 ⟨α⟩ ← solve linear system ⟨A⟩⟨α⟩ = ⟨b⟩;
16 return Σ_i^N ⟨α_i⟩
```

Figure 2.6: A pseudocode for two estimators with the approximated optimal MIS weights: the Progressive and Direct estimators (see Section 2.6.3). Differences are highlighted in red.

Direct estimator By definition (2.17), each α_i is equal to the integral of f weighted by the optimal weight w_i^o:

$$\alpha_i = \int_D f(x) w_i^o(x)\,\mathrm{d}x. \tag{2.30}$$

Because the weighting functions sum up to one for all $x \in D$, we can express the integral of f as

$$\int_D f(x)\,\mathrm{d}x = \int_D f(x)\left(\textstyle\sum_{i=1}^N w_i^o(x)\right)\,\mathrm{d}x = \textstyle\sum_{i=1}^N \alpha_i. \tag{2.31}$$

We can therefore obtain an estimator $\langle F \rangle$ by summing the elements of $\langle \boldsymbol{\alpha} \rangle$. Such a *Direct estimator* will be biased, but consistent as follows from biasedness and consistency of $\langle \boldsymbol{\alpha} \rangle$, discussed in Section 2.6.2.

The Direct estimator is simpler and more efficient than the Progressive one: in each iteration, it only updates the $\langle \mathbf{A} \rangle$ and $\langle \mathbf{b} \rangle$ estimates, and the linear system is solved for $\langle \boldsymbol{\alpha} \rangle$ only once after all iterations have been processed. See Algorithm 2 in Figure 2.6.

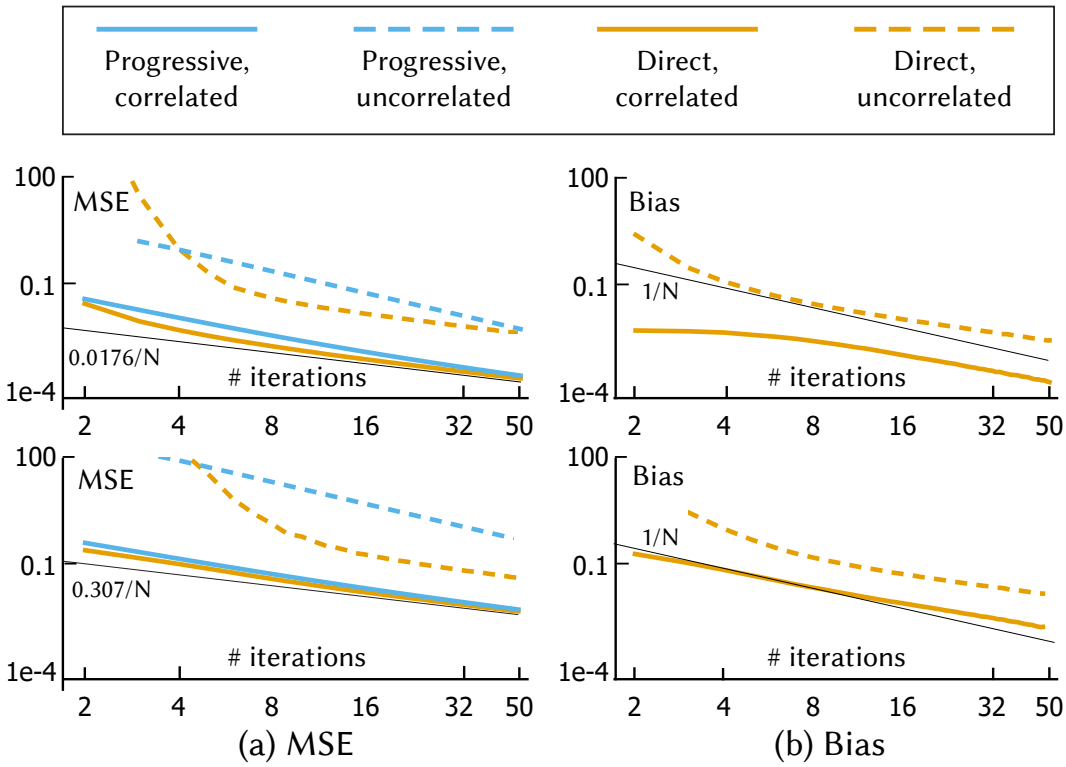


Figure 2.7: (a) MSE of the Progressive and Direct estimators versus the overall number of iterations plotted on the log-log scale, when used to estimate the first (top row) and second (bottom row) integration problem from Figure 2.3a. The black line represents the analytically computed variance of MIS estimator with the optimal weights divided by N iterations. (b) Bias of the Direct estimator on the log-log scale. The black line corresponds to $1/N$, where N is a number of iterations on the horizontal axis. For both (a) and (b) cases we show the correlated and uncorrelated estimator variants.

2.6.4 Empirical tests

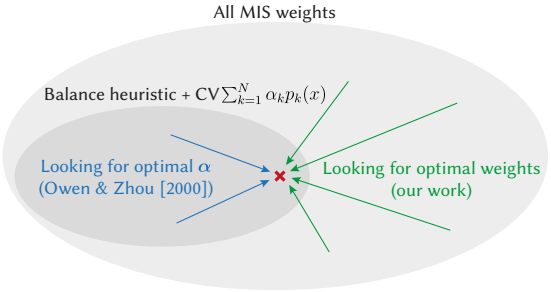
Figure 2.7 illustrates the behaviour of the Progressive and Direct estimators, described above, on the example integration problem from Section 2.3.1 (depicted in Figure 2.3a). The MSE of different estimators as a function of the number of iterations is shown in Figure 2.7a. The *uncorrelated* version uses two independent sets of samples to estimate the technique matrix $\langle \mathbf{A} \rangle$ and the contribution vector $\langle \mathbf{b} \rangle$, respectively. The *correlated* version uses a single sample set for both.

In the correlated case (solid lines), both Progressive (cyan) and Direct (orange) estimators have similar performance, almost as good as the reference optimal estimator with a known $\boldsymbol{\alpha}$ vector (solid black). Interestingly, the behaviour in the uncorrelated case (dashed lines) is vastly different, as both estimators perform much worse than in the correlated case. We hypothesize that the correlation between $\langle \mathbf{A} \rangle$ and $\langle \mathbf{b} \rangle$ is the key to a good performance of both estimators, though a full understanding of this effect remains for future work.

The Direct estimator is biased. In Figure 2.7b, we can observe that both the correlated and uncorrelated versions are consistent, with the bias diminishing roughly at a $O(N^{-1})$ rate with the total number of iterations.[5] Similarly to the MSE, the bias is much lower in the correlated case. As discussed above, the Progressive estimator is unbiased, which we have verified experimentally.

2.6.5 Discussion of related work

Interestingly, the optimal estimator (2.23) has the same form as the control variate estimator analysed by Owen and Zhou [2000]. They start off by postulating this form, using the mixture of sampling pdfs as a control variate, and then they estimate the optimal mixing parameters $\boldsymbol{\alpha}$ for this stated estimator form. We, on the other hand, show that both the form and the parameters of this estimator naturally emerge by direct minimization of the MIS estimator's variance, and that it provides the optimal solution in the MIS family.



Owen and Zhou estimate $\boldsymbol{\alpha}$ using linear regression on observed samples. For that they have to solve a (singular) linear system, but they also propose solving an equivalent (regular) truncated system, obtained by skipping some regressors. Though derived in a different way, their proposed $\boldsymbol{\alpha}$ estimator (denoted as $\hat{\boldsymbol{\beta}}$ in their Section 3), even in its truncated form, is in fact equivalent to our $\langle \boldsymbol{\alpha} \rangle$, *provided that the components of our technique matrix* $\mathbf{A}$ *and the contribution vector* $\mathbf{b}$ *are estimated with the balance heuristic as described in Section 2.6.1.* Hence, their approach can be seen as one particular way of approximating the optimal

[5] Bias is computed as the average absolute error of 1000 independent estimator realizations, each obtained using the number of samples on the horizontal axis.

solution given by THEOREM 1. Our result is more general as it is amenable to alternative strategies to approximate the optimal $\mathbf{A}$, $\mathbf{b}$, and $\boldsymbol{\alpha}$. See Appendix 2.10.4 for details.

2.7 Applications and results

In this section we apply the optimal weights to light transport, specifically to direct illumination estimation. We show that they perform particularly well when used for defensive sampling. Furthermore, we introduce new sampling techniques that further increase the efficiency when mixed by the optimal weights. Finally, we provide additional results including a comparison of the performance of the Progressive and Direct estimators or a comparison to an adaptation of the approach by Owen and Zhou [2000].

2.7.1 Implementation

Our applications are implemented in PBRT [Pharr et al., 2016], and a link to the implementation source code is provided in Appendix 2.10.6. All scenes were rendered on a machine with an Intel Core i7-5820K CPU (6 cores, 12 threads) and 64 GB of RAM.

We implement the Progressive and Direct estimators as described in Section 2.6. Calculation proceeds pixel-by-pixel, in each pixel the respective algorithm from Figure 2.6 is called and its output is stored in the pixel. We take one sample per technique per iteration, i.e., $n_i = 1, i = 1, \ldots, N, N = 2$ and set $maxIterations$ to the target number of samples per technique per pixel. For an equal-time comparison we set $maxIterations$ individually for each estimator so they all render for roughly the same time.

2.7.2 Results structure

In Section 2.7.3 and Section 2.7.4 we compare our Direct estimator to the power heuristic combination for two different applications. In Section 2.7.5 we compare the Direct and Progressive approaches, and the adaptation of the approach by Owen and Zhou [2000]. Appendix 2.10.6 then provides a link to a complete set of results including the Direct estimator, multiple versions of the Progressive estimator, and both the balance and power heuristic for all our test scenes.

2.7.3 Application I: Defensive sampling

One application where the optimal MIS weights have a particularly strong impact is defensive sampling. It is typically employed by adaptive approaches that construct sampling distributions based on previous samples [Herholz et al., 2016; Georgiev et al., 2012a]. The trained sampling technique is then mixed with one or more defensive techniques (e.g., uniform) to prevent bias and artefacts due to noise from the previous samples. Ideally, the trained technique has low variance across the majority of the domain, which is likely to trigger the low-variance problem discussed by Veach and Guibas [1995]. However, the power, maximum, and cutoff heuristics, proposed to address this case, still underperform (as pointed out

by Georgiev et al. [2012a]). While the heuristics improve robustness, they also increase variance where the trained technique works well.

Our optimal MIS weights are particularly effective at solving this issue: the optimal combination of multiple sampling techniques can never be worse than a single technique on its own.[6] Therefore, no ad hoc solutions are required and combinations with any number of defensive techniques is straightforward. We demonstrate this on a synthetic example as well as on a practical problem of light selection in direct illumination computation.

Synthetic example Our simple example in the first row in Figure 2.3 shows a combination of the almost ideal technique p_2 with defensive techniques p_1 and p_3. We can see that while the balance and power heuristic combinations produce more variance than the p_2 technique alone, with the optimal weights the variance is actually decreased.

Light selection As we discussed in the previous chapter, MC estimation of direct illumination often contributes a significant amount of noise to the image. Recall that direct illumination is computed as an integral $F_{\mathrm{DI}} = \int_A L_{\mathrm{e}} BVG \, \mathrm{d}\mathbf{y}$ (we omitted arguments for brevity), where L_{e} is the emitted radiance, B the BRDF, V the visibility, G the geometry factor, and the domain A is the set of all emissive surfaces. A standard approach to design a direct illumination estimator is to first randomly select one light source according to a light selection distribution and then sample a point on the selected light. A good light selection technique would select a light proportionally to its actual contribution to the integral (and the nested estimator variance as we proved in the previous chapter). Unfortunately, this quantity cannot be computed analytically, especially because of the possibly complex visibility factor V.

In the previous chapter, we utilized the Bayesian regression to learn the actual light contribution from previous samples. We were able to robustly learn this quantity including the visibility but we omitted the BRDF factor B. This was motivated by practical considerations of a production renderer, where the BRDF can be defined by arbitrarily complex shaders, often given as a black-box. Instead, we combined our light selection technique with BRDF sampling using MIS. While this ensured good performance of our method even in the presence of glossy materials and large area light sources, the combination using the power heuristic could be suboptimal on diffuse surfaces. Applying the optimal weights instead might be therefore beneficial and further improve the method. However, our approach to estimating the optimal weights described in Section 2.6 do not allow probability densities of the used sampling techniques to change over time (i.e., to be learned online during rendering) as this would change the estimated linear system $\langle \mathbf{A} \rangle \langle \boldsymbol{\alpha} \rangle = \langle \mathbf{b} \rangle$. Therefore, application of the optimal weights in adaptive methods with online learning is not straightforward and we leave it for future work.

Instead, we demonstrate the optimal weights on an offline adaptive light selection technique implemented in PBRT [Pharr et al., 2016]. It divides the scene using a regular grid, estimates the unoccluded contribution of all lights in each of its cells using a dedicated set of samples, and then uses these estimates as the

[6] Using a single technique on its own is identical to a weighting strategy assigning unit weight to that technique and zero to all other techniques.

light selection probabilities. This technique is called *Spatial* within PBRT, we will call it *Trained* to emphasize its adaptive nature. It is close to optimal on unoccluded surfaces but causes significant noise in shadows and must be combined with a defensive *Uniform* light selection technique.

One example is given in Figure 2.1 in the Staircase I scene. We discussed it in the chapter introduction and will provide a further insight in Section 2.7.4. Another example is presented in Figure 2.8 which shows the results in the Staircase II scene lit by several small area light sources. The *Trained* technique performs well on unoccluded surfaces but produces more noise than the *Uniform* technique in shadows. Intuitively, we would like to combine both techniques in the shadows and use the *Trained* technique alone on the unoccluded surfaces. However, the false colour insets show that the power heuristic gives the uniform technique a positive weight everywhere, improving the performance in the shadows, and degrading the quality on the unoccluded surfaces. On the other hand, the optimal weights are zero or even negative on the unoccluded surfaces. As a result, the optimal weights maintain the good properties of both techniques everywhere and thus achieve 2.7× lower mean-squared error per sample than the power heuristic (and 3.2× lower than the balance heuristic, as shown by the results linked in Appendix 2.10.6).

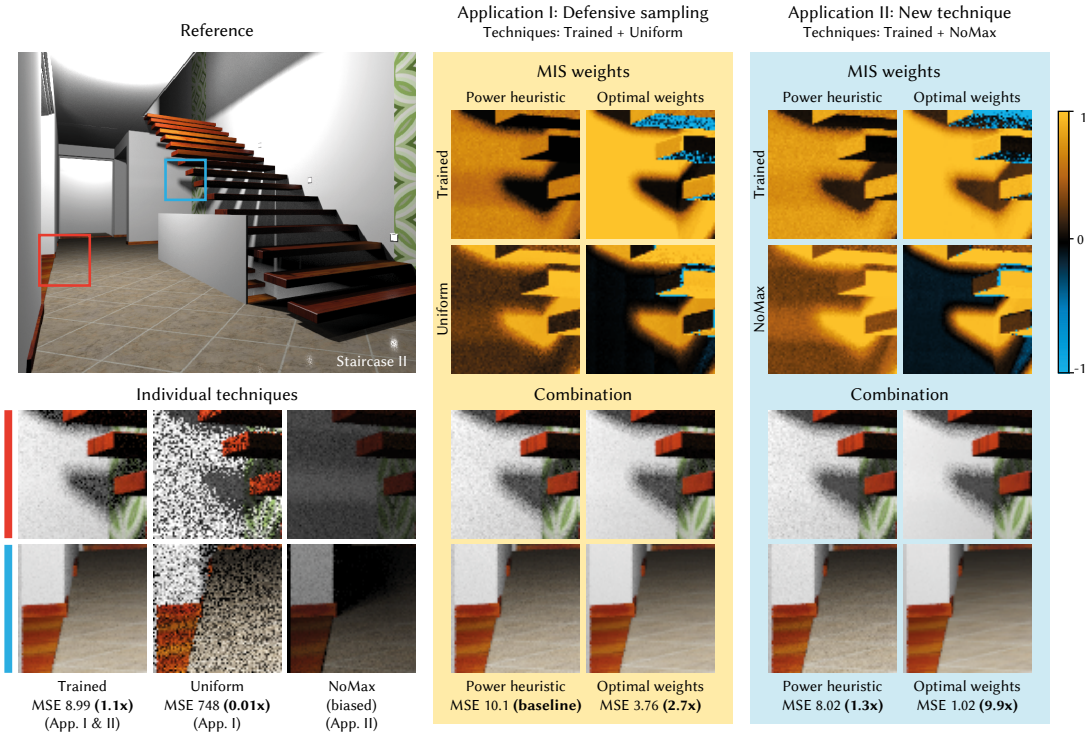


Figure 2.8: Equal-sample comparison (20 samples per technique per pixel) of different combination strategies for a trained light selection technique (*Trained*) and defensive techniques (*Uniform*, *NoMax*). In contrast to the power heuristic, the optimal MIS weights (via the Direct estimator) are never worse than any of the techniques alone. The false colour insets correspond to average weights per pixel for the three techniques. The MSE improvement in parentheses is with respect to the power heuristic combination of the *Trained* and *Uniform* techniques. All MSE values are $\times 10^{-4}$.

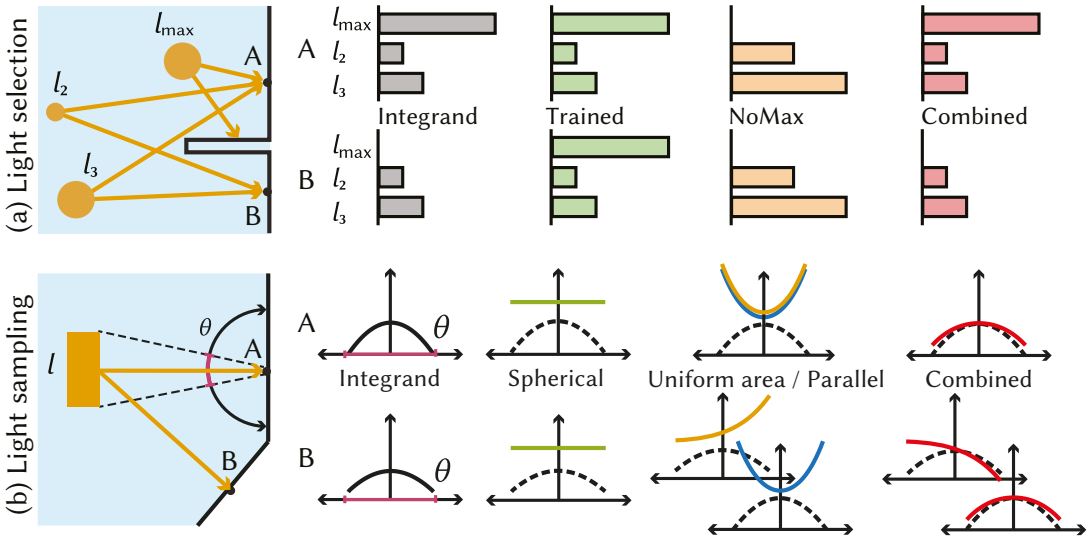


Figure 2.9: (a) Schematics illustrating the optimal combination of techniques *Trained* and *NoMax* for the light selection application, which can well approximate the integrand at both points A and B. (b) Schematics for the light sampling application illustrating the optimal combination of techniques *Spherical*, *Uniform area*, and *Parallel*. At point B, where the surface is not parallel to the light, the optimal combination *Spherical* + *Uniform area* approximates the integrand much worse, while the optimal combination *Spherical* + *Parallel* is still good. The displayed quantities are in the solid angle measure, their derivations can be found in Appendix 2.10.3.

2.7.4 Application II: Design of new sampling techniques

As discussed in Section 2.5, the optimal weights form a control variate as a linear combination of the sampling pdfs, i.e., as $\sum_i^N \alpha_i p_i$. We have shown that the closer the control variate approximates the integrand, the lower the variance. Introducing a new, properly designed technique (even a biased one!) can vastly expand the space of possibilities for the optimal weights to form a control variate closer to an integrand, and therefore can greatly improve the performance.

We first revisit the light selection problem for direct illumination computation from Section 2.7.3 and introduce a new technique that substantially lowers the variance. Then, we show new techniques that improve sampling of a single light.

New technique for light selection The *Trained* light selection technique from Section 2.7.3 neglects visibility. In shadows, the technique's pdf does not match the integrand well, and variance goes up.

We illustrate that in Figure 2.9a. For point A, the *Trained* technique (green) is a good fit to the integrand (gray), and performs well. For point B, however, the actual integrand has no contribution from the closest light due to occlusion, and there is a mismatch between the pdf of the *Trained* technique and the integrand itself.

To solve the issue at point B, we construct a new technique with a pdf that matches the integrand well specifically for that case. Then we leave it up to the optimal weights for a particular image pixel to decide which of the two cases has occurred (A or B), and to form the optimal control variate from pdfs of

both techniques. It is easy to construct such a technique from the pdf of the *Trained* technique: it is the same except it samples the strongest light with a zero probability. We call this technique *NoMax* (orange in Figure 2.9a).

We demonstrate that in the Staircase II scene (Figure 2.8). We see that using the *NoMax* technique alone causes a significant bias. But when optimally weighted with the *Trained* technique, it is much better than any other result in Figure 2.8. Note that the power heuristic is unable to create such a combination: it improves in shadows, but increases variance in the rest of the scene in comparison to *Trained* as well as to the power heuristic combination of *Trained* and *Uniform*. That gives the optimal weights 9.9× lower MSE per sample. Moreover, the optimal combination of the *Trained* and *NoMax* techniques improves 3.7× over the optimal combination of *Trained* and *Uniform*.

One special case, when the combination of the *Trained* technique and the *Uniform* technique works particularly well is when we have *exactly* two lights in the scene. We illustrate that on Staircase I scene in Figure 2.1. In that case a linear combination of the *Trained* and *Uniform* techniques can approximate virtually any distribution, which results in 9.6× lower MSE per sample than the power heuristic.

New techniques for light area sampling While light selection contributes most direct illumination variance in scenes with many small lights, careful sampling of the point on the light source becomes important in the presence of larger light sources. Figure 2.9b shows a schematic of a scene where a Lambertian area light source illuminates a point on a diffuse surface. The figure plots the sampling densities of various techniques over the part of the hemisphere that receives illumination, as well as the integrand itself (in black), which in this case becomes $L_\mathrm{e}G$, where L_e is the emitted radiance and G the geometry term. A typical technique is the uniform sampling of the light surface, we denote it *Uniform area* (Figure 2.9b, orange), but it is not a good approximation to the integrand as it neglects G. A better idea is to uniformly sample the light projection onto the unit sphere around the illuminated point [Arvo, 1995], and we call this technique *Spherical* (Figure 2.9b, green). It takes into account the geometric factor (except for the surface cosine) so it is closer to the integrand. But a linear combination of the *Uniform area* and *Spherical* techniques (shown in red), found by the optimal weights, performs even better. That is, as long as the light is parallel to the illuminated surface.

If the light is *not* parallel, the shape of the *Uniform area* technique deforms (see the point B in Figure 2.9b) and the optimal combination no longer matches the integrand. We now replace the *Uniform area* technique with a new one: uniform sampling of the light projection onto *a plane parallel to the surface*, denoted *Parallel* (Figure 2.9b, blue). Its pdf is similar to that of *Uniform area*, but does not depend on the light orientation. Therefore, the good match of the optimal combination is retained even at the point B.

We demonstrate these techniques in the Dining room scene (Figure 2.10) lit by one large area light from above. All images were rendered using the same *total* number of samples per pixel to see if any new technique can justify using an MIS combination instead of the *Spherical* technique alone. As expected, the *Spherical* technique alone generally performs better than the *Uniform area* and

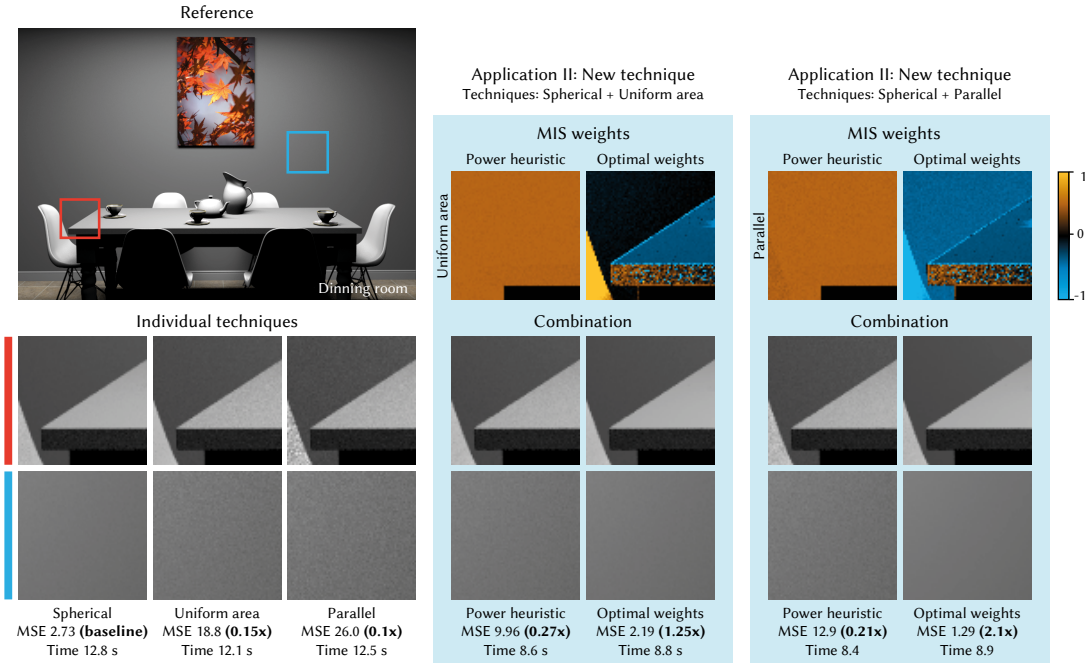


Figure 2.10: Equal-sample comparison (40 per pixel in total) of combinations of standard light sampling techniques (*Uniform area*, *Spherical*) and a new one (*Parallel*) motivated by properties of the optimal MIS weights. The combination with the new technique using the optimal weights performs best. The MSE improvement in parentheses is with respect to 40 samples from the *Spherical* technique alone. All MSE values are $\times 10^{-4}$. The false colour insets show weights of the *Uniform area* and *Parallel* techniques.

Parallel techniques. Therefore, their combination using the power heuristic will always be worse than relying only on the samples from *Spherical*. However, when they are combined using the optimal weights the result is much better. While the combination with *Uniform area* decreases variance mainly on the table, the combination with *Parallel* further improves the result also on surfaces not parallel to the light (e.g., the wall) and provides 2.1× lower MSE than the *Spherical* technique alone. Note the negative value of the optimal weights of the *Uniform area* and *Parallel* techniques in the improved regions.

Let us underline that the methods introduced in Section 2.7.4 are not meant to be ready for production use. They serve as a proof of concept showing that this approach to construction of sampling techniques has an interesting potential.

2.7.5 Additional results

Optimal weights for BRDF and light techniques. We investigated the behaviour of the optimal weights for an MIS combination of the light area and BRDF sampling techniques. For that we rendered the classic Veach's scene [Veach and Guibas, 1995]. Following Veach, we estimate illumination from individual lights separately, combining light area and BRDF sampling, and we add the contributions together. We combine the samples using the optimal weights and compare the result with the balance and power heuristics in Figure 2.11. In this setting, the power heuristic appears to be close to the optimum, but the optimal weights still slightly improve the result.

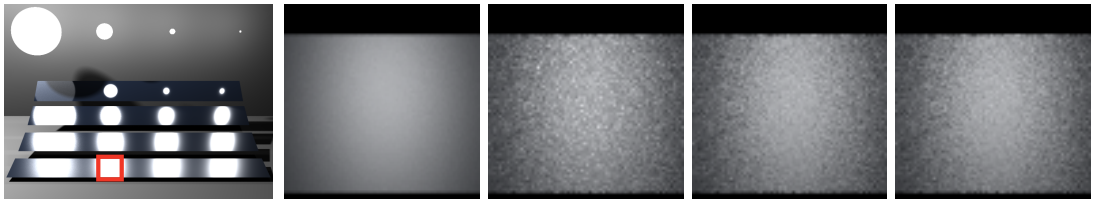


Figure 2.11: Equal-sample comparison of the optimal MIS weights with the balance and power heuristics in the classic light vs. BSDF sampling scenario in the Veach's scene. The MSE values (in parentheses, are $\times 10^{-4}$) are computed after 10 samples per light per technique per pixel.

Overhead We have so far focused on equal-sample comparisons to clearly show the effect of the combination strategies unaffected by the implementation. For the sake of completeness, equal-time comparisons are linked in Appendix 2.10.6 and summarized in Table 2.1. The overhead of the Direct estimator (caused mainly by the $\langle \mathbf{A} \rangle, \langle \mathbf{b} \rangle$ updates) is at most 10%, making the equal-sample MSE improvement close to the equal-time speedup. Note that when comparing to the *Spherical* technique in the Dining room scene the overhead is negative; sampling the perfect spherical projection is considerably more expensive than the other techniques.

Regarding memory overhead, we need to store estimates for the technique matrix for each pixel and estimates for the contribution vector for each pixel and colour channel, which in our cases meant storing $2^2 + 3 \cdot 2 = 10$ floats per pixel. When rendering the image by blocks, one pixel in a block at a time, the memory overhead is practically negligible.

	Staircase I Techniques: *Train* + *Uni* Baseline: Power *Train* + *Uni*			**Staircase II** Techniques: *Train* + *M* Baseline: Power *Train* + *Uni* / *Train* + *M*		
	Equal-time speedup	Equal-sample improvement	Overhead	Equal-time speedup	Equal-sample improvement	Overhead
Direct	8.89	9.56	6.20%	8.86 / 7.53	9.90 / 7.83	9.93% / 2.54%
Progressive $U = 1$	3.01	4.37	33.02%	5.25 / 4.46	6.68 / 5.29	35.32% / 26.23%
Progressive $U = 2$	2.76	3.42	19.32%	4.81 / 4.09	5.35 / 4.23	24.07% / 15.73%
Progressive $U = 4$	2.03	2.33	12.44%	3.82 / 3.25	3.90 / 3.09	17.64% / 9.73%
	Veach Techniques: *BSDF* + *Light* Baseline: Power *BSDF* + *Light*			**Dining room** Techniques: *Par* + *Sp* Baseline: *Sp* / Power *Par* + *Sp*		
	Equal-time speedup	Equal-sample improvement	Overhead	Equal-time speedup	Equal-sample improvement	Overhead
Direct	1.02	1.02	5.02%	3.40 / 9.99	2.12 / 10.05	-30.53% / 5.94%
Progressive $U = 1$	0.77	1.03	38.24%	1.87 / 5.48	1.27 / 6.00	-12.17% / 33.92%
Progressive $U = 2$	0.86	1.04	20.88%	1.87 / 4.92	1.03 / 4.88	-20.43% / 21.33%
Progressive $U = 4$	0.94	1.03	14.71%	1.50 / 4.40	0.74 / 3.50	-26.09% / 12.70%

Legend: *Train* = *Trained*, *Uni* = *Uniform*, *M* = *NoMax*, *Par* = *Parallel*, and *Sp* = *Spherical*

Table 2.1: Performance statistics of the Direct and Progressive estimators, the latter with different values of the update step U (Section 2.6.3). Speedup and equal-sample improvement are ratios of the mean-squared error. The overhead is the relative increase of the rendering time with the same total number of samples. The baseline for these values is the power heuristic combination, except for the Dining room which also compares to using the spherical projection sampling alone. Corresponding images are linked in Appendix 2.10.6.

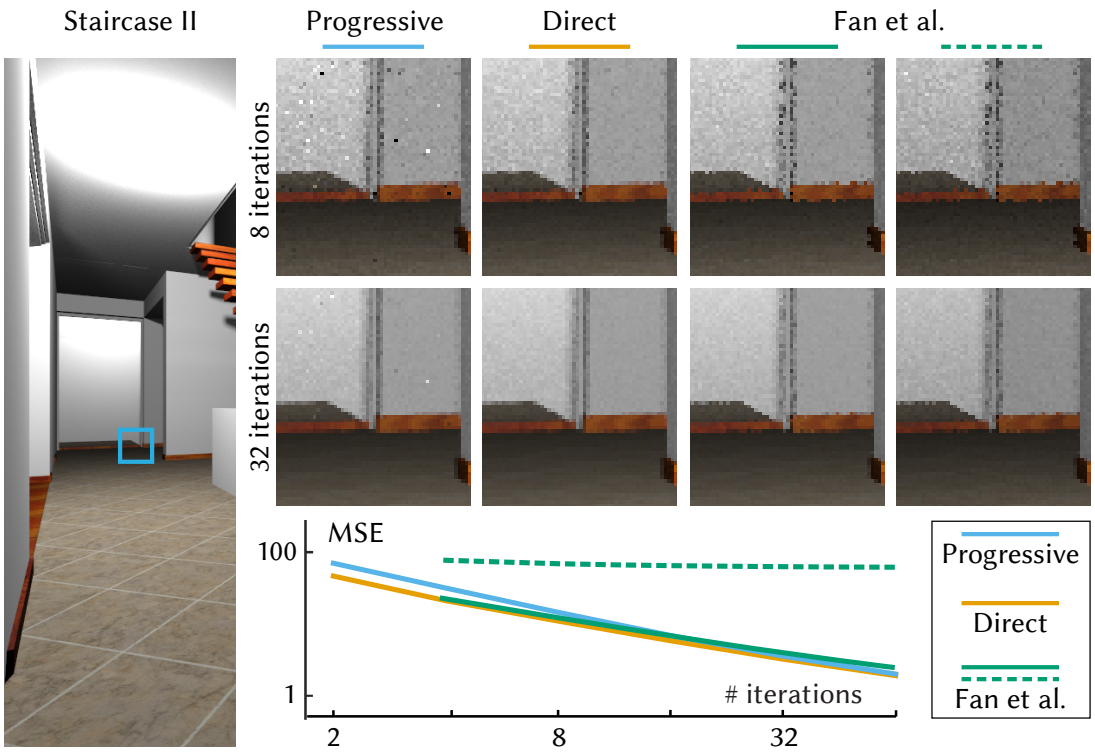


Figure 2.12: The insets along with MSE plots for the Staircase II scene rendered with an increasing number of samples with the Direct and Progressive estimators and the method of Fan et al. with either the *Uniform* (solid) or the *Trained* (dashed) technique skipped. See Section 2.7.5 for details.

Direct vs. Progressive estimators All our results shown in Section 2.7.3 and Section 2.7.4 were obtained by the Direct estimator. Its bias and variance with respect to the Progressive estimator could be a concern. We link both their equal-time and equal-sample comparisons in Appendix 2.10.6 with a summary in Table 2.1. In agreement with our synthetic tests from Section 2.6, the equal-sample MSE improvement of the Progressive estimator is always smaller (about 30%-40%), except for the Veach's scene, where both estimators perform equally. In Figure 2.12, we show insets and MSE plots of the renderings using an increasing number of samples per technique (from 2 to 64) in the Staircase II scene. The Progressive estimator (blue) is unbiased but gains a spiky noise in the initial iterations, from which it takes long to recover. The Direct estimator (yellow) is biased only for a low number of samples (<16) and practically unbiased afterwards, which is also in line with our synthetic tests.

As expected, the overhead of the Progressive estimator is higher than the Direct one because of the repeated solving of the linear system. As the update step U increases (Section 2.6.3), the overhead decreases from almost 40% for $U = 1$ to 15% for $U = 4$. But since the equal-sample MSE improvement also decreases, the equal-time speedup is actually worse as well. The best compromise seems to be using $U = 2$, yielding up to 5× speedup in our scenes.

Comparison to Fan et al. In Figure 2.12 we compare our approach to Fan et al. [2006], who adopted the approach by Owen and Zhou for rendering. They estimate $\boldsymbol{\alpha}$ by solving a truncated system obtained by skipping regressors corre-

sponding to a particular sampling technique from the data matrix. For a particular skipped technique their method is the same as our biased Direct estimator, except for two differences: First, they do not perform the estimation per pixel but by averaging per point estimates computed from fixed-sized batches, which makes their method not consistent. Second, they introduce a regularization strategy which can decrease variance at the cost of increased bias. For clarity, we provide pseudocode of our adaptation of their method in Appendix 2.10.5.

We set the batch size in their method to 8 samples (the same total number of samples as 4 iterations of our method) and rendered the Staircase II scene with an increasing number of batches. The green lines in the plot show their method when skipping the *Uniform* (solid) and *Trained* (dashed) technique, respectively. When the *Uniform* technique is skipped, their method behaves similarly to ours, and their regularization slightly reduces the noise in some parts of the image. When the *Trained* technique is skipped, the substantial bias of their method due to the computation in batches is further amplified by their regularization approach, resulting in a visibly darker image. As the performance of their method depends on a skipped technique, it might be difficult to predict the optimal technique for skipping for a given integration problem. Without the regularization, their method produces identical results to our Direct estimator for any technique skipped, but only for the first batch (with increasing number of batches the bias in their method does not diminish).

2.8 Limitations and future work

Overhead While we believe that a derivation of optimal MIS weights is an important theoretical result, their application in practice is more complicated than for the traditional balance or power heuristics. Estimation and solution of the linear system results in computational overhead that grows super-linearly with the number of combined techniques. While the overhead in our tests was modest, especially for the Direct estimator, this could become an issue as the number of sampling techniques increases.

Applications Our rendering applications provide a proof of concept, but are far from being production-ready and leave space for further investigation. An obvious next step would be to integrate the optimal weights into a full global illumination solution. One interesting direction is the optimal combination of sampling techniques in bidirectional path tracing and derived methods [Veach and Guibas, 1995; Georgiev et al., 2012b; Hachisuka et al., 2012], though handling the relatively high number of available sampling techniques could be challenging. Another class of algorithms that could greatly benefit from the optimal MIS weights is path guiding [Vorba et al., 2014; Herholz et al., 2016; Müller et al., 2017], where the necessity for defensive sampling limits the achievable improvements. However, since our approach to estimating the optimal weights do not allow the used sampling techniques to change over time, application to these algorithms will not be straightforward and the estimation will have to be adjusted (e.g., by interleaving updates of sampling distributions and the linear system).

New techniques We showed that the optimal weights motivate the design of new sampling techniques. We presented two new techniques which yield more efficient estimators when combined using the optimal weights but these were just the most obvious simple examples. We believe there is much more to explore in this direction.

The MIS framework A serious limitation of the MIS framework itself is its somewhat wasteful approach: samples are first taken but the contribution of many of them may be weighted almost to zero. Our optimal weights do not address this issue. More work is needed on optimizing the sample counts for different techniques (and whether or not some techniques should be included in the mix at all), while maintaining the estimator's robustness.

2.9 Conclusion

In this chapter, we focused on decreasing the variance of MC integration in rendering by improving the combination of sampling techniques. We presented optimal weighting functions for the multi-sample model of multiple importance sampling. In deriving the optimal weights, we pointed out, for the first time, an unnecessary assumption on the non-negativity of weighting functions underpinning the previous claims concerning variance bounds for the balance heuristic. We showed that this assumption effectively prohibited exploration of an entire class of efficient combination strategies, amongst them the optimal one.

We showed the connection of the optimal weights to control variates, which yields interesting observations on the relation of variance of the optimal weights and balance heuristic. In particular, the optimal weights are a good choice for defensive sampling, where the balance heuristic is particularly inefficient. Our proof of concept applications in direct illumination estimation showed that new sampling strategies motivated by the variance properties of the optimal weights yield further benefits. We believe that our work opens up new directions for improving efficiency of combined estimators.

2.10 Appendix

2.10.1 Calculus of variations

Our derivation of the optimal weights relies on the *calculus of variations* [Aubert and Kornprobst, 2006], the basic elements of which we now informally review. It is typically used to find extrema of a *functional* – a mapping from some space of functions Ω onto real numbers. In our case, the functional of interest – the variance – conforms to a general form $\mathbf{F}(h) = \int \hat{F}(h(x))\,\mathrm{d}x$, where $h \in \Omega$ is a function (in our case the weights) and $\hat{F}$ is some operation on h.

A basic tool used to locate extrema of a functional $\mathbf{F}$ is its *functional derivative* $\frac{\partial \mathbf{F}}{\partial h}$, i.e., the rate of change of $\mathbf{F}$ with infinitesimally small perturbations of the function h. Similar to classic calculus, the extrema are given by the function(s) h for which the functional derivative equals to zero, i.e., ${}^{\partial \mathbf{F}}/_{\partial h}(x) = 0$.

Calculation of the functional derivative can be transformed to classic differentiation from 'ordinary' calculus using the relation

$$\left\langle \frac{\partial \mathbf{F}}{\partial h}, \delta \right\rangle = \frac{d}{d\varepsilon}\Big|_{\varepsilon=0} \mathbf{F}(h + \varepsilon\delta), \tag{2.32}$$

where $\delta \in \Omega$ is a variation (a function), while $\varepsilon \in \mathbb{R}$ is a number. To obtain the functional derivative, we 1) replace any occurrence of h in the functional by $h+\varepsilon\delta$, 2) take derivative with respect to ε, 3) set $\varepsilon = 0$. This yields an expression that is, by the relation (2.32), equal to the inner product of the variation δ and the functional derivative $\partial\mathbf{F}/\partial h$ that we seek to find, i.e., to the integral $\int_D \frac{\partial \mathbf{F}}{\partial h}\, \delta \, \mathrm{d}x$. The last step is therefore to extract the part of the expression corresponding to the functional derivative.

As in classic calculus, Lagrange multipliers can be used to handle *constraints*. To find extrema of $\mathbf{F}$ satisfying a constraint $g(h(x))=0$, we formulate a constraint functional $\mathbf{G}(h) = \int \lambda(x) g(h(x))\, \mathrm{d}x$, where $\lambda \in \Omega$ is the Lagrange multiplier. We then locate extrema of the *Lagrangian* $\mathbf{L}(h, \lambda) = \mathbf{F}(h) - \mathbf{G}(h, \lambda)$ both in terms of h and λ.

2.10.2 Proof of the relationship (2.25)

The variance of the optimal estimator (2.23) can be expanded as

$$\mathrm{Var}[\langle F \rangle^o] = \mathrm{Var}[\langle F \rangle^b] + \mathrm{Var}[\langle G \rangle^b] - 2\mathrm{Cov}[\langle F \rangle^b, \langle G \rangle^b]. \tag{2.33}$$

We now express the variance $\mathrm{Var}[\langle G \rangle^b]$ and covariance $\mathrm{Cov}[\langle F \rangle^b, \langle G \rangle^b]$ from (2.33) in terms of the technique matrix, contribution vector and $\boldsymbol{\alpha}$. Using the shorthand notation $q = (\sum_i^N n_i p_i)^{-1}$ and dropping the function arguments, we obtain

$$\mathrm{Var}[\langle G \rangle^b] = \int_D q \, \Big(\sum_i^N \alpha_i p_i \Big)^2 \mathrm{d}x - \sum_i^N n_i \Big(\int_D q \; p_i \sum_{j=1}^N \alpha_j p_j \, \mathrm{d}x \Big)^2, \tag{2.34}$$

Because the elements of $\mathbf{A}$ are given by $a_{ik} = \langle p_i, \; p_k \, q \rangle$, we can rewrite the first term in (2.34) as:

$$\textstyle\sum_{i=1}^N \sum_{k=1}^N \alpha_i a_{ik} \alpha_k = \boldsymbol{\alpha}^\intercal \mathbf{A} \boldsymbol{\alpha}. \tag{2.35}$$

The second term in (2.34) can be transformed in a similar fashion:

$$\sum_{j=1}^N n_j \left(\textstyle\sum_{i=1}^N \alpha_i a_{ij} \right) \left(\textstyle\sum_{k=1}^N a_{jk} \alpha_k \right) = \boldsymbol{\alpha}^\intercal \mathbf{A} \mathbf{N} \mathbf{A} \boldsymbol{\alpha}, \tag{2.36}$$

with $\mathbf{N}$ being a diagonal $N \times N$ matrix with the sample count n_i along the diagonal. Putting together (2.35), (2.36), and factoring out $\boldsymbol{\alpha}$, we obtain

$$\mathrm{Var}[\langle G \rangle^b] = \boldsymbol{\alpha}^\intercal \left(\mathbf{A} - \mathbf{A}\mathbf{N}\mathbf{A} \right) \boldsymbol{\alpha}. \tag{2.37}$$

Now, we express the covariance $\mathrm{Cov}[\langle F \rangle^b, \langle G \rangle^b]$. Denoting $\langle F \rangle^b_{ij}$ and $\langle G \rangle^b_{ij}$ the parts of the MIS estimators for i-th technique and j-th independent sample, the covariance becomes

$$\mathrm{Cov}[\langle F \rangle^b, \langle G \rangle^b] = \sum_i^N n_i \mathrm{Cov}[\langle F \rangle^b_{i1}, \langle G \rangle^b_{i1}]. \tag{2.38}$$

That is because $\langle F \rangle_{ij}^b$ and $\langle G \rangle_{kl}^b$ are independent whenever $i \neq k$ and $j \neq l$, and thus their covariance is zero. Again, using $q = (\sum_i^N n_i p_i)^{-1}$, the relation (2.38) can be further expanded

$$\begin{aligned}\sum_i^N n_i \mathrm{Cov}[\langle F \rangle_{i1}^b, \langle G \rangle_{i1}^b] = \int_D q\ f \sum_{i=1}^N \alpha_i p_i \,\mathrm{d}x\ - \\ - \sum_i^N n_i \left(\int_D q\ p_i f \,\mathrm{d}x \right) \left(\int_D q\ p_i \sum_{j=1}^N \alpha_j p_j \,\mathrm{d}x \right).\end{aligned} \tag{2.39}$$

The first term in (2.39) equals to $\mathbf{b}^\intercal \boldsymbol{\alpha}$ where $\mathbf{b}$ is the contribution vector. The second term could be expanded as:

$$\sum_i^N n_i b_i \left(\sum_{k=1}^N a_{ik} \alpha_k \right) = \mathbf{b}^\intercal \mathbf{N} \mathbf{A} \boldsymbol{\alpha}. \tag{2.40}$$

Subtracting (2.40) from $\mathbf{b}^\intercal \boldsymbol{\alpha}$ yields the desired relation for the covariance:

$$\mathrm{Cov}[\langle F \rangle^b, \langle G \rangle^b] = \mathbf{b}^\intercal (\mathbf{I} - \mathbf{N}\mathbf{A}) \boldsymbol{\alpha}. \tag{2.41}$$

Finally, expanding (2.33) using the relationships (2.37) and (2.41), we obtain:

$$\mathrm{Var}[\langle R \rangle^b] = \mathrm{Var}[\langle F \rangle^b] + \boldsymbol{\alpha}^\intercal \left(\mathbf{A} - \mathbf{A}\mathbf{N}\mathbf{A} \right) \boldsymbol{\alpha} - 2\mathbf{b}^\intercal (\mathbf{I} - \mathbf{N}\mathbf{A}) \boldsymbol{\alpha}. \tag{2.42}$$

By using $\mathbf{b}^\intercal = \boldsymbol{\alpha}^\intercal \mathbf{A}$ and simplifying, we obtain the desired relationship (2.25).

2.10.3 Light sampling techniques formulas

In Section 2.7.4 and Figure 2.9b we discuss different light sampling techniques. Here we provide a derivation of the quantities illustrated in Figure 2.9b. If expressed in the solid angle measure, the integrand and probability density functions of the techniques read:

$$\begin{aligned} f(\theta) &= L_\mathrm{e} \cos\theta \propto \cos\theta \\ p_\mathrm{Spherical}(\theta) &= \frac{1}{|A_\mathrm{Spherical}|} \propto 1 \\ p_\mathrm{UniformArea}(\theta) &= \frac{1}{|A_\mathrm{UniformArea}|} \frac{d(\theta)^2}{\cos l(\theta)} \\ &= \frac{1}{|A_\mathrm{UniformArea}|} \frac{d_\perp^2}{\cos^3 l(\theta)} \propto \frac{1}{\cos^3 l(\theta)} \\ p_\mathrm{Parallel}(\theta) &= \frac{1}{|A_\mathrm{Parallel}|} \frac{d(\theta)^2}{\cos l(\theta)} = \frac{1}{|A_\mathrm{Parallel}|} \frac{d_\perp^2}{\cos^3 l(\theta)} \\ &\propto \frac{1}{\cos^3 l(\theta)} = \frac{1}{\cos^3 \theta} \end{aligned} \tag{2.43}$$

The quantities used in the formulas are shown in Figure 2.13. As discussed in Section 2.7.4, linear combination of the *Uniform area* and *Spherical* techniques is a good approximation for the integrand as long as the lit surface is parallel to the light source. For that case it holds $\cos^{-3}\theta = \cos^{-3} l(\theta)$, but that relation breaks for points on differently oriented surfaces, and the linear combination

of the *Uniform area* and *Spherical* techniques on such surfaces can no longer approximate the integrand well.

The above problem does not occur with the *Parallel* technique, which first projects the light onto a plane parallel to the shaded surface, and then samples that projection. Therefore, a linear combination of its sampling density ($\propto \cos^{-3}\theta$) with the *Spherical* technique ($\propto 1$) better approximates f irrespective of the light orientation.

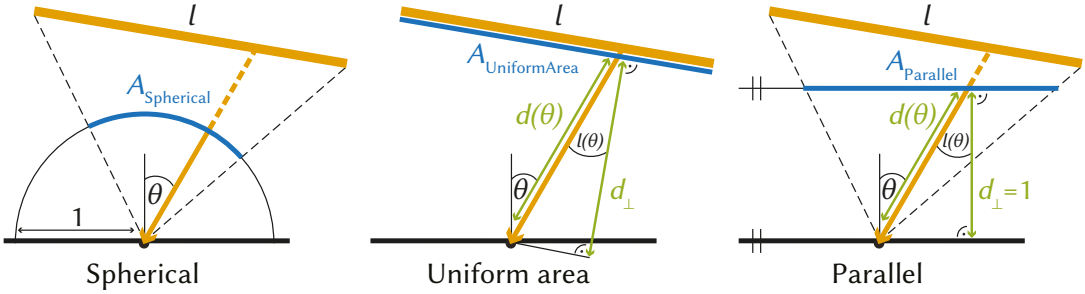


Figure 2.13: Illustration of the quantities used in formulas in Appendix 2.10.3. A denotes surface area of the sampled light/projection, θ angle at the surface, $l(\theta)$ angle at the light/projection, $d(\theta)$ distance between the point on the surface and on the light/projection, $d_\perp$ perpendicular distance of the light/projection.

2.10.4 Relationship to Owen and Zhou

Approximating $\boldsymbol{\alpha}$ in (2.23) can be viewed as a regression problem, as Owen and Zhou [2000] did. To explain their approach, we denote parts of (2.23) using the following notation

$$f_{ij} = f(X_{ij})/p_{\mathbf{c}}(X_{ij}), \qquad d_{ijk} = p_k(X_{ij})/p_{\mathbf{c}}(X_{ij}), \tag{2.44}$$

where $p_{\mathbf{c}}(x) = \sum_{k=1}^{N} n_k p_k(x)/M$ and $M = \sum_{k=1}^{N} n_k$. Let us uniquely map an index pair $(i,j), i = 1,\dots,N, \quad j = 1,\dots,n_i$ to an index $l = 1,\dots,M$ and denote quantities from (2.44) as f_l and d_{lk} in the following text. Owen and Zhou approximate the optimal coefficients $\boldsymbol{\alpha}$ by multiple linear regression of observations f_l on regressors d_{lk} along with an intercept term $\langle\alpha_0\rangle$, i.e.,

$$\langle\alpha_0\rangle + \textstyle\sum_{k=1}^{N}\langle\alpha_k\rangle d_{lk} \approx f_l, \quad l = 1,\dots,M. \tag{2.45}$$

In matrix form,

$$\mathbf{D}\mathbf{h} \approx \mathbf{f}, \tag{2.46}$$

where each row corresponds to (2.45) for a particular index l. Therefore $\mathbf{D}$ is a matrix $M \times (N+1)$ with the first column composed of ones and the $(k+1)$ column being $(d_{1k},\dots,d_{Mk})^\intercal$, $\mathbf{f}$ is a column vector $(f_1,\dots,f_M)^\intercal$, and $\mathbf{h}$ is a column vector of length $N+1$ representing the terms $\langle\alpha_0\rangle$ and $\langle\alpha_k\rangle, k = 1,\dots,N$. Note that the above regression problem can be composed from several MIS sample batches by concatenating the corresponding matrices and vectors.

To solve the regression problem (2.46), Owen and Zhou minimize $\|\mathbf{D}\mathbf{h} - \mathbf{f}\|_2^2$ in terms of $\mathbf{h}$, which leads to the *normal equation* for $\mathbf{h}$

$$\mathbf{D}^\intercal\mathbf{D}\mathbf{h} = \mathbf{D}^\intercal\mathbf{f}. \tag{2.47}$$

The above equation is singular, because the first column of ones in $\mathbf{D}$ is a linear combination of the others, i.e., $\sum_{k=1}^{N} n_k d_{lk}/M = 1, l = 1, \ldots, M$. Let $\mathbf{h}_0$ be a solution of (2.47), and $\mathbf{v} = (-M, n_1, \ldots, n_N)^\intercal \in \text{Null}(\mathbf{D}^\intercal\mathbf{D})$. Then each $\mathbf{h} \in \{\mathbf{h}_0 + s\mathbf{v} | s \in \mathbb{R}\}$ solves (2.47). Because the sum of elements of $\mathbf{v}$ equals 0, it holds for all $\mathbf{h}$ that the sum of their elements equals *the same* number, and we show in the next paragraph that it must be an estimate of the integral F. We also show that an alpha estimator extracted from any $\mathbf{h}$ estimates some $\tilde{\boldsymbol{\alpha}}$, which belongs to the full solution for alphas (see Section 2.4.2).

We can find a solution to (2.47) by SVD applied directly (preferred by Owen and Zhou), but we can also solve a *truncated* system $\hat{\mathbf{D}}^\intercal\hat{\mathbf{D}}\hat{\mathbf{h}} = \hat{\mathbf{D}}^\intercal\mathbf{f}$, where $\hat{\mathbf{D}}$ is obtained by dropping one column from $\mathbf{D}$. That yields a truncated solution vector $\hat{\mathbf{h}}$, and it is equivalent to finding a solution $\mathbf{h}$ which has the element corresponding to the skipped column equal to zero. Therefore, summing up the elements of such a truncated vector gives the same estimate of F. Dropping the first column from $\mathbf{D}$ related to $\langle\alpha_0\rangle$ *makes the truncated system even the same (up to a scaling factor) as our system estimated by the balance heuristic* (2.28) described in Section 2.6.1, because then

$$\hat{\mathbf{D}}^\intercal\hat{\mathbf{D}} = M^2\langle\mathbf{A}\rangle, \quad \text{and} \quad \hat{\mathbf{D}}^\intercal\mathbf{f} = M^2\langle\mathbf{b}\rangle. \tag{2.48}$$

The truncated vector $\hat{\mathbf{h}}$ solving such a system is then equal to the $\langle\boldsymbol{\alpha}\rangle$ estimate described in Section 2.6.2, and there exists an $\mathbf{h}_0$, with the first component equal to zero, corresponding to such a truncated vector. Therefore, using $\mathbf{n} = (n_1, \ldots, n_N)^\intercal$, an alpha estimate represented by $\mathbf{h} = \mathbf{h}_0 + s\mathbf{v}, s \in \mathbb{R}$ equals to $\langle\boldsymbol{\alpha}\rangle + s\mathbf{n}$, which is an estimate of $\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha} + s\mathbf{n}$ from the full solution for alphas. It follows that the sum of elements of any such $\mathbf{h}$ must be equal to the sum of elements of $\langle\boldsymbol{\alpha}\rangle$ and therefore it is an estimator of F. In other words, the solutions given by Owen and Zhou's approach are equivalent to the solution of the system from THEOREM 1 as long as the system parts $\mathbf{A}$ and $\mathbf{b}$ are estimated by the balance heuristic. Our result is more general, and it suggests the existence of some alternative strategies how to approximate $\mathbf{A}$, $\mathbf{b}$, and $\boldsymbol{\alpha}$.

2.10.5 Pseudocode of Fan et al.

Figure 2.14 presents pseudocode of our adaptation of the method by Fan et al. [2006] who modified Owen and Zhou's approach by using regularization and applied it in rendering (see the previous section for details of Owen and Zhou's method). The computation is performed in batches. In each batch, n_i samples are drawn from each of the N sampling techniques $p_i, i = 1, \ldots, N$ (we use $N = 2, n_1 = n_2 = 4$, each batch therefore consists of $M = 8$ samples, the same total number of samples as 4 iterations of our Direct estimator). For each sample one row of the data matrix $\mathbf{D}$ and vector $\mathbf{f}$ is computed according to (2.44). After all samples in one batch are processed, one column of $\mathbf{D}$, corresponding to $\langle\alpha_k\rangle, k = 1, \ldots, N$, is dropped. Then the regularized truncated system $(\hat{\mathbf{D}}^\intercal\hat{\mathbf{D}} + \lambda\mathbf{I})\hat{\mathbf{h}} = \hat{\mathbf{D}}^\intercal\mathbf{f}$ is solved, where $\mathbf{I}$ is the identity matrix and λ is the weight of the regularization (we use $\lambda = 1$ as suggested by Fan et al.). Finally, the sum of the elements of the solution $\hat{\mathbf{h}}$ is added to the final result and the algorithm proceeds to the next batch.

Note that in practice this algorithm can be implemented to directly compute $\hat{\mathbf{D}}^\intercal\hat{\mathbf{D}}$ and $\hat{\mathbf{D}}^\intercal\mathbf{f}$ instead of first computing $\hat{\mathbf{D}}$ and $\mathbf{f}$ and then multiplying by $\hat{\mathbf{D}}^\intercal$. Such an implementation has the same computational complexity but smaller memory requirements. Fan et al. do not mention this optimization but our implementation of this algorithm applies it. This optimized implementation is included in the provided source code.

ALGORITHM 3: Fan et al.

$M \leftarrow \sum_i^N n_i$; // batch size
$result \leftarrow 0$;
for $batch \leftarrow 1$ **to** $maxBatches$ **do**
  $\mathbf{D} \leftarrow 0^{M\times(N+1)}$; $\mathbf{f} \leftarrow 0^{M\times 1}$; $l \leftarrow 0$;
  for $i \leftarrow 1$ **to** N **do**
    for $j \leftarrow 1$ **to** n_i **do**
      $X_{ij} \leftarrow$ draw j-th sample from technique p_i;
      $l \leftarrow l + 1$;
      $\mathbf{D}_{l0} \leftarrow 1$; // intercept term
      for $k \leftarrow 1$ **to** N **do**
        $\mathbf{D}_{lk} \leftarrow d_{ijk}$; // (2.44)
      end
      $\mathbf{f}_l \leftarrow f_{ij}$; // (2.44)
    end
  end
  $\hat{\mathbf{D}} \leftarrow$ drop one column of $\mathbf{D}$;
  $\hat{\mathbf{h}} \leftarrow$ solve regularized linear system $(\hat{\mathbf{D}}^\intercal\hat{\mathbf{D}} + \lambda\mathbf{I})\hat{\mathbf{h}} = \hat{\mathbf{D}}^\intercal\mathbf{f}$;
  $result \leftarrow result + \sum_i^N \hat{\mathbf{h}}_i$
end
return $result/maxBatches$

Figure 2.14: A pseudocode of the method of Fan et al.

2.10.6 Additional materials

We provide additional results for the presented scenes[7] online at

`https://cgg.mff.cuni.cz/~jaroslav/papers/2019-optimal-mis`

These include full-size images and rendering statistics for both equal-time and equal-sample comparisons of different combination strategies and sampling techniques described in this chapter.

Furthermore, the same web page offers downloading a ZIP archive containing 2 folders: `figure2` and `implementation`.

The first folder, `figure2`, contains a Mathematica notebook used for producing Figure 2.15, i.e., an extended version of Figure 2.3 with additional results for the cutoff and maximum heuristics. Running the notebook requires the Mathematica software (version 11.0 or newer)[8] with the MaTex package (version 1.7.4 or newer)[9]. Label positions in the produced image were tweaked manually.

The second folder, `implementation`, contains our source code of the optimal MIS weights. We implemented the optimal MIS weights as a new integrator called *optmis* in pbrt-v3[10]. It computes direct illumination only, and it has several parameters (described in `implementation/Params.html`) for specifying light selection/light sampling techniques, combination strategies, switching to the method of Fan et al. [2006], etc.

[7] Based on scenes created by Benedikt Bitterli (available at `https://benedikt-bitterli.me/resources/`).

[8] `https://www.wolfram.com/mathematica/`

[9] `https://library.wolfram.com/infocenter/MathSource/9355/`

[10] `https://www.pbrt.org/`

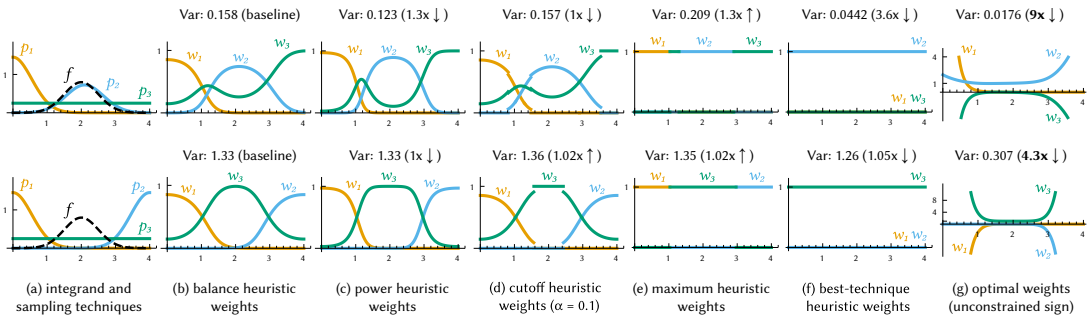


Figure 2.15: An extended version of Figure 2.3 with cutoff (d) and maximum (e) heuristic weights included. (a) depicts an integration problem where the integral of a function f is estimated via MIS. Three sampling techniques, p_1, p_2, and p_3, are used, and one sample is taken from each. The two rows differ solely in the sampling technique p_2: while p_2 closely matches f in the first row, in the second row it is fairly different. (b)–(e) plot, respectively, the balance, power, cutoff, and maximum heuristic weights as defined by Veach [1997]. (f) and (g) depict, respectively, the best-technique heuristic and the optimal weights as defined in this chapter.

3. Pre-computation

In this thesis, we focus on improving the performance of MC rendering. In the previous two chapters, we did so by decreasing the variance of the MC estimators, either by finding a single better sampling technique (Chapter 1) or by finding a better combination of multiple techniques (Chapter 2). In this chapter, we present a rather different approach. Sometimes it is more practical to pre-compute difficult parts of light transport, thus excluding their high computational cost from rendering completely. This is especially useful, if the pre-computation can be done once and then used repeatedly in different settings, scenes or even in different renderers. A particularly good example is rendering of the sky.

For high quality renderings of outdoor scenes, one needs realistic models of sky dome radiance, atmospheric scattering, and optionally also cloudscapes. It has been known for a considerable time how to compute these, via brute force path tracing of realistic atmospheric and cloud models. However, the computational cost of this is still infeasible for routine production use, and will remain so for the foreseeable future – in particular for interactive use cases. Three families of techniques have established themselves to cover for this performance deficiency of full atmospheric path tracing:

1. *HDR sky dome captures* are intrinsically realistic, and can include clouds: typical usage is as an HDR environment map. But they lack matching atmospheric scattering information, are static, and cannot easily be manipulated to, e.g., modify solar elevation or atmospheric parameters.

2. *Approximative sky models* such as [Hillaire, 2020] provide excellent results for interactive settings. But all techniques in this category are based on approximating light transport in the atmosphere and typically neglect higher order scattering events.

3. *Fitted analytical models of sky dome radiance* that are based on brute force simulations of atmospheric light transport have proven popular for use in offline rendering. Due to being based on physical simulations, such models can, at least in theory, deliver visual fidelity on par with HDR captures: but with the added ability to modify sky appearance and solar position.

In spite of the impressive performance of current interactive techniques, they cannot be used if high degrees of accuracy are required, due to their use of simplified light transport, and due to performing computations in a colour space instead of using spectral rendering. For several application areas, such as movie VFX, or predictive rendering applications like training of autonomous vehicle sensor software, one needs more realistic models: and especially for the latter, reliable spectral data for a wide range of configurations is also needed. As brute force approaches are too slow, fitted models will remain in use: but improvements in this area are needed, as existing techniques are all lacking a number of crucial features.

Current fitted models usually provide sky radiance data only for a ground-based observer and if they do support higher observer altitudes, they are missing matching finite distance in-scattered radiance and transmittance models. All

these components are necessary for realistic rendering of distant geometry, e.g., in views looking down on an aircraft in flight or mountains receding into the distance. Without them, a slow full atmospheric simulation has to be carried out. Furthermore, existing models neglect sky polarisation which is important for accurate rendering of specular objects in outdoor scenes, a feature needed for example for autonomous vehicle sensor training.

3.1 Prague Sky Model

In this chapter, we review the *Prague Sky Model* [Wilkie et al., 2021] which advances the state of the art of the fitted models. It is an integrated model of clear sky radiance and attenuation which follows the general approach of previous fitted models by first running brute force simulations, and then fitting a model to the obtained data. However, it improves on practically every component of this process. Its main benefits are:

- The use of reference data from atmospheric science to define realistic vertical scatterer distribution profiles: these profiles are then used in a polarisation-aware path tracer to generate a large database of polarised reference images.
- Fully spherical reference images, which are generated for a range of observer altitudes up to 15 km: this is a considerable improvement over current hemispherical models that are only defined for ground-based observers.
- Verification of these reference images against the output of dedicated atmospheric simulation software.
- A new tensor decomposition approach to compress the reference image dataset. With it, artefact-free interpolated sky dome images can be reconstructed from a coefficient set that is a fraction of the size of the reference images themselves.
- Solar elevations down to $-4.2°$ are included in the model, as the new compression technique is powerful enough to deal with the changing sky dome radiance patterns after sunset.
- A matching model for atmospheric transmittance is provided for describing attenuation in the atmosphere. Also allows computation of finite distance in-scattered radiance.
- And finally, there is also a matching model for sky dome polarisation.

While rich in features, the Prague Sky Model remains easy to use and can be integrated in any path tracer. It has been implemented in the Corona renderer [Chaos Czech a.s., 2023] and successfully used there to this day. A few examples of its output for different settings and scenes are shown in Figure 3.1. They do not demonstrate just the clear sky colours but also the haze covering more distant objects requiring the knowledge of finite distance in-scattered radiance and transmittance. Thanks to all of this difficult atmospheric light transport being pre-computed once and then supplied by the Prague Sky Model, these

examples were rendered in a fraction of time needed for full atmospheric path tracing (in minutes instead of days).

The author was not the primary investigator of the Prague Sky Model, its initial version was published in a doctoral thesis by Hošek [2019]. However, the author then introduced several improvements to the model and collaborated on publishing its final version [Wilkie et al., 2021] while sharing the first authorship with Alexander Wilkie. In this chapter, the author will review this final version while clearly stating what his contribution is. Some parts of this chapter content (Appendix 3.12.3 and several figures) were taken with minimum modifications from the final version publication, but also appeared in the Lukáš Hošek's doctoral thesis. These parts are clearly marked, they do not contain any contribution of the author but are necessary for completeness of this chapter.

The description of the Prague Sky Model will also serve as a background for introducing the author's very own extension of the model spectral range into the short-wavelength infrared (SWIR) region [Vévoda et al., 2022].

Figure 3.1: **Top row:** A mountain landscape rendered from 4.8 km altitude using the Prague Sky Model with finite-distance in-scattered radiance and transmittance. **Left:** 2° solar elevation, 30 km visibility. **Right:** 15° solar elevation, 60 km visibility. **Bottom row:** A cityscape rendered in a late afternoon setting, and post sunset. All rendered using the Prague Sky Model in the Corona renderer.

3.1.1 SWIR extension

The Prague Sky Model has, same as all extant pre-computed models, one important limitation – it was designed with the visible spectral range, and a human observer, in mind. While the visible range is sufficient for many practical use cases of the model, there are applications that require considerably wider range of wavelengths. In particular, for photovoltaic plant yield simulation and thermal analysis of buildings, the model lacks a significant part of the solar irradiance spectrum: specifically, the short-wavelength infrared (SWIR) region up to around 2500 nm is missing. With the importance of renewable energy sources rapidly increasing, and the thermal performance of buildings also becoming more and more

important, accurate design tools for these applications are urgently needed. Even though these application areas developed their own ad hoc prediction toolchains over the years (some of which we discuss in Section 3.2), the trend is now to move towards the kind of MC rendering technology that was originally developed for "normal" computer graphics. The reason being that only MC rendering can handle predictions for arbitrarily complex input geometries, non-trivial surface materials, transparency and translucency (for e.g. complex photovoltaic module coverings), and similar advanced appearance features. Therefore, we derived a suitably extended form of the Prague Sky Model as an initial reference model for these communities.

The extended model is only hemispherical, i.e. limited to the ground level observer altitude, as this is the main use case for photovoltaic simulation and building analysis. It retains all the other components of the Prague Sky Model, such as the transmittance term and polarisation patterns. The main difference is a considerably extended spectral range (280 nm to 2480 nm, see Figure 3.2 for illustration), and the corresponding adaptations that were necessary in the brute force pre-computation step and the fitting. Outside the visible range, several additional factors needed to be included in the simulation, which we discuss throughout this chapter. We also provide an implementation that servers not only for easy evaluation of the extended model in a renderer, but also for interactive visual exploration of the dataset. The implementation is flexible and can be used for evaluation of the original Prague Sky Model as well.

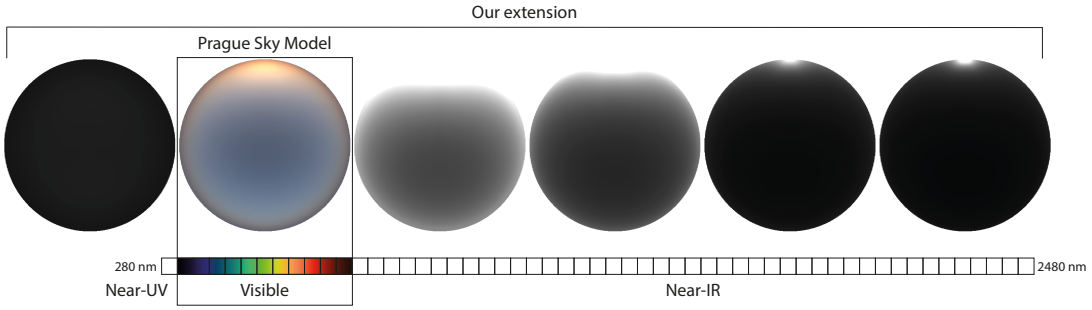


Figure 3.2: Our extension of the Prague Sky Model adds another 44 wavelength channels in the near-UV and near-IR range on top of its original 11 channels in the visible range. Altogether, our extension covers wavelengths from 280 nm to 2480 nm using 55 regularly spaced bins. The displayed monochrome images correspond to 280, 1200, 1600, 2000 and 2480 nm bins; the colour image is a composition of 11 bins from 320 – 760 nm range. They all show up-facing fish-eye views of the sky for solar elevation 3° and visibility 20.0 km

3.2 Previous work

As already outlined in the introduction, work on sky dome radiance and atmospheric transmittance falls into four broad categories within computer graphics:

1. Capture and measurement of real skies

2. Interactive approximations to atmospheric scattering

3. Fitted analytical models

4. Brute force simulations of light transport in the atmosphere

For fitted analytical models and brute force simulations there exists a considerable body of work within the atmospheric research community, for which we provide a summary in Table 3.1, and we refer the reader to the comprehensive overview and evaluation of fitted analytical models and brute force solvers by Bruneton [2016]. In the remainder of this section we provide more details about selected work from each category.

	sun below horizon	**arbitrary observer altitude**	**spectral**	**polarisation**	**approach** b (brute force) f (fitted model)
Nishita93	+	+	-	-	b
Nishita96	+	+	-	-	b
Preetham	-	-	-	-	f
Haber	+	-	-	-	b
Bruneton	+	+	+	-	b
Elek	+	+	+	-	b
Hošek	-	-	+	-	f
libradtran	+	+	+	+	b
Prague Sky Model	+	+	+	+	f

Table 3.1: A comparison of several clear sky models and their features.

3.2.1 Capture and measurement

Captures of sky dome radiance are commonly used in production environments. There are collections of stock HDR sky images, the film industry have been relighting their scenes using on-set captured sky dome images for years. However, as mentioned in the introduction, these captures lack flexibility and matching associated atmospheric data (in-scattering, transmittance, polarisation). Neither they can be used for creation or at least verification of fitted sky dome models as they are rarely taken in a calibrated manner and cannot therefore serve for rendering in absolute units. For the wider spectral range of our SWIR extension, there are hardly any captured datasets at all.

The work of Kider et al. [2014] containing systematic measurements of sky dome radiance is a very helpful exception. It can be used to directly illuminate scenes as well as to verify sky models and it provides data for both the visible and SWIR spectral range. The only shortcoming of this work is the lack of the exact atmospheric parameters at the time of capture. Since there are many degrees of freedom in how a clear atmosphere can be structured, verification against the dataset of Kider et al. [2014] requires assuming a particular atmosphere model and experimental search for its parameters yielding the best match. However, as the comparison in Figure 3.52 shows, the U.S. Standard Atmosphere, which is used as basis of the Prague Sky Model, is a reasonable fit for the measurements.

3.2.2 Interactive approximations

Interactive approximations like the work of Hillaire [2020] provide excellent results for interactive settings but are problematic for our purpose due to the potentially unbounded error they introduce. Also, these methods usually operate in a colour space and there is no data on how to make them perform reasonably well in the infrared spectral region.

3.2.3 Fitted analytical models

There are several models with widespread industrial use that attempt to fit parametric functions to patterns observed in brute force sky dome radiance simulations [Perez et al., 1993; Preetham et al., 1999; Hošek and Wilkie, 2012, 2013]. They are useful for fast, high-quality scene modelling and rendering, but they typically provide realistic results only for a limited parameter range. Moreover, the applicability of these models is narrowed down because they all were designed with only a ground-based observer in mind. The work of Hošek and Wilkie [2012, 2013] also lacks a dedicated atmospheric transmittance model that matches the sky dome radiance. The Prague Sky Model adds all these features, albeit at the cost of requiring pre-computation for the required range of parameters, and resulting in a model which has larger memory requirements than previous techniques. Figure 3.3 illustrates the conceptual difference of these models to a full solution like the Prague Sky Model provides, while Figure 3.4 shows the consequences of omitting individual components.

Regarding the SWIR extension, to our knowledge, no fitted model that covers the spectral range needed for full photovoltaic assessments exists.

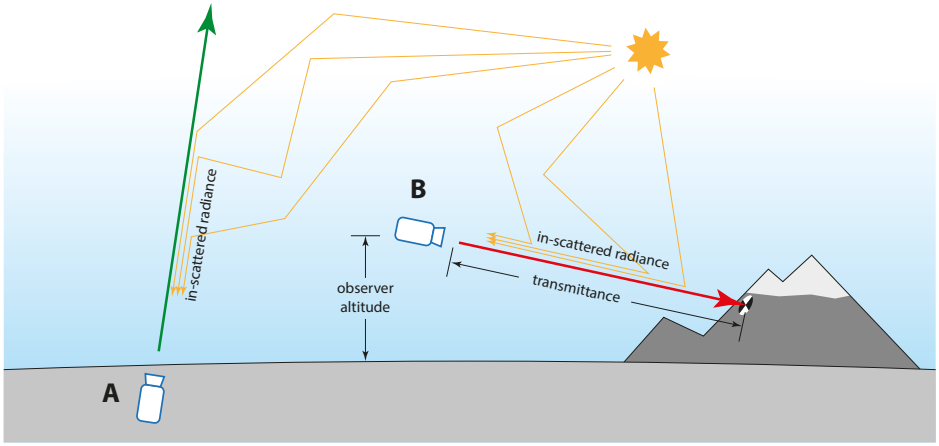


Figure 3.3: Capabilities of fitted sky radiance models. Existing models (such as the one by Hošek and Wilkie [2012]) correspond to case **A**: they are quite simple, and only provide information for paths that directly go to space without hitting any objects, and with the observer at ground level. They also do not provide a model for transmittance over finite viewing distances. But for non-trivial renderings, one *additionally* needs to cover case **B**: with an observer viewpoint that is not necessarily on the ground, with in-scattered radiance information for finite viewing distances when objects are hit, and with matching transmittance information. The Prague Sky Model provides all this.

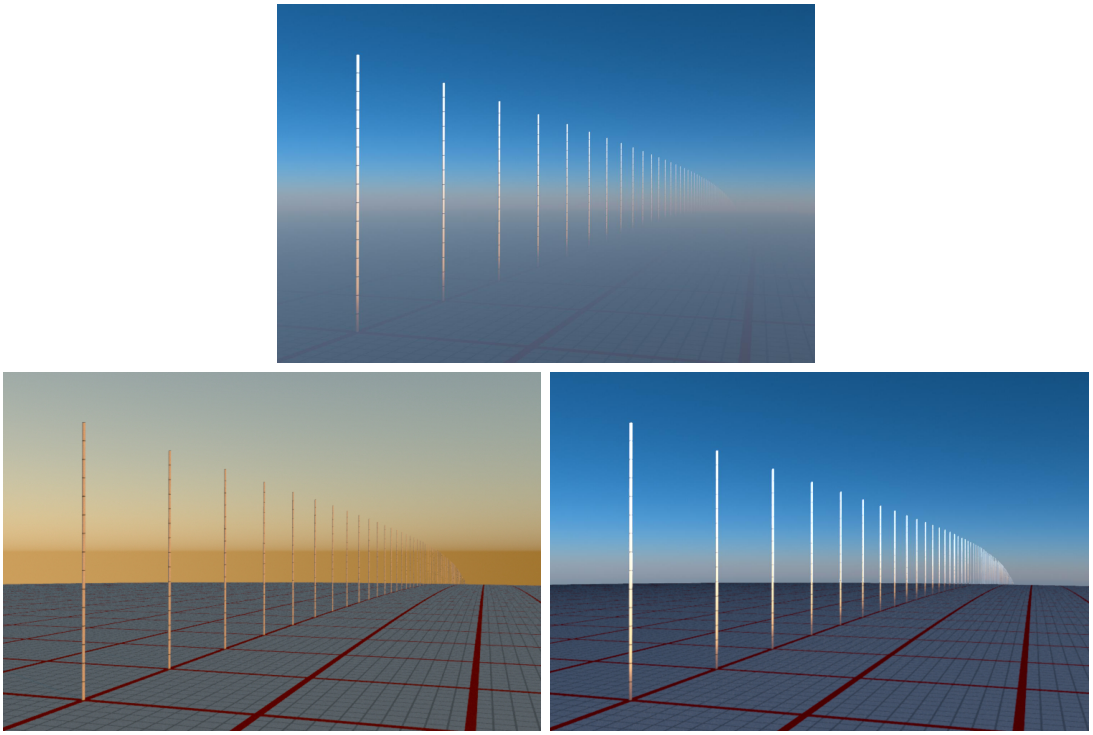


Figure 3.4: Consequences of omitting some of the pre-computation components identified in Figure 3.3. **Top:** a full brute force rendering of a synthetic test scene (the Columns scene described in Section 3.9.1), observer altitude 8 km above ground, close to sunset. **Bottom left:** the model by Hošek and Wilkie [2012]. It neither has a concept of observer viewpoints above ground level (so downward looking rays need to extrapolate, and solar radiance stays the same at all altitudes), nor provides an expression for in-scattered radiance or transmittance for finite distances. **Bottom right:** the Prague Sky Model with in-scattered radiance and transmittance for finite viewing distances switched off: as one can see, this component is absolutely crucial for outdoor scene realism.

Transmittance

Transmittance models have been included in some works [Preetham et al., 1999], however these typically rely on simplifications such as exponential distributions of aerosols. Other approaches for transmittance operate, and require features, in image space so cannot be easily applied to multiple bounces of lighting [Hansard, 2019]. The Prague Sky Model provides a specialised transmittance parametrisation and fit, is accurate to the underlying atmospheric configuration, and can be used when computing indirect lighting as well.

Polarisation

An approximate analytical fit of sky dome polarisation was proposed by Wilkie et al. [2004]. However, for lack of reference data their model was based entirely on indirect reasoning (as per their own admission in the paper), which led to rather sub-optimal results as showed by Wang et al. [2016]. Therefore, Wang et al. designed a new analytical model of sky dome polarisation. It is accurate and efficient to evaluate, but no matching radiance or transmittance data were

provided. As a consequence, it is of limited applicability to rendering of complete skies with all aspects of sky dome radiance, i.e., in-scattered radiance, transmittance, and polarisation. By contrast, these are all covered by the Prague Sky Model in one integrated solution.

3.2.4 Brute force simulations

Brute force simulations tend to yield great results in terms of visual quality but are usually not nearly fast enough for production use. This category includes, e.g., the work of Nishita et al. [1993, 1996], Haber et al. [2005], Bruneton and Neyret [2008] or Guimera et al. [2018]. Most of them actually contain a pre-computation step that reduces the complexity of the models' evaluation at the time of rendering, albeit for the set of parameters (observer altitude, sun elevation) fixed at the pre-computation step. Real-time methods, such as O'Neil [2005] compute approximations to atmospheric lighting; these typically trade effects such as multiple scattering for fast computation.

Significantly more powerful general simulation packages such as `libradtran` [Emde et al., 2016] has been developed by the atmospheric research community. These can serve as a valuable source of reference solutions for graphics research, such as those shown in the work of Wang et al. [2016]. However, for the reasons we discuss in Section 3.6.1, and notwithstanding all its excellent capabilities, `libradtran` is not well suited for general rendering tasks, and would not have been a good solution for the very specific problem of obtaining reference images for the Prague Sky Model.

3.2.5 Wide spectral range

Besides `libradtran`, a few more infrared-capable models was developed. For example, the SMARTS spectral irradiance model [Gueymard, 2019] provides high-resolution spectral irradiance output for potentially sloped surfaces at ground level, and for a variety of clear-sky scenarios. Over 25+ years of usage, it has been extensively validated, and is widely used in many simulation fields.

The remote sensing community has developed a number of wide-band hyper-spectral reference solvers, such as DART [Grau et al., 2009]. Several other software packages of roughly the same type exist (e.g. 6S [Vermote et al., 1997]): common to all of them is that hyper-spectral predictions for scene appearance *when viewed from orbit* are provided, to aid with the interpretation of real data acquired during satellite passes. In the case of DART, satisfactory performance is achieved via careful trade-offs between discretisation and other approximations. For the purposes it is intended for, the accuracy it reliably delivers is perfectly sufficient and has become a standard tool for such applications. However, for simulation of photovoltaic plant yield *viewed from the ground*, all these solvers suffer from the fact that they are custom-made for the specific purpose of "seeing things from orbit" and altering them to provide such functionality would mean a partial or complete rewrite.

Common to all these models is that they are very sophisticated, highly accurate and extensively verified but that they would be hard to integrate into modern MC rendering software. This was not their original purpose, so this fact

in no way is to be held against them. But, as already stated in the introduction, light transport simulation technology is generally moving in the direction of MC rendering, due to its significantly higher predictive capabilities for complex scenes and materials.

3.3 Physics background

In this section, we briefly review physics aspects of atmospheric rendering. First, we describe light transport in the atmosphere using the radiative transfer equation. Then, we name the typical atmosphere components and explain what light transport processes they contribute to. Finally, we give a short introduction into light polarisation.

3.3.1 Radiative transfer equation

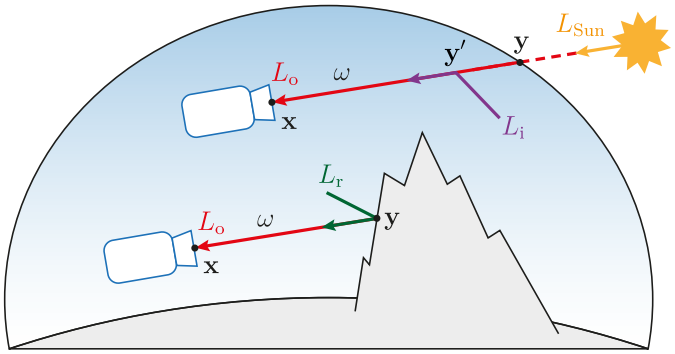
A complete description of light transport in participating media, such as in atmospheres, is given by the radiative transfer equation [Subrahmanyan, 1960], for which an exhaustive introduction is given by Pharr et al. [2016]. Here we describe a version of the radiative transfer equation specific for atmospheric rendering.

There are two main processes of light transport in the atmosphere: *absorption*, when light collides with an atmosphere particle and is converted to another form of energy (e.g., heat), and *scattering*, when light collides with an atmosphere particle and is scattered into a different direction. While absorption only reduces radiance along a ray, scattering causes both radiance decrease, when light travelling along the ray is scattered away (out-scattering), as well as increase, when light travelling away is scattered into the path of the current ray (in-scattering).

Intuitively, the radiative transfer equation expresses radiance arriving along a ray as: radiance entering at the beginning of the ray (attenuated by absorption and out-scattering along the entire ray) plus all radiance that gets in-scattered into the ray anywhere along its path (attenuated by absorption and out-scattering along the remaining part of the ray).

Formally, radiance $L_\mathrm{o}(\lambda, \omega \to \mathbf{x})$ of a wavelength λ arriving along a ray from a direction ω to a point $\mathbf{x}$ is computed as

$$L_\mathrm{o}(\lambda, \omega \to \mathbf{x}) = Tr(\lambda, \mathbf{y} \to \mathbf{x}) L_\mathrm{e}(\lambda, \mathbf{y} \to \omega) + \int_\mathbf{x}^\mathbf{y} Tr(\lambda, \mathbf{y}' \to \mathbf{x}) L_\mathrm{i}(\lambda, \mathbf{y}' \to \omega) \, \mathrm{d}\mathbf{y}', \quad (3.1)$$



where:

- $\mathbf{y}$ is a point at the beginning of the ray, either on a solid surface nearest to $\mathbf{x}$, or in infinity, if the ray does not hit any surface. Note that the latter case is equivalent to point $\mathbf{y}$ located at the atmosphere boundary, since we do not consider any light interactions in outer space.

- $\int_{\mathbf{x}}^{\mathbf{y}}$ is an integral over all points on a line segment connecting $\mathbf{x}$ and $\mathbf{y}$.

- $Tr(\lambda, \mathbf{y} \to \mathbf{x})$ is the *transmittance*. It is a value between 0 and 1 and expresses the fraction of radiance of wavelength λ that is transmitted from $\mathbf{y}$ to $\mathbf{x}$. The transmittance is 1 in a vacuum but lesser than 1 in the atmosphere because of radiance lost due to absorption and out-scattering. It can be computed as

$$Tr(\lambda, \mathbf{y} \to \mathbf{x}) = \mathrm{e}^{-\int_{\mathbf{x}}^{\mathbf{y}} \sigma_\mathrm{t}(\lambda, \mathbf{y}')\,\mathrm{d}\mathbf{y}'}, \tag{3.2}$$

  where $\sigma_\mathrm{t}(\lambda, \mathbf{y}')$ is the *extinction coefficient* [m^{-1}]. It is the probability density that light of the wavelength λ is absorbed or scattered at the point $\mathbf{y}'$ per unit distance travelled in the atmosphere. For a clear sky, the extinction coefficient depends only on λ and the altitude of $\mathbf{y}'$ and can be computed as a product of the wavelength-dependent *extinction cross section* [m^2] and the altitude-dependent *particle concentration* [m^{-3}]. Similarly, absorption coefficient σ_a and scattering coefficient σ_s can be defined as the probability densities of just absorption or scattering and can be computed using the respective absorption and scattering cross sections. It holds that $\sigma_\mathrm{t} = \sigma_\mathrm{a} + \sigma_\mathrm{s}$.

- $L_\mathrm{e}(\lambda, \mathbf{y} \to \omega)$ is radiance of the wavelength λ emitted or reflected from the point $\mathbf{y}$ in the direction ω. We define it as

$$L_\mathrm{e}(\lambda, \mathbf{y} \to \omega) = \begin{cases} 0 \ldots \text{for rays coming from outer space,} \\ L_\mathrm{Sun}(\lambda) \ldots \text{for rays coming from the Sun,} \\ L_\mathrm{r}(\lambda, \mathbf{y} \to \omega) \ldots \text{for rays coming from a solid surface,} \end{cases}$$

  where L_Sun is the *extraterrestrial solar radiance* and L_r is radiance reflected from the surface. For Lambertian surfaces L_r satisfies

$$L_\mathrm{r}(\lambda, \mathbf{y} \to \omega) = \frac{\alpha(\lambda, \mathbf{y})}{\pi} \int_{H^2} L_\mathrm{o}(\lambda, \omega' \to \mathbf{y})\, \mathbf{n}_\mathbf{y} \cdot \omega' \,\mathrm{d}\omega', \tag{3.3}$$

  where $\int_{H^2}$ denotes the hemispherical integral and $\mathbf{n}_\mathbf{y}$ is the surface normal at $\mathbf{y}$. $\alpha(\lambda, \mathbf{y})$ is the *surface albedo* for the wavelength λ at $\mathbf{y}$. It is a value between 0 and 1 and expresses the ratio between the reflected and incoming radiance.

- $L_\mathrm{i}(\lambda, \mathbf{y}' \to \omega)$ is radiance of the wavelength λ in-scattered at the point $\mathbf{y}'$ into the direction ω. Intuitively, it is all the radiance coming from anywhere in the scene scattered into ω, attenuated by the probability of scattering and the probability of changing its direction into ω. It is defined as

$$L_\mathrm{i}(\lambda, \mathbf{y}' \to \omega) = \sigma_\mathrm{s}(\lambda, \mathbf{y}') \int_{S^2} \rho(\lambda, \omega' \to \mathbf{y}' \to \omega) L_\mathrm{o}(\lambda, \omega' \to \mathbf{y}')\,\mathrm{d}\omega', \tag{3.4}$$

  where $\int_{S^2}$ denotes the spherical integral and $\rho(\lambda, \omega' \to \mathbf{y}' \to \omega)$ is the *phase function*, which defines the probability density of scattering radiance of the wavelength λ arriving from direction ω' at the point $\mathbf{y}'$ into the direction ω.

3.3.2 Typical atmosphere composition

Now we mention atmosphere particles that are typically responsible for the absorption and scattering described above. In physical simulations, the atmosphere is considered to consist of gas molecules, aerosols and clouds, where clouds can be further divided into water clouds and ice clouds [Emde et al., 2016]. Since the Prague Sky Model only deals with clear skies, we do not discuss clouds further. Also, two main scattering mechanisms prevail in the atmosphere: Rayleigh and Mie scattering. Other types of mechanisms, e.g., scattering by non-spherical particles such as ice crystals, are not present in all atmospheric configurations, and are not taken into account in the Prague Sky Model.

Gas molecules

The two main constituents of air, N_2 and O_2, form roughly 78.1% and 20.9% of air, respectively, up to the altitude of about 90 km, where the overall air concentration is already negligible. These molecules are responsible for Rayleigh scattering and are the main cause of the blue colour of the sky.

Rayleigh scattering describes the interaction of light with particles smaller then the light wavelength. Closed-form expressions for calculating its cross section and phase function are available [Bodhaine et al., 1999].

Besides scattering, O_2 is also responsible for a small absorption band around 760 nm. Even further into the infrared region, there are absorption bands caused by CO_2 and water vapour H_2O. See Figure 3.5 for an illustration.

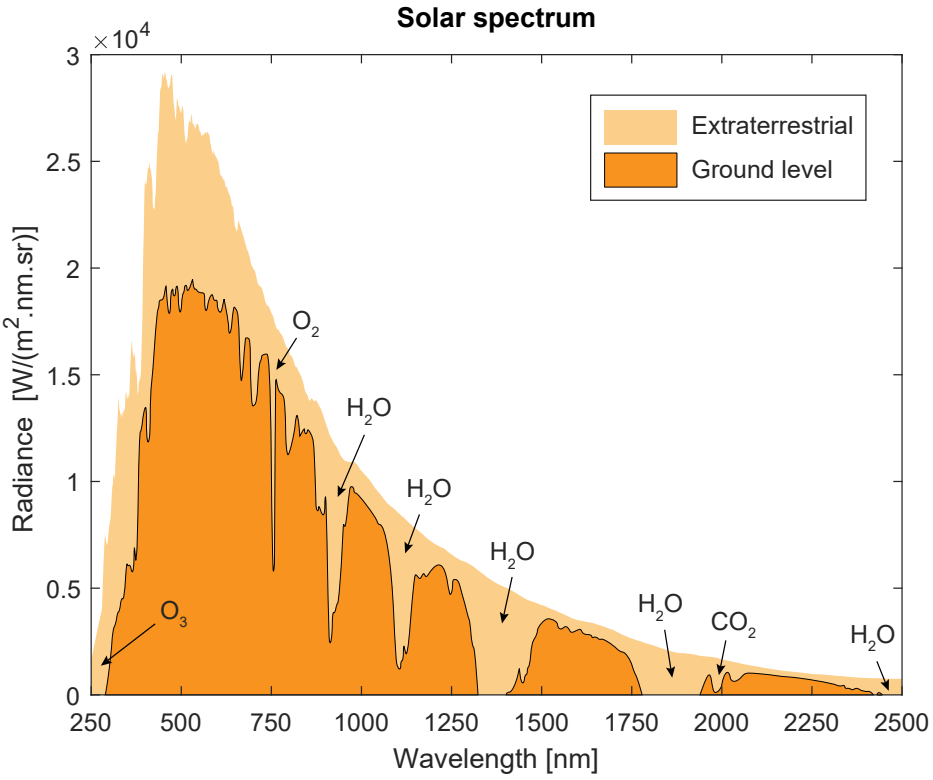


Figure 3.5: A comparison of the extraterrestrial solar spectrum and the solar spectrum at ground level. Note the absorption bands caused by molecules of CO_2, H_2O, O_2 and O_3.

On the other side of the spectrum, around 250 nm, strong absorption is caused by ozone O_3, which has long been assumed to be important for sky appearance at dusk [Hulburt, 1953]. Although the direct correspondence between twilight sky colour and ozone concentrations has been recently called into question [Lee et al., 2011], if skies with low solar elevations are to be rendered correctly, the inclusion of O_3 is a necessity, as shown in Figure 3.6. So far, it has mostly been omitted in computer graphics sky models [Nishita et al., 1993; Preetham et al., 1999], and even though two models already include it [Haber et al., 2005; Kutz, 2013], no widely used fitted model features the effect yet.

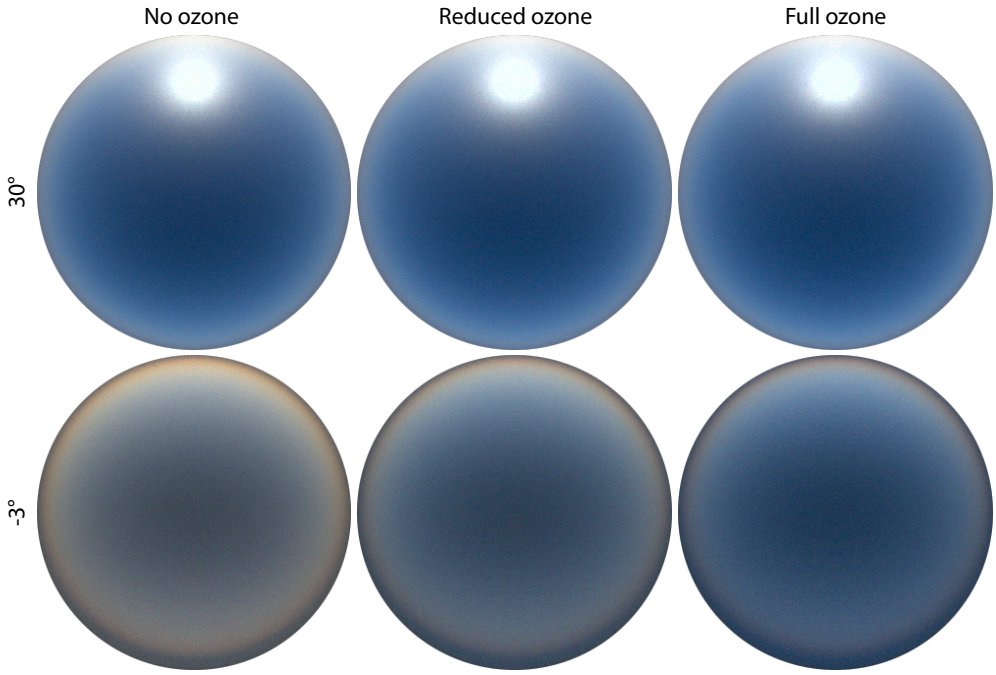


Figure 3.6: A demonstration of the impact of ozone on sky dome appearance. **From left to right:** Up-facing fish-eye views of the sky with no ozone, with a reduced ozone profile typical for ozone hole conditions, and a healthy mid-latitude ozone layer. **Top row:** Solar elevation 30°. **Bottom**: Solar elevation −3°. While there is no visible change for the high solar elevation, the increasing ozone concentration causes a dramatic change in hue and radiance for the low elevation. For this figure, the atmosphere was simulated with gas molecules only (i.e., without aerosols). *Note: this figure taken from the paper of Wilkie et al. [2021] also appeared in the doctoral thesis of Hošek [2019] (on page 54).*

Aerosols

The second major constituent of the atmosphere are aerosols. An aerosol property database for these particles called OPAC (Optical Properties of Aerosols and Clouds) is available [Hess et al., 1998]: it contains several basic aerosol types that are typically present in an atmosphere, e.g., water-soluble (WASO), water-insoluble (INSO) and black carbon (SOOT). These particles are responsible for absorption and since they are larger, they are also responsible for the second type of scattering – Mie.

Mie scattering describes the interaction of light with scatterers that are larger than the light wavelength. As the scattering favours forward directions, it produces characteristic coronas around light sources in foggy environments. In the atmosphere, the aerosol particles have various size distributions [Hess et al., 1998], which influences the cross sections and phase functions. Therefore, unlike Rayleigh scattering, simple closed-form formulas are not available for Mie scattering, and it has to be approximated [van de Hulst, 1957] or pre-computed from the size distributions and tabulated separately for each wavelength [Emde et al., 2016]. Mie phase function is often approximated by Henyey-Greenstein [Henyey and Greenstein, 1941] or Cornette-Shanks [Cornette and Shanks, 1992] phase functions.

3.3.3 Polarisation

Light emitted from the Sun is unpolarised. Only after scattering on atmosphere particles it becomes polarised. Both Rayleigh and Mie scattering are polarising light-matter interactions: but for sky dome scenes, the macroscopically resulting polarisation is often rather weak in the case of Mie scattering.

Mueller calculus [Mueller, 1948] describes polarising and attenuating properties of one such interaction using a 4×4 Mueller matrix and the full state of light between the interactions using a Stokes vector. A Stokes vector is a four-component vector (I, Q, U, V), where I is the light radiance, Q is the amount of linear horizontal polarisation, U is the amount of linear diagonal polarisation, and V is the amount of circular polarisation. For example, $(I, 0, 0, 0)$ denotes unpolarised light, $(I, 1, 0, 0)$ horizontally polarised light, and $(I, -1, 0, 0)$ vertically polarised light.

A Stokes vector is always associated with a reference frame given by the vector of light propagation and two other perpendicular vectors: one for the horizontal direction and one for the vertical direction. A Mueller matrix is associated with two such reference frames: an input one corresponding to the incoming light direction and an output one corresponding to the outgoing light direction. When light undergoes an interaction (e.g., scattering on an atmosphere particle), the Stokes vector representing the light is multiplied by the Mueller matrix representing the interaction, but only after the vector is rotated so its reference frame matches the input reference frame of the matrix. Effects of multiple interactions can be concatenated together by multiplying their Mueller matrices. However, this involves costly rotation of the matrices so the output and input frames of matrices of each two consecutive interactions match.

In general, the sky is polarised tangential to a circle centered around the sun and maximum polarisation is found at 90° from it. Therefore, the sky is polarised mostly horizontally when the sun is close to the zenith, and mostly vertically when the sun is close to the horizon, as illustrated in Figure 3.7. On very clear days, the degree of polarisation can reach up to 70%, but usually multiple scattering tends to reduce polarisation.

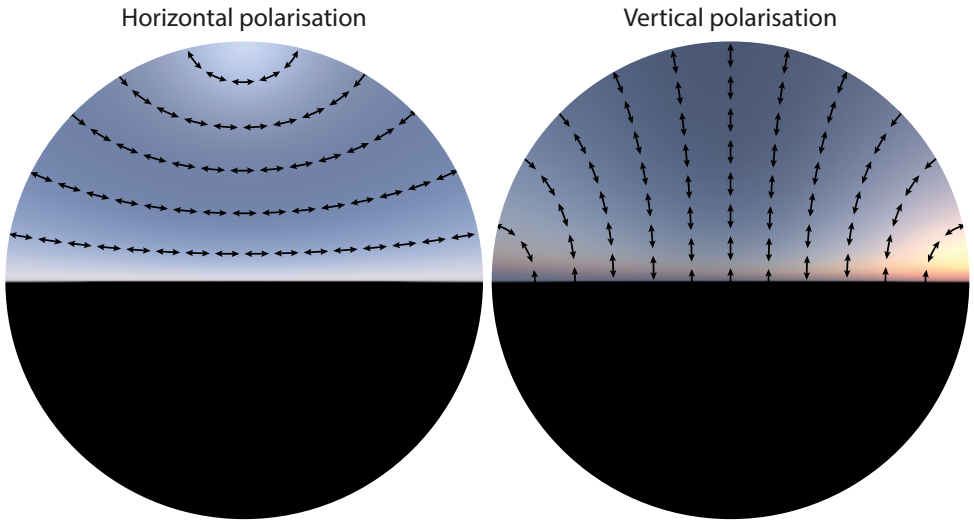


Figure 3.7: Two side-facing fish-eye views of the sky demonstrating different polarisation. **Left:** Mostly horizontally polarised sky when the sun is close to the zenith. **Right:** Mostly vertically polarised sky when the sun is close to the horizon.

3.4 Model parameters

Knowing the physics background of light transport in the atmosphere, we can now start describing the Prague Sky Model and its SWIR extension. We begin with specifying their input parameters.

Spectral range

The Prague Sky Model covers wavelengths in the range from 320 nm to 760 nm, i.e., slightly wider visible range. In contrast, our SWIR extension covers wavelengths in the range from 280 nm to 2480 nm (see Figure 3.2 for a visual comparison of the two ranges). It extends the range a bit on the ultraviolet side of the spectrum, as there is some residual solar radiation as well (see Figure 3.5), but mostly it adds wavelengths from the short-wavelength infrared (SWIR) range. The exact terminology of what constitutes SWIR varies by application area: the boundary between SWIR and medium-wavelength infrared is usually placed somewhere in the region between 3000 nm and 4000 nm. As we are focused on solar radiation, which at ground level practically goes to zero beyond 2500 nm, we limit the range of our model to near that value: explicitly covering the entirety of SWIR was not our goal, especially as there are no really official definitions for these region boundaries anyway. As Figure 3.8 shows, a model that explicitly only targets photovoltaic simulations could probably have stopped even earlier (around 2000 nm): but we wanted to cover the entire solar irradiance spectrum, and even though it does not have a lot of energy beyond 2000 nm, this might still matter for thermal irradiance analysis.

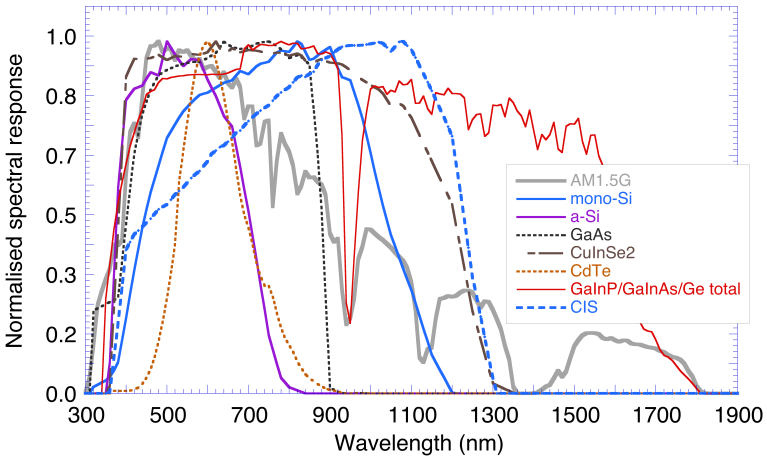


Figure 3.8: Spectral response curves for several solar cell types, with typical ground level solar radiation shown in grey. Note that several types absorb significant amounts of radiation way beyond the visible range: any predictions of photovoltaic plant yield based only on visible wavelengths would be considerably in error. Image courtesy of Chris Guyemard, used by permission.

Ground albedo

The effect of surface properties of any local scene geometry on the sky appearance is negligible. What matters is the overall reflectance of large areas on the Earth surface, i.e., if any significant portion of the incoming light is reflected back into the atmosphere. To account for this, the Prague Sky Model assumes the Earth surface to be Lambertian so the ratio between the reflected and incoming light is given by the surface albedo (see (3.3)). The average surface albedo of the ground in the geographic region where the model is used is then one of the model parameters. The SWIR extension uses this ground albedo as well, but thanks to the wider spectral range the ground albedo can specify reflection also for infrared wavelengths.

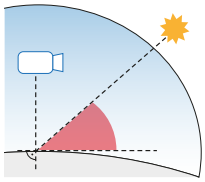
Observer altitude

One of the benefits of the Prague Sky Model is not being limited to ground-based observers like most of the current models. It is parametrized by the observer altitude which extends from ground level up to 15 km – altitude high enough to allow rendering of airliners in flight. The SWIR extension has a different application though. It is meant for photovoltaic plant yield simulation and thermal analysis of buildings which is usually not needed above ground level. Therefore, the spectrally extended model is only hemispherical. However, this is just an optimization to reduce the model size and the time required for the brute force simulation and adding higher altitudes would be straightforward.

Solar elevation

Solar elevation is the elevation of the sun above ground level, i.e., the angle between a plane tangent to the Earth at the point on the ground directly below the observer and the direction from that point to the Sun. It defines the vertical

position of the sun in the sky independently on the current observer altitude (horizontal position can be changed trivially just by rotating the sky around zenith). The Prague Sky Model accepts values from $-4.2°$ to $90°$: $90°$ means the sun is in the zenith, $0°$ means the center of the sun is exactly on the horizon when viewed from the ground, and $-4.2°$ was set so the sun is completely below horizon even when viewed from the maximum observer altitude 15 km. The SWIR extension keeps this parameter unchanged.



Visibility

In order to control the haziness of outdoor scenes, existing sky dome models feature a user-controllable parameter called *turbidity*. This is a relative measure of the fraction of additional scattering due to aerosols as opposed to molecules only [Preetham et al., 1999], i.e., how much more the atmosphere scatters compared to an ideally clean molecular atmosphere:

$$T = \frac{\tau_\mathrm{m} + \tau_\mathrm{h}}{\tau_\mathrm{m}}, \tag{3.5}$$

where τ_m is the optical thickness of molecules only and τ_h is the optical thickness of aerosols only. This value is typically reported at 550 nm and *measured towards the zenith*. As we will see in the next section, such vertical measurements are not a good correlate of haziness when using realistic scatterer profiles: they only correlate well when using exponential scatterer profiles.

As the Prague Sky Model is using realistic scatterer profiles with a large amount of aerosols close to the ground (described in the next section), the model is parametrised via horizontal viewing distance at ground level – *visibility* for short. It is defined by the Koschmieder's formula [Horvath, 1971] as

$$V = \frac{-\ln(0.02)}{\sigma_\mathrm{t}(550, 0)}, \tag{3.6}$$

where $\sigma_\mathrm{t}(550, 0)$ is the extinction coefficient at 550 nm and ground level. It tells the maximum distance at which an object is still recognizable against the sky on the horizon, so it is also a more intuitive parameter for end users then the turbidity. Visibility in both the Prague Sky Model and its SWIR extension ranges from 20 km to 131.8 km (the range is given by the available atmospheric data). This would correspond to turbidities from 3.8 to 1.4.

3.5 Model atmosphere

Aim of this section is to give an overview of what atmosphere-related data are used by the Prague Sky Model and its SWIR extension and how they were obtained. Appendix 3.12.1 then presents all these data explicitly in the form of plots.

We start outside the atmosphere by describing the only light source considered by the models – the Sun. Its angular diameter in the sky ranges from $0.5242°$ to $0.5422°$, the models use the mean value $0.5334°$. The emission is defined by the extraterrestrial solar radiance [Wehrli, 1985] shown in Figure 3.9.

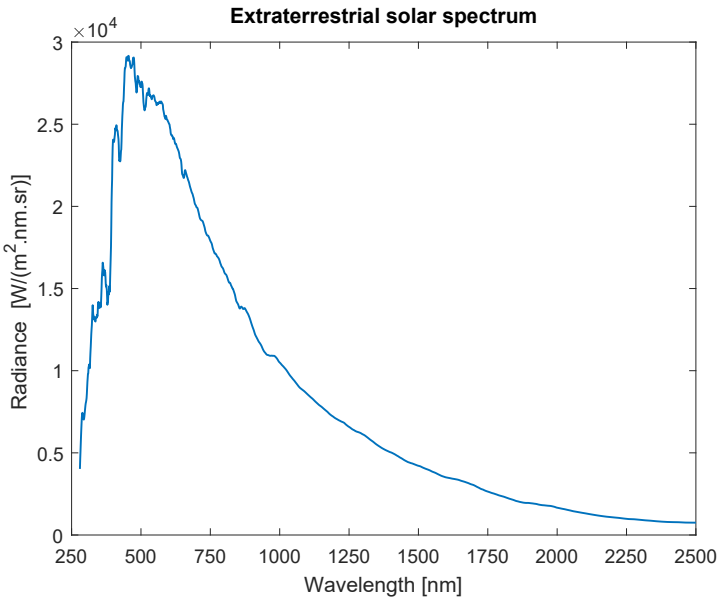


Figure 3.9: Extraterrestrial solar radiance used used by the Prague Sky Model and its SWIR extension.

Now we can continue with the Earth and its atmosphere. The Earth is modelled as a perfect sphere with 6378 km in radius, the atmosphere is assumed to be 120 km thick [Anderson et al., 1986]. To describe the atmosphere composition, we need to list all particles considered by the models together with their properties required by the radiative transfer equation (3.1): absorption cross sections and particle concentrations for absorbing particles, and scattering cross sections, phase functions and particle concentrations for scattering particles.

3.5.1 Gas molecules

Scattering molecules

As discussed in Section 3.3.2, most of air is formed by N_2 and O_2 molecules which cause Rayleigh scattering. The Prague Sky Model and its SWIR extension include both types of molecules, the scattering cross section and phase function are computed using the closed-form expressions from Bodhaine et al. [1999], the particle concentration comes from the U.S. Standard Atmosphere [Anderson et al., 1986] (obtained via `libradtran` [Emde et al., 2016]).

Absorbing molecules

The Prague Sky Model includes absorption from O_3, the SWIR extension takes into account also three additional types of molecules: CO_2, water vapour H_2O, and O_2. These were omitted from the Prague Sky Model since their effect for wavelengths shorter than 760 nm is negligible. However, they are responsible for significant absorption bands in the SWIR part of the spectrum, as shown in Figure 3.5. The absorption cross section of O_3 comes from Gorshelev et al. [2014], the rest from the HITRAN database [Gordon et al., 2022]. All particle concentrations come from the U.S. Standard Atmosphere [Anderson et al., 1986] (obtained via `libradtran` [Emde et al., 2016]).

3.5.2 Aerosols

Both the Prague Sky Model and its SWIR extension contain the three main types of aerosols mentioned in Section 3.3.2: water-soluble (WASO), water-insoluble (INSO) and black carbon (SOOT). These aerosols are responsible for both absorption and Mie scattering, all their necessary properties (cross sections, phase functions, particle concentrations) were computed from the tabulated data provided by OPAC [Hess et al., 1998] and `libradtran` [Emde et al., 2016].

WASO phase function

For simplicity, instead of full Mie scattering both the Prague Sky Model and its SWIR extension use a phase function based on the closed-form Henyey-Greenstein (HG) approximation [Henyey and Greenstein, 1941]. The HG function is parametrized by the asymmetry parameter g, which is also provided by OPAC, but with limited accuracy. Therefore, a separate wavelength-dependent g was numerically fitted to better match the phase function tabulated in OPAC. Figure 3.10 shows an example of the fitting results.

INSO phase function

The models use the HG function for INSO as well. However, for the strongly forward scattering INSO particles, the numerical fit of g resulted in values close to 1, which caused firefly artefacts during brute force rendering, and a narrow and hard to fit high energy region around the solar disc. For both these reasons, the models resorted to using the less accurate asymmetry parameter provided by OPAC, which is slightly lower, and which slightly blurs the circumsolar region. Figure 3.11 shows an example of quality of the fit using the OPAC asymmetry parameter, its impact on validity of the results is discussed in Appendix 3.12.2.

SOOT phase function

The Prague Sky Model approximates the Mie phase function for the SOOT particles using the HG function and fitted g similarly to WASO. While this approximation is sufficient in the original wavelength range, it starts to deviate significantly from the true Mie phase function for longer wavelengths. Therefore, the SWIR extension uses the Cornette-Shanks phase function [Cornette and Shanks, 1992] which is a much better match in this range, as shown in Figure 3.12.

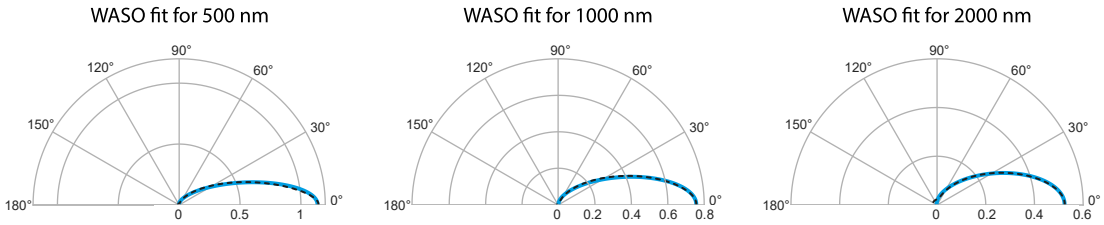


Figure 3.10: A result of approximating the Mie phase function tabulated in OPAC for the WASO aerosol by a fitted Henyey-Greenstein phase function. Black dashed line corresponds to the tabulated Mie function, blue line to the HG function obtained by fitting its asymmetry parameter g so the two functions match.

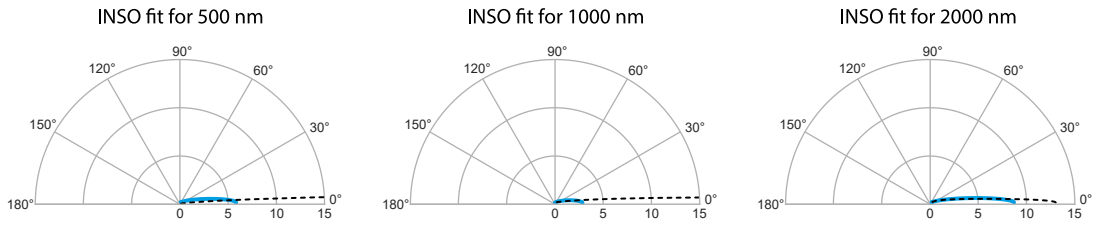


Figure 3.11: A result of approximating the Mie phase function tabulated in OPAC for the INSO aerosol by a fitted Henyey-Greenstein phase function. Black dashed line corresponds to the tabulated Mie function, blue line to the HG function with asymmetry parameter g provided by OPAC.

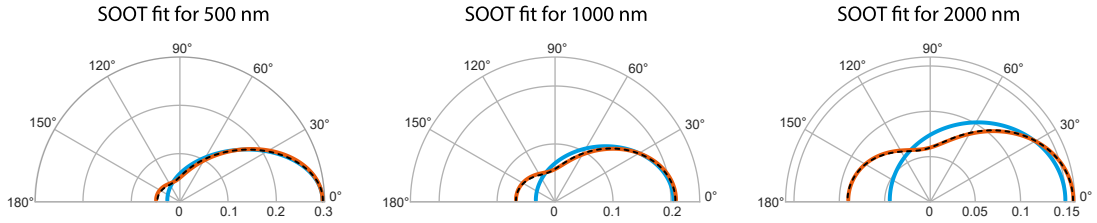


Figure 3.12: A comparison of Henyey-Greenstein (blue) and Cornette-Shanks (red) phase functions with the true Mie phase function (black dashed) provided for the SOOT aerosol by OPAC. While Cornette-Shanks matches the true phase function well, Henyey-Greenstein deviates significantly for longer wavelengths.

Particle concentration

There are several vertical profiles of the aerosol concentrations provided by OPAC for various environments, the Prague Sky Model and its SWIR extension use three: *continental clean*, *average* and *polluted*. These standard profiles are based on real measurements, but simplified insofar as in OPAC these are modelled by two exponentials with a sharp transition at the inversion layer at 2 km. For atmospheric research purposes, the discontinuities in the OPAC data likely do not matter: but as Figure 3.13 shows, they are clearly visible for observer altitudes near the transition.

In order to obtain practically useful scatterer profiles, the models had to slightly modify the OPAC profiles by smoothing the inversion layer transition. They did so by fitting a sigmoid function to the flat regions of the original profiles at 2 km and 12 km.

The three profiles, clean, average and polluted, correspond to three visibilities (as defined in Section 3.4): 27.6 km, 59.4 km and 131.8 km, respectively. To provide more than just three visibilities and stay as close to the real atmosphere as possible at the same time, the models had to interpolate between the three profiles and even extrapolate. Since differences between the three profiles for a single aerosol are defined in OPAC solely by the ground level particle concentration (i.e., the "left" exponential in the top left image in Figure 3.13 is kept fixed and the "right" exponential is scaled so as to start at one the three ground level particle concentrations), interpolating between the three profiles can be done by interpolating between the three ground level particle concentrations with respect to the visibility.

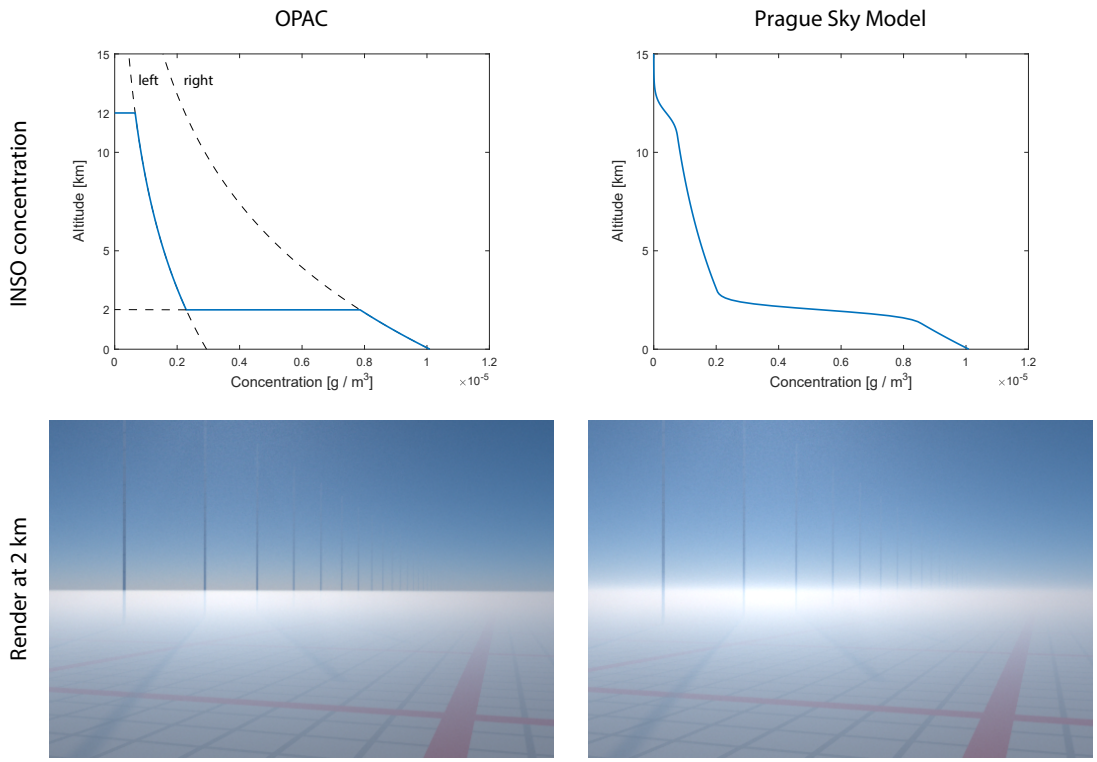


Figure 3.13: A brute force comparison of the original OPAC atmosphere and the smoothed version used by the Prague Sky Model and its SWIR extension. **Top:** The OPAC and smoothed vertical particle concentration profiles for the INSO aerosols. **Bottom:** Reference renderings corresponding to the vertical profiles above, for observer altitude 2 km in a synthetic test scene (the Columns scene described in Section 3.9.1). The sharp horizontal line seen in the left image is not the horizon, but the top of the denser scatterer region.

The interpolation curves shown in Figure 3.14 in the top left image were computed in three steps:

1. A ground level extinction coefficient $\sigma_{\mathrm{t}}^{\mathrm{aero}}(V)$ corresponding to absorption and scattering by all the three aerosol types is computed for a desired visibility V by using (3.6) and subtracting the known visibility-independent extinction coefficient of the gas molecules.

2. Ratios $r(V, A)$ of how much each aerosol type A contributes to $\sigma_{\mathrm{t}}^{\mathrm{aero}}(V)$ are computed using a spline interpolation from the ratios $r(27.6, A)$, $r(59.4, A)$, $r(131.8, A)$ know from the three available profiles (the interpolation is easy as shown in Figure 3.14 in the top right image).

3. Ground level concentrations $c(V, A)$ are then obtained by dividing the extinction coefficient $\sigma_{\mathrm{t}}^{\mathrm{aero}}(V)r(V, A)$ by the respective sums of the known absorption and scattering cross sections.

This way realistic vertical profiles of the aerosol concentrations can be computed for any visibility while keeping the standard profiles where available. An example for the INSO aerosol is shown in Figure 3.14 in the bottom image, profiles for the other aerosols can be found in Appendix 3.12.1.

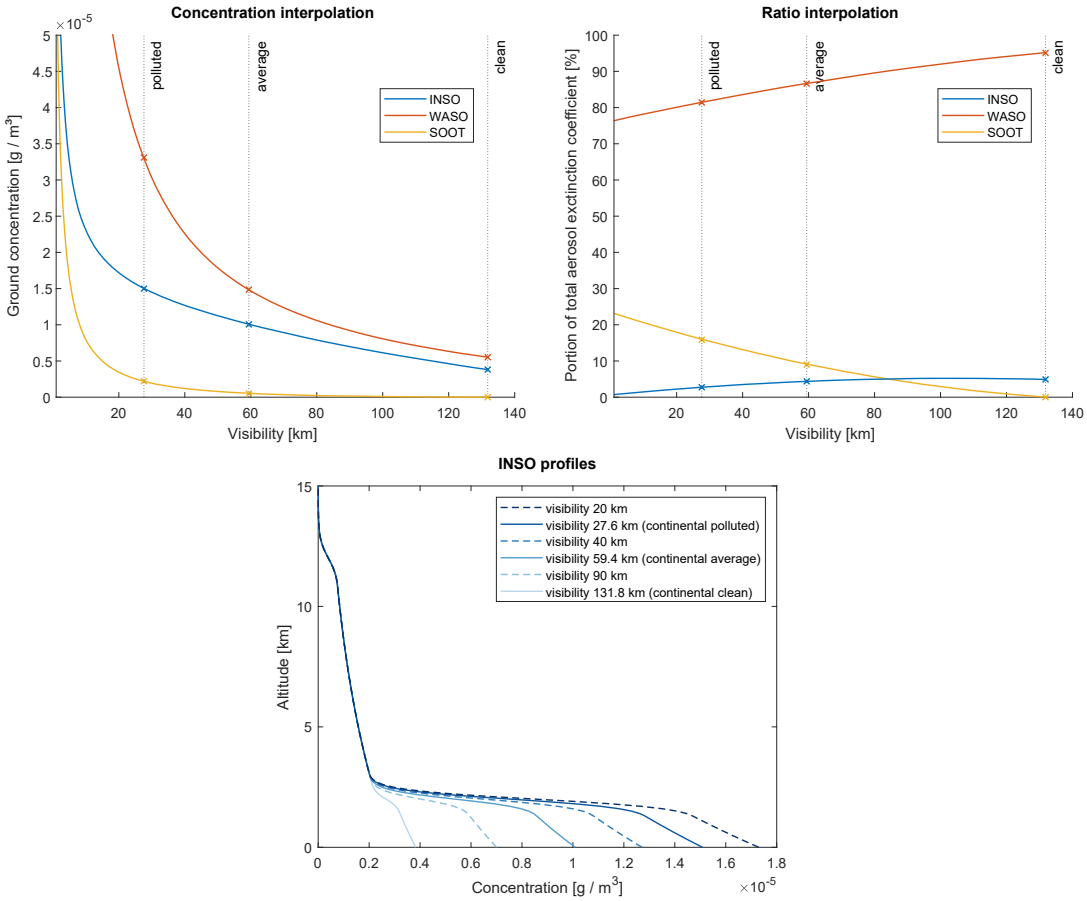


Figure 3.14: Interpolation of vertical concentration profiles of the aerosols. **Top left:** Curves interpolating between three ground level aerosol concentrations corresponding to the three known concentration profiles – clean, average and polluted. **Top right:** Curves interpolating between *ratios* of three ground level extinction coefficients corresponding to the three known concentration profiles. The curves are easily obtained by spline interpolation and defines the curves in the left image. **Bottom:** An example of resulting interpolated (and extrapolated) concentration profiles for the INSO aerosols.

3.5.3 Author's contribution

Apart from all the changes necessary for the SWIR extension (computing all data for the SWIR region, including the additional absorbing molecules, changing the SOOT phase function), the author was also responsible for the smoothing and interpolation of the OPAC aerosol concentration profiles in the Prague Sky Model. The initial version published by Hošek [2019] used an empirically derived ad hoc function to define synthetic concentration profiles. Thanks to the author's work, more realistic profiles based on real measurements could be used.

3.6 Brute force simulation

The Prague Sky Model is a fitted model and as such it follows the general approach of first running brute force simulations, and then fitting a model to the obtained data. The goal of brute force atmospheric light transport is to generate

authoritative images of complex, realistic simulated atmospheres. In this section, we give a technical overview of the path tracer used to run the brute force simulations, as well as of the resulting reference dataset.

3.6.1 Atmospheric path tracer

Generating reference datasets to fit the Prague Sky Model and its SWIR extension requires an unbiased brute force renderer which is capable of computing large amounts of spectral images for chosen atmospheric configurations. These input images for the fitting need to contain polarisation information, and the used rendering algorithm should be flexible enough that it can exclude directly viewed ground hits – only in-scattered radiance that originates from atmospheric events needs to be fitted. In addition to these requirements, the brute force renderer has to be able to load scene geometry, to run direct brute force computations for non-trivial scenes for validation.

With regard to accuracy of radiative transfer computations in the atmosphere, the `libradtran` software package [Emde et al., 2016] is the yardstick to measure against. However, while `libradtran` is an excellent tool for reference computations, it would not have been well suited to compute the large sets of images needed for fitting the two models. Part of the efficiency of `libradtran` comes from the fact that it always runs only for a single wavelength, and a single query direction: this makes limited bi-directional tracing (and therefore faster convergence) considerably easier. However, the models need entire reference images that contain fairly broad spectral regions: and repeatedly running `libradtran` for multiple query directions and wavelengths is not very efficient. Also, it is not clear if it is possible in `libradtran` to only compute in-scattered energy from atmospheric events, and omit specific ground hits: so their Monte Carlo estimator would likely have to be significantly modified.

`atmo_sim`

Instead, a dedicated brute force simulator called `atmo_sim` was implemented based on the ART framework [Wilkie, 2018]. ART provides infrastructure for storing, analysing and manipulating spectral images that contain polarisation information, and it includes the command line tool `polvis` which was used for the polarisation visualisations in Figure 3.19 and 3.35. `atmo_sim` is an uni-directional path tracer which uses next event estimation and multiple importance sampling [Veach, 1997], and is optimised to deal with scattering events in an atmosphere around an idealised Lambertian planet. Via Null Scattering [Miller et al., 2019] in combination with Hero wavelength sampling [Wilkie et al., 2014], it achieves a rendering performance which is sufficient for the generation of reference datasets, even for below sunset scenarios.

Since `atmo_sim` is a polarisation path tracer, a rare structural feature discussed by Wilkie and Weidlich [2012] was implemented in it for performance reasons: when working with polarised light, attenuations from light-matter interactions that occur along a path should not be concatenated together. As discussed in Section 3.3.3, attenuations in a polarisation renderer are represented as 4×4 Mueller matrices. When two such attenuations are concatenated, one of them has to be rotated to the reference frame of the other: this adds significant

overhead compared to just rotating the simpler Stokes vector of a light sample. Which means that a polarisation-capable path tracer is actually faster if it avoids attenuation concatenation entirely: instead, it retains them separately for the entire path, and individually multiplies each of them with a light sample on its way to the image plane.

A simplified version of `atmo_sim` was used for rendering reference data for transmittance. Instead of full path tracing only ratio tracking was used to compute transmittance at a set of non-uniformly distributed points in the atmosphere illustrated in Figure 3.15. Since transmittance features are related to the 1D distribution of atmospheric constituents which is always aligned with the normal of the planet, for transmittance fitting it is enough to parametrise the atmosphere by altitude from ground level *alt* and distance along the planet surface *d*. This parametrisation is then non-uniformly sampled in the region of the atmosphere which covers the maximum visible distances from the camera to produce the set of points shown in the figure.

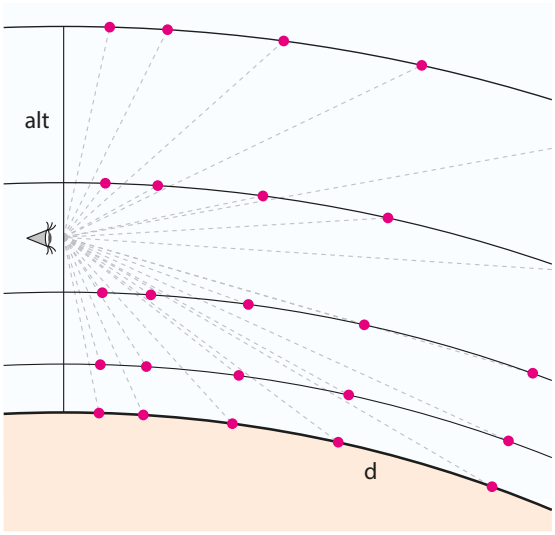


Figure 3.15: An illustration of the 2D atmospheric parametrisation showing the non-uniformly distributed set of points (pink dots) where transmittance is calculated for the model. *Note: this figure taken from the paper of Wilkie et al. [2021] also appeared in the doctoral thesis of Hošek [2019] (on page 98).*

The SWIR extension

For the SWIR extension we had to modify both `atmo_sim` and ART to support the much wider spectral range. This included, e.g., extending all data types used for storing input spectral atmospheric data, implementing the Cornette-Shanks phase function, and extending the output image format. More importantly, we had to also modify the way the path tracer samples wavelengths and stores results. For the Prague Sky Model, `atmo_sim` rendered 11-channel images, i.e., it splatted MC samples into 11 wavelength bins of 40 nm width, in the range from 320 nm to 760 nm. In order to cover the wavelength range up to 2500 nm, we added 43 more such channels from 760 nm to 2480 nm. We also added one more such channel on the ultraviolet side of the spectrum, as there is some residual solar radiation as

well: so our extended model covers wavelengths from 280 nm to 2480 nm using 55 regularly spaced bins. This is not a very high spectral resolution, at least not by the standards of atmospheric science: but as `atmo_sim` splats MC samples into the spectral bins via a tent kernel, spectral aliasing is kept low, and the overall energy of the result spectra is maintained.

As expected, rendering 55-channel images of the more absorbing atmosphere significantly lowered the path tracer performance (in some case it was up to 20× slower). To at least partially alleviate this slowdown, we increased the original number of 4 simultaneously traced Hero wavelength samples to 16. Please note that the worse performance of the SWIR-capable `atmo_sim` only affects the brute force pre-computation step. Also, a sizeable part of this slowdown is intrinsic to the problem at hand (far higher atmospheric absorption in certain wavelength bands outside the visible range), so that the lower rendering speeds for a SWIR solution are actually a good argument to use a pre-computed model like ours in the first place.

Validation

In Appendix 3.12.2, `atmo_sim` simulations are validated against measurements provided by Kider et al. [2014], results obtained with `libradtran`, and also against empirical observations. The validations show that `atmo_sim` simulation provides results which are highly physically plausible and suitable for generating the reference datasets.

3.6.2 Reference datasets

The datasets used to fit the Prague Sky Model and its SWIR extension consist of renderings which systematically cover the parameter space described in Section 3.4. Rendered values for the parameters can be found in Table 3.2.

Value selection

The ground albedo parameter range is covered uniformly by four samples. Observer altitude and solar elevation samples were initially distributed exponentially to sample the ranges more densely near the ground and horizon, respectively. These initial distributions were then iteratively refined by running test renderings and analysing how fast sky dome features change between each two samples. As a result, the observer altitude parameter is additionally sampled at the edge of the inversion layer at 2 km and solar elevation is uniformly densely sampled below the horizon.

Each visibility is associated with corresponding aerosol concentration profiles, as explained in Section 3.5.2. Three visibilities are given by the measured data available in OPAC: 27.6 km, 59.4 km and 131.8 km. By their extrapolation and interpolation (also described in Section 3.5.2) three more visibilities were obtained: 20.0 km, 40.0 km and 90.0 km. Without more measured data any further extrapolation was deemed ad hoc (e.g., visibility 131.8 km already corresponds to zero ground level concentration of the SOOT aerosol so it is not clear how to extrapolate further).

	Rendered values
Ground albedo	0.00, 0.33, 0.66, 1.00
Observer altitude (metres)	*Prague Sky Model*: 0.00, 1.87, 15.00, 50.62, 120.00, 234.38, 405.00, 643.12, 960.00, 1366.90, 1875.00, 2000.00, 2495.60, 3240.00, 4119.40, 5145.00, 6328.10, 7680.00, 9211.90, 10935.00, 12861.00, 15000.00 *SWIR extension*: 0.00
Solar elevation (degrees)	-4.20, -4.00, -3.50, -3.00, -2.50, -2.00, -1.50, -1.00, -0.50, 0.00, 1.00, 2.85, 5.23, 8.05, 11.25, 14.79, 18.64, 22.77, 27.17, 31.82, 36.71, 41.83, 47.16, 52.71, 58.46, 64.40, 70.53, 76.84, 83.34, 90.00
Visibility (kilometres)	20.0, 27.6, 40.0, 59.4, 90.0, 131.8
Wavelength bins (nanometres)	*Prague Sky Model*: 320 – 760 by 40 *SWIR extension*: 280 – 2480 by 40

Table 3.2: Parameter values used for rendering the reference datasets used for fitting the Prague Sky Model and its SWIR extension. If no model is stated for a parameter, the same values were used for both. Note the uneven sampling of observer altitude and solar elevation that ensures more samples are taken in areas of the parameter space where large changes in sky dome features occur.

The Prague Sky Model uses the same 11 wavelength bins of 40 nm width as Hošek and Wilkie [2012], the SWIR extension adds 44 more as discussed above. Experiments with wider bins led to noticeable colour shifts, narrower bins yielded no tangible benefits. If nanometer-level spectral accuracy were desired, one would need to resort to direct computations via `libradtran`, or a similar specialised tool: pre-computing a sky dome model to such a fine-grained spectral resolution would require prohibitive amounts of storage space.

Image format

In order to provide in-scattered radiance data even for observer positions above ground level, the Prague Sky Model has to fit a fully spherical function instead of a hemispherical one common to previous work. Using one up-facing and one down-facing fish-eye view of the sky as the input to the fitting algorithm would lead to discontinuities at the horizon. Therefore, *side-facing fish-eye views* are used instead. Moreover, since the in-scattered radiance data are symmetrical left and right of the sun position for the idealised planet used by the model, only a single side-facing view is needed as long as it looks exactly 90° away from the sun set position. Figure 3.16 shows an example of these side-facing views for one particular ground albedo, solar elevation and visibility, and increasing observer altitudes. One such side-facing fish-eye view has to be generated for every parameter value combination and each of the four Stokes vector components.

For the transmittance fit, one more image has to be rendered for each combination of observer altitude, visibility and wavelength (transmittance depends

neither on ground albedo nor solar elevation). This image is not a fish-eye view of the sky, instead it stores transmittance for the 2D distribution of points in the atmosphere illustrated in Figure 3.15. Figure 3.17 shows three such images.

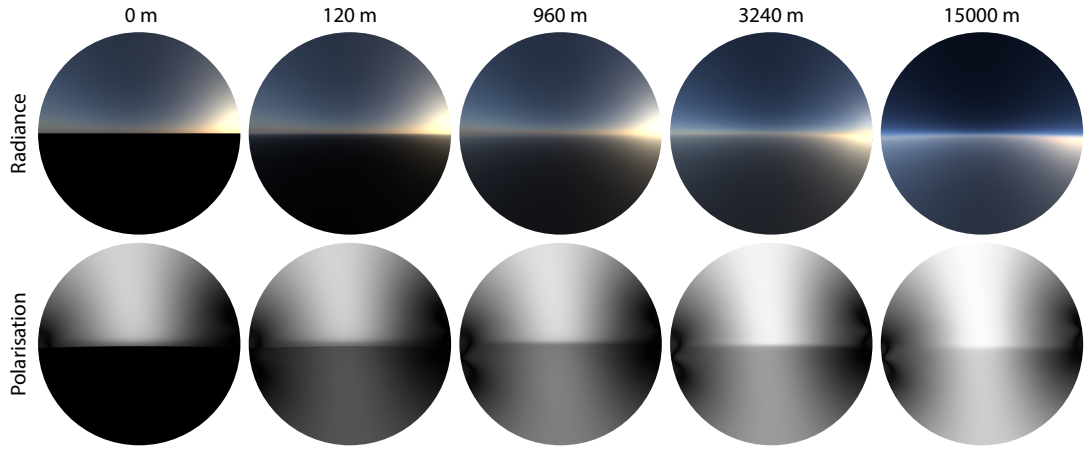


Figure 3.16: An example of side-facing fish-eye views of the sky used in fitting the model. They show in-scattered radiance and polarisation for ground albedo 0.33, solar elevation 8.05°, visibility 20 km, and observer altitudes (from left to right) 0, 120, 960, 3240 and 15 000 m. Since ground reflections are intentionally omitted, the bottom half of the images starts completely black at ground level and then changes with increasing observer altitude due to increasing amount of in-scattered radiance. Radiance shown in the top row corresponds to the first Stokes vector component, polarisation shown in the bottom row corresponds to the second Stokes vector component (after the alignment illustrated in Figure 3.19).

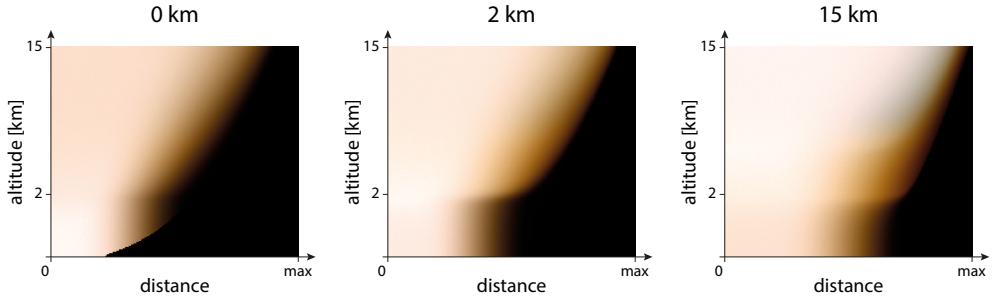


Figure 3.17: Three examples of the reference transmittance images rendered for observer altitudes 0, 2, 15 km, and visibility 59.4 km. The images store transmittance for the 2D distribution of points defined in Section 3.6.1, the image height corresponds to altitude of the points *alt* (in range 0 – 15 km), the image width corresponds to distance of the points along the Earth's surface *d* (in range 0 – distance of the furthest visible point). For this figure, the images were created by combining all visible wavelength channels into a single sRGB image, the brown tint corresponds to the more prominent scattering of blue light in the atmosphere. Note how transmittance decreases with increasing distance (horizontally from left to right) and with increasing difference in altitude of the points from the rendered observer altitude (vertically away from it). The sudden change of shape for points at 2 km corresponds to the end of the inversion layer.

Dataset rendering and statistics

For the Prague Sky Model, 690 960 side-facing fish-eye images (4 ground albedos × 22 observer altitudes × 30 solar elevations × 6 visibilities × 11 wavelengths × 4 Stokes vector components) and 1 452 transmittance images (22 observer altitudes × 6 visibilities × 11 wavelengths) had to be rendered (in resolution 512 × 512 and 202 × 172, respectively). This was achieved by running `atmo_sim` in 3960 render jobs on a scientific supercomputing cluster (TACC at The University of Texas at Austin), each of which gave fish-eye images for 4 ground albedos × 11 wavelengths × 4 Stokes components, and in additional 132 jobs, each of which gave transmittance images for 11 wavelengths. In order to obtain a low level of noise in the images, 200 thousand samples per pixel were used for fish-eye images with post-sunset solar elevations, and 100 thousand samples per pixel for the remainder. As a result, one render job took up to 600 core hours and rendering the entire dataset required about 1.5 million core hours. The resulting dataset size is 530 GB in uncompressed form, 200 GB when ZIP compressed.

We took similar approach when rendering the dataset for the SWIR extension. This time only 158 730 images were rendered (i.e., more than 4 × less) but more samples per pixel were needed because of strong absorption is certain wavelength bands outside the visible range. Therefore, rendering the entire dataset required about 800 thousand core hours (i.e., only 2 × less). The resulting dataset size is 122 GB in uncompressed form, 46 GB when ZIP compressed.

3.6.3 Author's contribution

Apart from all the changes in `atmo_sim` and ART necessary for the SWIR extension, the author was also responsible for the optimizations of `atmo_sim` during development of the Prague Sky Model. The author implemented the next event estimation and multiple importance sampling (only phase function sampling was used before), Null Scattering and Hero wavelength sampling (only single wavelength tracing was used before), and the reversed attenuation computation which avoids the rotation of Mueller matrices. All these optimizations greatly reduced noise in rendered images and were essential for feasibility of rendering the reference datasets. The author was also responsible for optimizing the selection of parameter values for rendering as well as for executing the whole rendering for both models and assembling the reference datasets.

3.7 Creation of the fitted model

In the following sections we discuss the fitted components of the Prague Sky Model and its SWIR extension: in-scattered radiance, transmittance, and polarisation.

3.7.1 In-scattered radiance

As the inner workings of the in-scattered radiance fitting are rather complex and not the author's contribution, their detailed explanation is given in Appendix 3.12.3, and only the main characteristics are presented here.

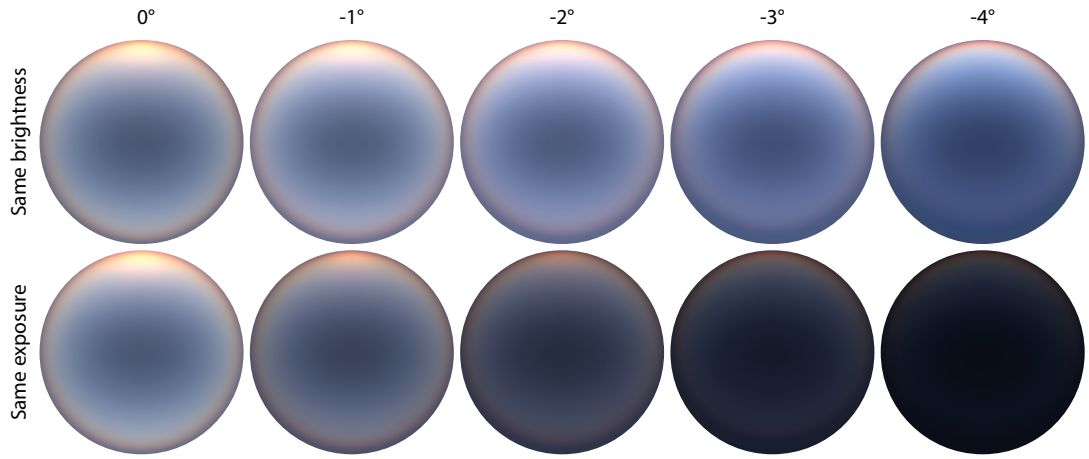


Figure 3.18: Radiance patterns in the post sunset sky for visibility 59.4 km. **From left to right**: Solar elevations 0° to −4°. **Top row**: Images tone mapped individually for similar overall brightness. They reveal the shadow of the Earth rising opposite the setting sun and then becoming less and less distinct as it grows into night. **Bottom row**: The same images all tone mapped using the same exposure as solar elevation 0° to show the actual decrease of brightness. Note that up-facing fish-eye views are used in this figure, as these give a better view of the complete radiance pattern. For the actual model fitting, side-facing fish-eye views were used, as discussed in the text.

Because of the large number of appearance features that are present on a fully spherical sky dome, especially if post-sunset scenarios are included (see Figure 3.18 for examples of these), an entirely new fitting approach was developed that is distinct from previous techniques. In the Prague Sky Model and its SWIR extension, the radiance patterns of the sky are obtained as a sum of outer products of single variable functions. The functions themselves are free-form, tabulated and were obtained by Canonical Polyadic Decomposition (CPD) [Kolda and Bader, 2009], a process very similar to SVD low rank approximation. This approach can be thought of as a specialised compression scheme, however it is also essentially a decomposition of the radiance patterns into an optimal orthogonal set of "features".

CPD performance critically depends on using a suitable input parametrisation that allows the separation to take place cleanly. A suitable scheme described in Appendix 3.12.3 is based on the solar angle plus the shadow and zenith angles, which makes the gradient of the solar glow and the shadow/horizon lines aligned with both axes of the fish-eye input images that are re-projected to this *tensor space*. There, they are expressed as an outer product of two vectors using the CPD decomposition, and the two vectors are stored for later retrieval and reconstruction.

Subtle changes to sky dome appearance both with changing observer altitude and with the sun going beneath the horizon actually require further refinements to this basic idea, like the high altitude correction and the image pre-emphasis to improve horizon interpolation and decomposition of post-sunset images which are described in Appendix 3.12.3. It has to be noted that while the basic idea of using CPD to handle a dataset like the one used by the models is conceptually

simple, the approach would not have yielded a useful result without these further refinements.

A crucial aspect of the new fitting approach is also the interpolation, i.e., how images are reconstructed from the tabulated functions for parameter combinations that were not in the reference dataset. Naive reversing of the fitting process for the nearest available combinations and then interpolating between them in image space would lead to substantial artefacts (e.g., two half-visible suns or horizons in a single image). Instead, the nearest combinations are first reconstructed in the tensor space and then projected into image space using the *target* parameters. This way artefact-free images are obtained for any combination of parameters from the entire parameter space of the model.

3.7.2 Transmittance

Similarly to using CPD for in-scattered radiance fitting, Singular Value Decomposition (SVD) is used for transmittance fitting. Again, to keep features as axis-aligned as possible, a suitable parametrisation is used: in this case the (alt, d) parametrisation described in Section 3.6.1. Since the reference transmittance dataset was already rendered in this parametrisation, no projecting is necessary and SVD can be directly used to produce a low rank approximation of these pre-computed data. The only preprocessing step before the fitting is a non-linear transformation of the data via a square root to boost small transmittance values, which allows using a lower rank approximation than if untransformed data were used. The SVD then produces a sorted list of singular vectors and values $U\Sigma V^*$ for each observer altitude a. Only the first $R = 12$ bases $U_a(R)$ are kept, with associated coefficients $C_a(R) = \Sigma_a(R)V_a^*(R)$.

During reconstruction, the transmittance $Tr(\lambda, \mathbf{y} \rightarrow \mathbf{x})$ between two points in the atmosphere $\mathbf{x}$ and $\mathbf{y}$ at a given wavelength λ is computed from the reduced rank approximation. First, two observer altitudes a_1 and a_2 nearest to $\mathbf{x}$ are located and their stored bases $U_{a_1}(R)$ and $U_{a_2}(R)$ and coefficients $C_{a_1}(R)$ and $C_{a_2}(R)$ are retrieved. Then $\mathbf{y}$ is projected into the 2D parametrisation, leading to $alt_{\mathbf{y}}$ and $d_{\mathbf{y}}$. The inner products between the bases evaluated at $alt_{\mathbf{y}}$ and $d_{\mathbf{y}}$ and the coefficients are then computed for both observer altitudes a_1 and a_2:

$$Tr_{a_i}(\lambda, \mathbf{y} \rightarrow \mathbf{x}) = \langle U_{a_i}(R)(alt_{\mathbf{y}}, d_{\mathbf{y}}) | C_{a_i}(R) \rangle^2 \tag{3.7}$$

This value is interpolated between the two altitudes resulting in the required transmittance value $Tr(\lambda, \mathbf{y} \rightarrow \mathbf{x})$.

3.7.3 Finite distance in-scattered radiance

Thanks to the fittings described in the previous two sections, the Prague Sky Model and its SWIR extension allow calculating in-scattered radiance for *infinite* paths and transmittance for *finite* paths. This section shows, how these quantities can be used for computing in-scattered radiance for *finite* paths.

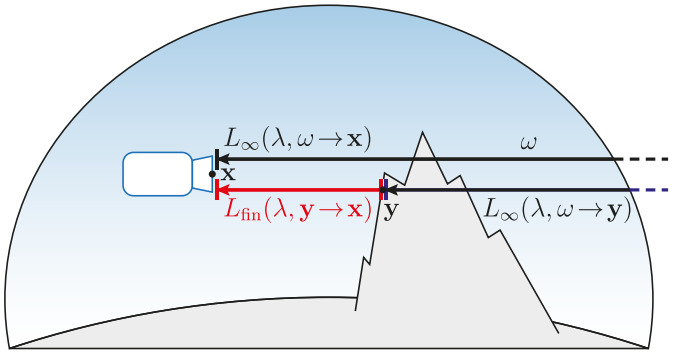
When expressed using notation introduced with the radiative transfer equation (3.1) in Section 3.3, for any observer location $\mathbf{x}$ and a direction ω of a ray incoming from infinity, the models provide total in-scattered radiance along this

infinite ray $L_\infty(\lambda, \omega \to \mathbf{x})$ and transmittance $Tr(\lambda, \mathbf{y} \to \mathbf{x})$ for any position $\mathbf{y}$ on the ray. $L_\infty(\lambda, \omega \to \mathbf{x})$ is given by

$$\begin{aligned} L_\infty(\omega \to \mathbf{x}) &= \int_{\mathbf{x}}^{\infty} Tr(\mathbf{y}' \to \mathbf{x}) L_\mathrm{i} \, \mathrm{d}\mathbf{y}' \\ &= \int_{\mathbf{x}}^{\mathbf{y}} Tr(\mathbf{y}' \to \mathbf{x}) L_\mathrm{i} \, \mathrm{d}\mathbf{y}' + \int_{\mathbf{y}}^{\infty} Tr(\mathbf{y}' \to \mathbf{x}) L_\mathrm{i} \, \mathrm{d}\mathbf{y}' \\ &= \int_{\mathbf{x}}^{\mathbf{y}} Tr(\mathbf{y}' \to \mathbf{x}) L_\mathrm{i} \, \mathrm{d}\mathbf{y}' + Tr(\mathbf{y} \to \mathbf{x}) \int_{\mathbf{y}}^{\infty} Tr(\mathbf{y}' \to \mathbf{y}) L_\mathrm{i} \, \mathrm{d}\mathbf{y}' \\ &= L_\mathrm{fin}(\mathbf{y} \to \mathbf{x}) + Tr(\mathbf{y} \to \mathbf{x}) L_\infty(\omega \to \mathbf{y}). \end{aligned} \tag{3.8}$$

For clarity, we dropped wavelength and L_i arguments and also emphasized a change of transmittance arguments in the second integral between the second and third equation by the red colour. $L_\mathrm{fin}(\lambda, \mathbf{y} \to \mathbf{x})$ in (3.8) is the wanted finite in-scattered radiance between $\mathbf{x}$ and $\mathbf{y}$ and can be therefore computed as

$$L_\mathrm{fin}(\lambda, \mathbf{y} \to \mathbf{x}) = L_\infty(\lambda, \omega \to \mathbf{x}) - Tr(\lambda, \mathbf{y} \to \mathbf{x}) L_\infty(\lambda, \omega \to \mathbf{y}). \tag{3.9}$$



This formula also works for all components of polarised radiance in a Stokes vector.

For practical use, it has to be noted that (3.9) only holds when the two L_∞ values used in it are both highly numerically accurate. However, in the fitted models, for any two distinct query locations, there can sometimes be small radiance discrepancies compared to the ground truth. In rendered images, these deviations will manifest themselves as horizontal stripe artefacts for finite viewing distances, usually for observer altitudes near the ground, and at viewing angles near the horizon.

To solve these stability issues, one has to consider the effect the fitting has on sky dome radiance patterns near the horizon. As shown in Figure 3.28, the most visible change compared to the original brute force images is that the models slightly blurs the horizon region. What is not immediately obvious is that it does so within a small and not entirely predictable blur range. For directly observed radiance, these small variations in blur are imperceptible: but for the above mentioned viewing geometries, (3.9) still suffers numerical stability issues because of it.

As there is no way to get rid of the existing variable blur near the horizon region, a workable alternative is to *always actively bring all such lookups to a consistent minimum level of blurriness*. (3.9) becomes stable if the data does not exhibit small random variations due to the fitting: and this can easily be achieved by not taking a single sample in the exact path direction, but the average of a few directions slightly above and below ω instead.

Use in a path tracer With $L_\infty(\lambda, \omega \to \mathbf{x})$, $L_{\text{fin}}(\lambda, \mathbf{y} \to \mathbf{x})$ and $Tr(\lambda, \mathbf{y} \to \mathbf{x})$, the models provide everything a path tracer needs to evaluate the radiative transfer equation (3.1). For a ray leaving a scene into the sky, radiance of the ray satisfies:

$$L_{\text{o}}(\lambda, \omega \to \mathbf{x}) = L_\infty(\lambda, \omega \to \mathbf{x}). \tag{3.10}$$

For a ray leaving a scene into the sun:

$$L_{\text{o}}(\lambda, \omega \to \mathbf{x}) = L_\infty(\lambda, \omega \to \mathbf{x}) + Tr(\lambda, \mathbf{y} \to \mathbf{x}) L_{\text{Sun}}(\lambda) \tag{3.11}$$

for $\mathbf{y}$ located on the ray at the atmosphere boundary. And for a ray hitting a geometry at $\mathbf{y}$:

$$L_{\text{o}}(\lambda, \omega \to \mathbf{x}) = L_{\text{fin}}(\lambda, \mathbf{y} \to \mathbf{x}) + Tr(\lambda, \mathbf{y} \to \mathbf{x}) L_{\text{r}}(\lambda, \mathbf{y} \to \omega), \tag{3.12}$$

where the path tracer follows reflection at $\mathbf{y}$ to obtain $L_{\text{r}}(\lambda, \mathbf{y} \to \omega)$.

3.7.4 Polarisation

A polarisation fit could be obtained by fitting the three additional Stokes vector components using the same approach as for in-scattered radiance. However, several optimizations were made.

First, sky dome polarisation is almost entirely linear, the fourth component was therefore omitted. Second, using a suitable rotation of pixel reference frames illustrated in Figure 3.19, the third component became very weak and was omitted as well. This idea of the reference frame re-alignment was already used by Wilkie et al. [2004] but without any solid evidence that this was actually permissible. During rendering of the reference dataset for the Prague Sky Model it was verified that the remaining signal in the third component has no perceptible effect for any parameter combination (even in scenes that otherwise exhibit visual differences due to sky dome polarisation). The polarisation information encoded in the modified second component has to be rotated back before use, but otherwise it is a good approximation of the dominant polarisation features in the sky. Finally, since the polarisation patterns are much simpler than the radiance patters, the rank of CPD was reduced (to $n = 5$ instead of $n = 9$) as well as the sizes of the individual function tables.

3.7.5 Complete fitted datasets

Each of the approximately 350 thousand images (only 2 Stokes vector component had to fitted) for the Prague Sky Model and 80 thousand for the SWIR extension was fitted separately, which took about 1 core hour per fit and was carried out on the same supercomputing cluster as rendering of the reference dataset. The output of the fitting is a set of coefficients for each of the fitted image (about 400 coefficients per image), we call it a fitted dataset.

The size of a complete fitted dataset for the Prague Sky Model is 375 MB per visibility: 242 MB radiance data + 6 MB transmittance data + 127 MB polarisation data, i.e., 2.25 GB in total. Reduced variants of the dataset can be created easily, e.g., non-polarising version (1.5 GB) or even ground level only version (103 MB). The total size of a complete fitted dataset for the SWIR extension is 550 MB.

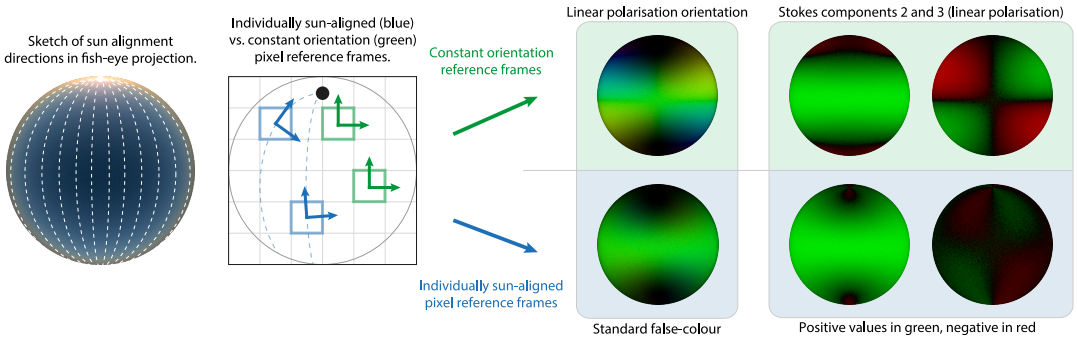


Figure 3.19: This figure illustrates sun-aligned polarization reference frames which allow omitting the third Stokes vector component when describing linear sky dome polarisation. While the third component after the alignment is not zero, it is negligible as it is caused only by multiple polarising scattering events, which are rare. Note that in this figure the concept is demonstrated with up-facing fish-eye views, as these give a view of the complete polarisation pattern. For the actual model fitting, side-facing fish-eye views were used, as discussed in the text: the re-orientation works the same in either case. Standard reference frames are shown in green, individually aligned ones in blue. The reference frames are right-hand coordinate systems, as the light is coming from the image plane towards the observer. The colour scheme used in the "Standard false colour" images is a linear polarisation orientation plot provided by `polvis` (see Section 3.6.1). *Note: this figure taken from the paper of Wilkie et al. [2021] also appeared in the doctoral thesis of Hošek [2019] (on page 71).*

When compared to sizes of the brute force rendered reference datasets that entered the fitting (530 GB for the Prague Sky Model and 122 GB for the SWIR extension), the compression ratio of the fitting process is more than 220, which is about 100× better than what ZIP compression achieved on these data.

3.7.6 Modifications for the SWIR extension

Since reference images corresponding to different wavelength bins are fitted independently, modifying the fitting to work with the extended spectral range was straightforward. The only issue we encountered were artefacts caused by higher levels of noise in the reference images. As we discuss in Section 3.9.4, some wavelengths are more difficult to render because less light is transported in these spectral bands. As a result, there is more noise in the corresponding images, which makes the fitting less stable. To avoid undesirable artefacts, we had to increase the strength of the filtering step that is applied before the fitting in these cases. Figure 3.20 shows an example in which this modification successfully removed artefacts from the resulting fit.

Wavelength compression

Using 5× more wavelength channels than the Prague Sky Model does not come for free. Besides slowing down the generation of the reference dataset and its fitting, it also means a 5× larger final dataset that constitutes the model. This dataset has to be stored on disk and then loaded into computer memory before

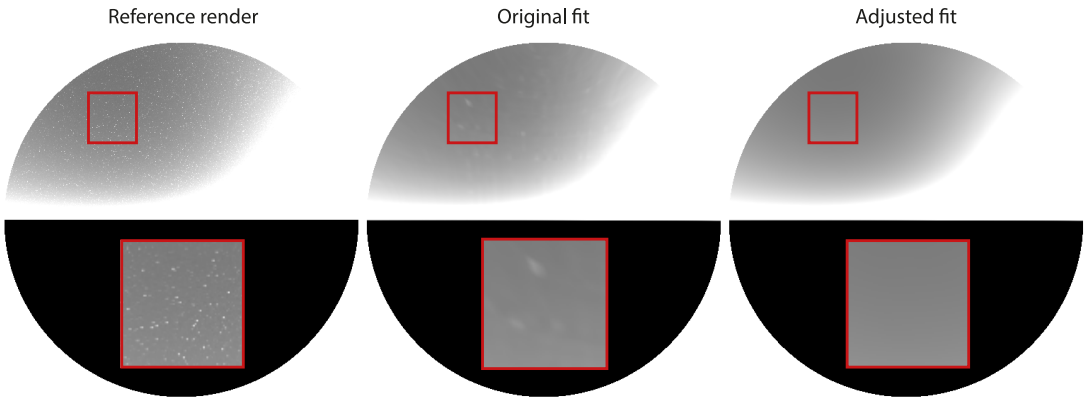


Figure 3.20: An example of a successfully removed noise from a fitted image thanks to our slightly increased filtering in the fitting preprocessing step. **Left:** A reference side-facing fish-eye image rendered for visibility 59.4 km, solar elevation 0°, ground albedo 0.5 and wavelength channel 2400 – 2440 nm. **Middle:** A fit obtained with the original code of the Prague Sky Model. **Right:** A fit after increasing the filtering.

use. Therefore, we also investigated the possibility of compressing the dataset by omitting some of the 55 channels we computed.

We tested a simple greedy algorithm for selecting unnecessary channels. We kept the first 12 channels (280 – 760 nm) fixed and then tried leaving out immediately following channels as long as the maximum relative error caused by replacing them by interpolated values does not exceed a fixed threshold. Once that happens, the last tested wavelength is kept and the omitting starts from the next one. Figure 3.21 shows dataset size reduction we can achieve with this algorithm for different error thresholds. So if a particular application does not require maximum accuracy and can allow e.g. 15% error, it can save almost 22% of the dataset size.

To ensure maximum quality and flexibility at the same time, we supply the complete dataset with all channels (in Section 3.8.1), together with a list of channels that can be omitted for various error thresholds (in Appendix 3.12.5).

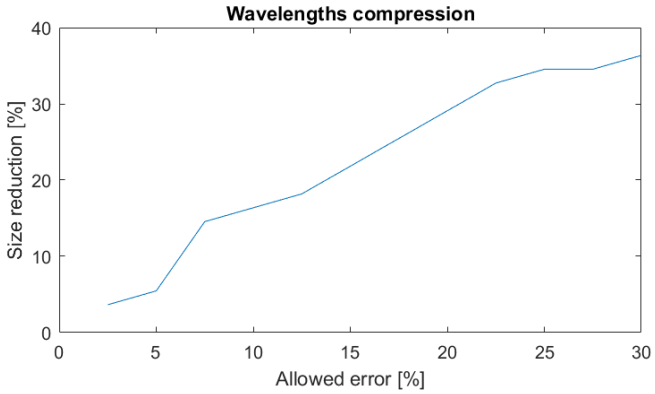


Figure 3.21: A size reduction of the fitted SWIR dataset that is achievable by replacing some of the 55 channels with interpolated values while keeping the introduced error under a selected threshold.

3.7.7 Author's contribution

Apart from the changes in the fitting necessary for the SWIR extension described above, the author was also responsible for tweaking and optimizing the fitting of the Prague Sky Model. From the most important modifications we name two. First, the author improved the projection of images into the tensor space. Originally, this projection was computed in a forward manner, where pixels from a reference fish-eye image were transformed into a tensor image. This led to uneven coverage of the tensor space and produced artefacts on the horizon causing it to look "wavy". The author was able to replace the forward transform by re-projection, i.e., reverse look-up of positions in the fish-eye image for every pixel of the tensor image by implementing a specialized solver for the corresponding system of multivariate quadratic equations. As shown in Figure 3.22, this removed the artefacts completely and thus significantly improved quality of the results provided by the Prague Sky Model.

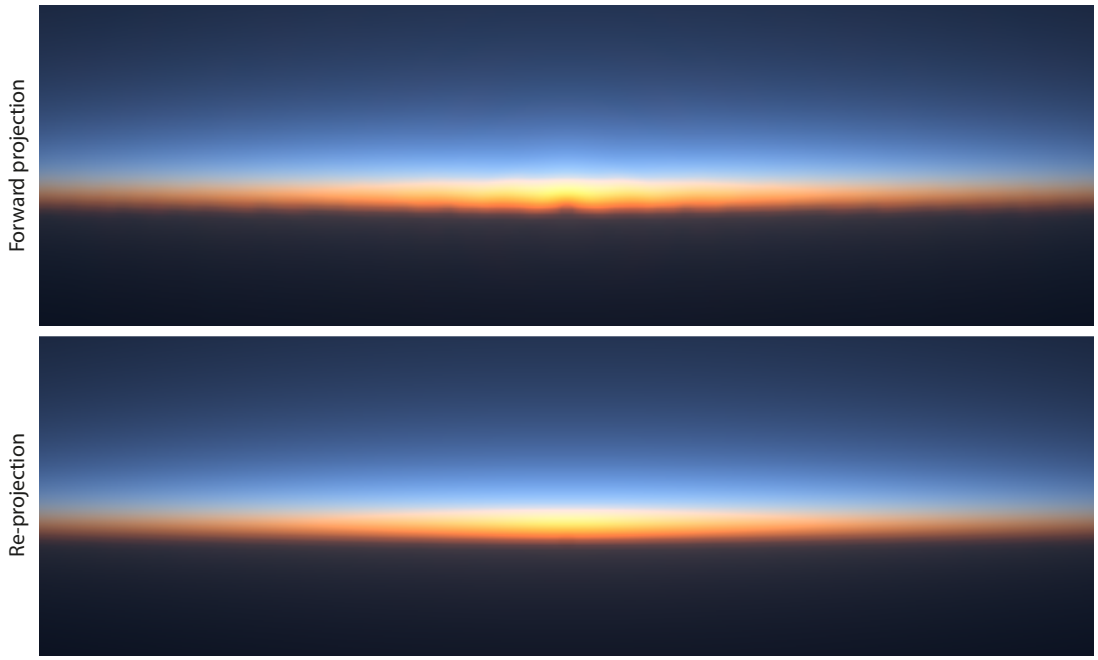


Figure 3.22: An example of fitting artefacts in a close-up view of the horizon produced by the Prague Sky Model for observer altitude 15 km and solar elevation $-4°$. **Top:** A "wavy" horizon produced by a previous version of the fitting using a forward projection of reference images into the tensor space. **Bottom:** A smooth horizon produced by the current version of the fitting using the re-projection described in the text.

Second, the author implemented an additional compression for the fitted radiance datasets using the half-precision floating-point number format. It required adjusting the CPD decomposition during the radiance fitting to produce only non-negative coefficients and their careful scaling, strong artefacts shown in Figure 3.23 were produced otherwise. This reduced size of the fitted radiance datasets to half (sizes listed in Section 3.7.5 are already after this reduction).

The author was also responsible for executing the whole fitting for both models and assembling the fitted datasets.

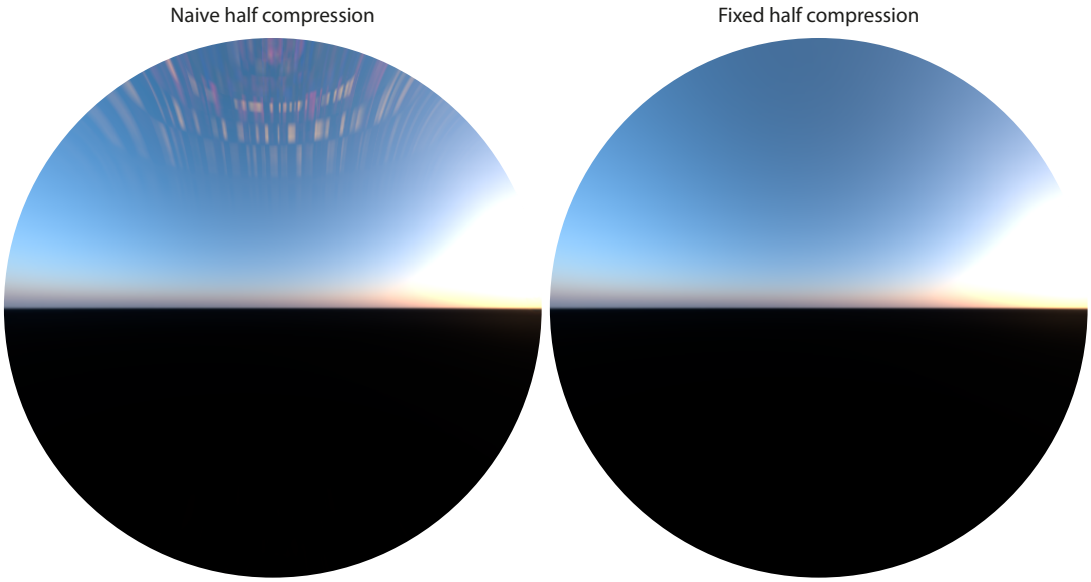


Figure 3.23: An example of compression artefacts in a side-facing fish-eye view of the sky produced by the Prague Sky Model for observer altitude 1.87 m and solar elevation 2.85°. **Left:** Reconstruction artefacts caused by a direct conversion of fitted coefficients of the image from single-precision to half-precision floating-point numbers. **Right:** An artefact-free image achieved by a more careful compression method described in the text.

3.8 Implementation

The final model (either the Prague Sky Model or SWIR extension) shipped to a user consists of two parts: the fitted dataset and a reconstruction code that takes care of retrieving radiance, transmittance and polarisation from the fitted dataset and their interpolation for any combination of values within the parameter ranges described in Section 3.4. There are currently several versions of both the fitted dataset and reconstruction code available.

3.8.1 Fitted datasets

There are four fitted dataset versions: complete, ground level, SWIR and XYZ. The first three datasets can be downloaded from the project GitHub page[1] under the Apache-2.0 license, the XYZ version is proprietary to the Corona renderer and is not publicly available.

Complete (2.25 GB)

A complete dataset containing all features and parameter ranges of the Prague Sky Model including the fully spherical radiance, polarisation and transmittance fits. Limited to the visible spectral range.

[1] `https://github.com/PetrVevoda/pragueskymodel`

Ground level (103 MB)

A dataset limited to ground level radiance and transmittance only, created from the complete one by dropping polarisation and all observer altitudes except for the zero one. As discussed below, incorporating the complete model into an existing path tracer requires some modifications of the target system. On the other hand, this version can be used as a drop-in replacement of whatever hemispherical sky dome model was already present in the path tracer.

SWIR (550 MB)

A dataset of the SWIR extension. Provides the much wider spectral range but is limited to the zero observer altitude only. All other features of the Prague Sky Model are retained, including the polarisation and transmittance fits.

XYZ (226 MB)

While all the three previous versions are spectral and provide at least 11 wavelength channels, this version provides only 3 channels corresponding to the CIE XYZ colour space. It also lacks the polarisation fit, the remaining features are the same as in the complete version. Therefore, the dataset size is much smaller and the model can be more easily used in non-spectral renderers. In fact, this dataset was created specifically for the Corona renderer which operates in an RGB colour space. It was obtained in a similar way as the complete dataset but with an additional step of converting all spectral images in the reference dataset to XYZ prior to the fitting. Note that non-spectral renderers can use the spectral dataset versions as well, the conversion to XYZ is merely an optimization to save reconstruction time and reduce the dataset size.

3.8.2 Reconstruction code

There are three versions of the reconstruction code: ART implementation, Corona implementation and Standalone implementation.

ART implementation

When integrating the Prague Sky Model in an actual path tracing software for testing purposes, two main obstacles were faced. First, evaluation of a precomputed sky dome model which features a full spherical radiance fit that changes with observer altitude requires more than just a drop-in replacement of whatever sky dome model is already present in a given path tracer: making full use of the altitude-dependent capabilities of the model requires modifications to the light source sampling code of the target system. To our knowledge, all existing renderers assume hemispherical models that do not change with altitude when dealing with analytical sky dome radiance. Second, in order to fully test the Prague Sky Model, which is spectral and polarisation capable, the target rendering system also needed to be spectral and polarisation capable.

The spectral requirements narrowed the field down considerably. At the time of developing the Prague Sky Model there were only two fully spectral and polarisation capable renderers available: Mitsuba 2 [Nimier-David et al., 2019] and

ART [Wilkie, 2018]. Since the model already used `atmo_sim`, ART was selected and the reconstruction code was added there. This ART implementation of the Prague Sky Model has then become available as Open Source in the 2.1.1 release of that system.

With its added capabilities such as fully spherical radiance patterns, transmittance and in-scattering for finite distances, the rendering performance of the Prague Sky Model is not easily comparable to previous models like Preetham or Hošek. When the in-scattering computations for finite distances are used, a renderer using the model of course runs slower than with, e.g., pure Hošek sky dome look-ups, and no volumetric computations. But if the model is used to only provide sky dome radiance, it yields results that are qualitatively similar to, but still more realistic than, the Hošek model: see Figure 3.4 for an example of this. And if both the Hošek model and the Prague Sky Model are used in exactly the same render they run at practically the same speed. For instance, we tested both in Figure 3.35 and 3.36, and render times were, e.g., 228 vs. 208 seconds (Hošek was slightly faster), and 124 vs. 126 seconds (slight advantage to the Prague Sky Model), respectively. And that was with the more complex, altitude-dependent light source evaluation code running for the Prague Sky Model: so the actual raw model queries are definitely faster than Hošek.

Please note that this ART implementation is an unoptimized proof-of-concept C code and is capable of loading only the complete and ground level fitted datasets. Nevertheless, it was used for generating all results of the Prague Sky Model in this chapter except for Figure 3.1, 3.2, 3.33, 3.34 and 3.39.

Corona implementation

In 2021 an implementation of the Prague Sky Model was released in version 7 of the Corona renderer and it has been successfully used there to this day. It is a highly optimized implementation designed to work with the XYZ fitted dataset. While Corona is neither spectral nor polarisation capable, it allows rendering of virtually any scenes as opposed to the rather limited range of scenes supported by ART. It was used to render Figure 3.1, 3.33 and 3.34.

Standalone implementation

Since the ART implementation is a rather sub-optimal research code that might be difficult to use and the code of the Corona implementation is not publicly available, we decided to create a clean and optimised C++ 17 implementation that would be sufficiently documented, easy to use and available to everyone. The resulting implementation does not support rendering of arbitrary scenes but it is tiny (only 3 MB large) and standalone (does not require any other rendering or modelling software). It consists of 3 parts: model library, example renderer, and front end.

The library is formed by just two files and everything one needs to use the sky model is in this code. It works with the SWIR dataset as well as with the complete and ground level datasets released for the Prague Sky Model. In comparison with the original implementation, our code significantly lowers memory consumption. Instead of loading the entire dataset into memory and completely unpacking it there, we perform part of the unpacking on demand and provide an option to load

only a part of the dataset needed for rendering a selected configuration. This way we can reduce memory use up to 24 times.

To illustrate how to use the library, we implemented a simple example renderer. It shows how to query the library to render a fish-eye image of the sky. On AMD Ryzen 9 3900X 12-core processor one 1000x1000 pixels large image with all 55 channels takes around 600 ms to render which allows use in interactive applications. This part was used for rendering Figure 3.2 and 3.39.

Finally, for immediate model testing and dataset exploration, we accompany the example renderer with front end with both command-line and graphical user interfaces (GUI). Using the GUI (shown in Figure 3.24) users can easily load datasets, interactively change rendered configurations, and save rendered images.

Our implementation was tested on Windows and Linux and is available for download under the Apache-2.0 licence at the project GitHub page[2].

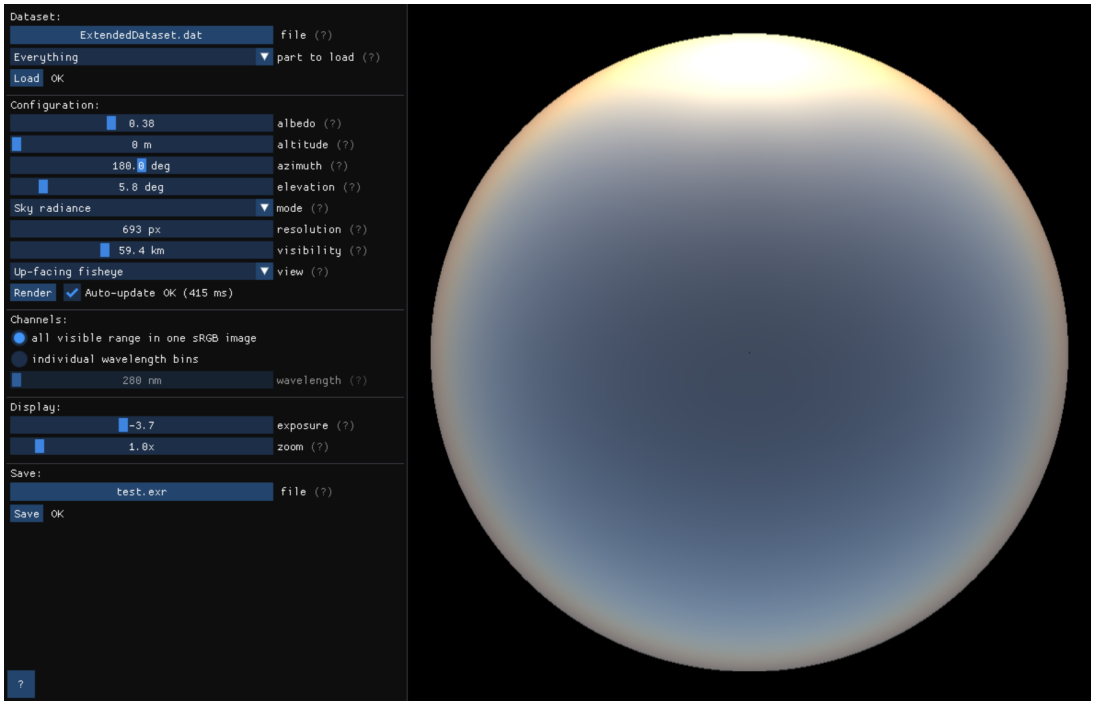


Figure 3.24: A screenshot of the GUI provided by our implementation.

3.8.3 Author's contribution

The author was responsible for creation of all the four fitted dataset versions and the Corona and Standalone implementations.

3.9 Results

After describing all the steps leading to creation of the Prague Sky Model and its SWIR extension we can now proceed with demonstrating their results. We start with the Prague Sky Model and evaluate its fitting accuracy, demonstrate some

[2] `https://github.com/PetrVevoda/pragueskymodel`

of its features and compare it with other sky models. Finally, we present results of the SWIR extension.

3.9.1 Fitting accuracy

To assess the accuracy of the Prague Sky Model fitting, i.e., how well the final model corresponds to brute force reference renderings, both visual comparisons as well as numerical error analysis were carried out.

Visual comparisons

Figure 3.25 shows a comparison of the fitted radiance and polarisation against brute force renderings for a configuration present in the reference dataset. They match closely, the only issue is the slight blur of the horizon and of the negative polarisation values (the black "wings").

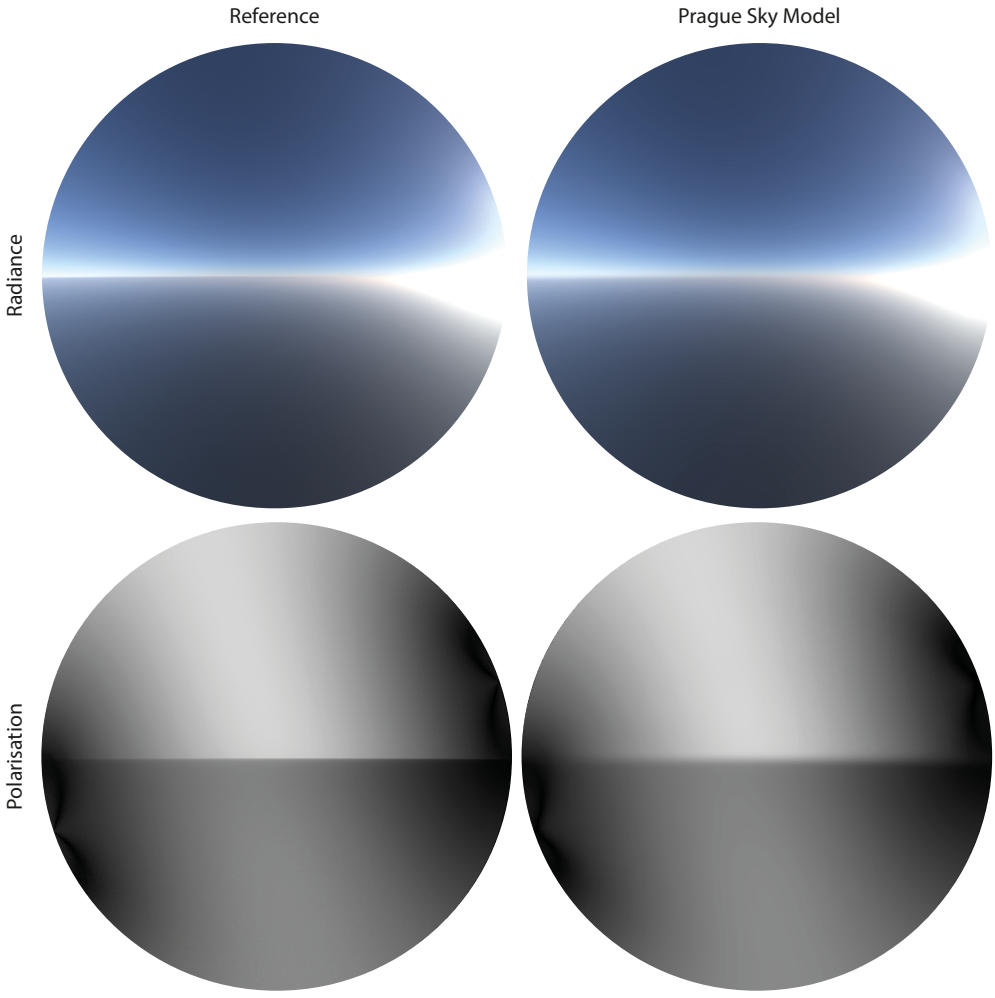


Figure 3.25: An example of how the results of the brute force renderer (left column) and the fitting (right column) match, for radiance (top row) and for the sun-aligned (cf. Figure 3.19) first polarisation component (bottom row). Side-facing fish-eye views, observer altitude 2495 m, solar elevation 18.64°, visibility 59.4 km.

To verify finite distance in-scattered radiance and transmittance as well, a special test scene shown in Figure 3.26 was modelled. Its massive extent makes the effects of finite distance in-scattered radiance and transmittance strong and easy to assess. We refer to this scene as to the Columns scene.

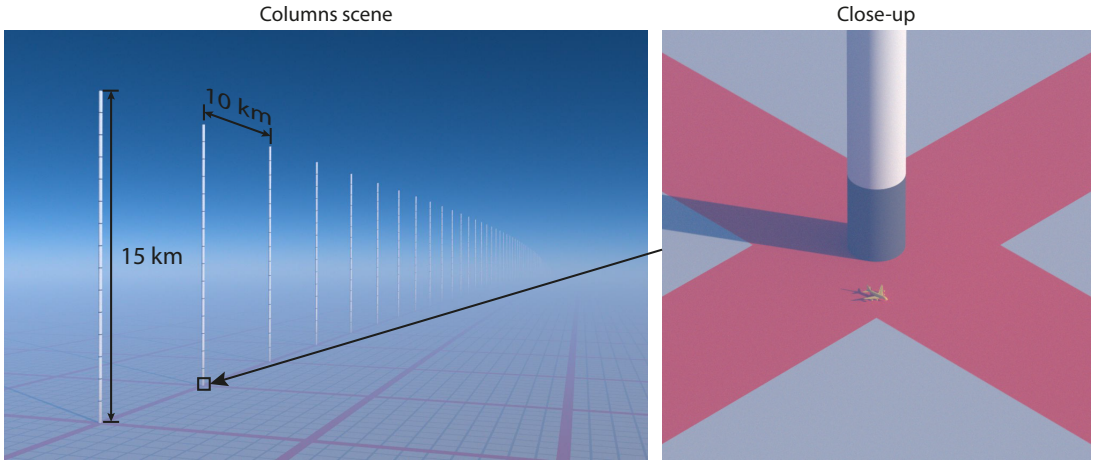


Figure 3.26: An example of the Columns scene used in many of our tests. It contains a planet with a 10×10 km red grid texture, with 15 km high columns with stripes every 1 km. The close-up of one of the column bases shows an Airbus A380 airliner placed next to it for scale. This intends to give a sense of the massive extent of this scene: as it shows a somewhat sterile "atmospheric debugging" set-up, it is easy to under-estimate how large everything seen in it is.

First, the Columns scene is used in Figure 3.27 to test the quality of the transmittance fit. The figure shows that the transmittance fit closely matches the brute force reference, i.e., the model provides correct transmittance for a wide range of distances present in the scene.

Second, in Figure 3.28 the Columns scene is used to compare the complete model against brute force rendering for different solar elevations. It shows that even finite distance in-scattered radiance is matched well. Note that the reference

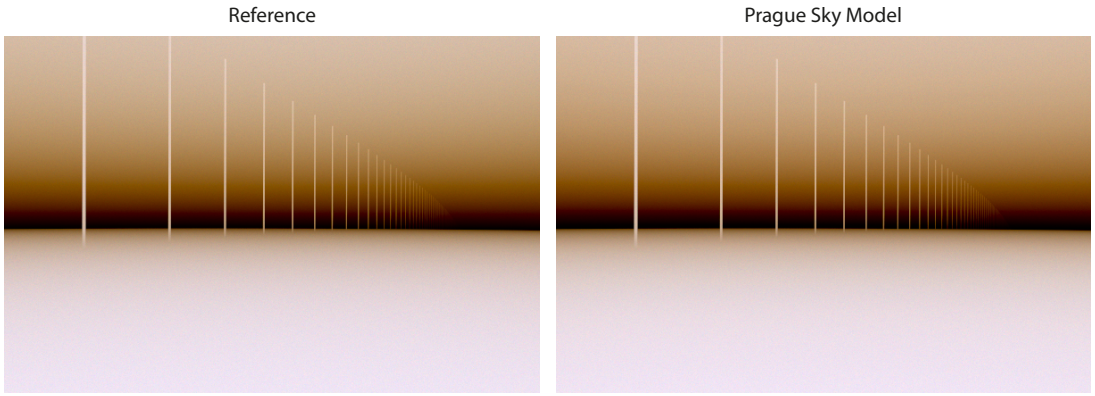


Figure 3.27: Visualisations of the transmittance component for 8 km observer altitude in the Columns scene. The left image is a reference rendering obtained by brute force path tracing, the right is the result obtained by the fitted model. *Note: this figure taken from the paper of Wilkie et al. [2021] also appeared in the doctoral thesis of Hošek [2019] (on page 100).*

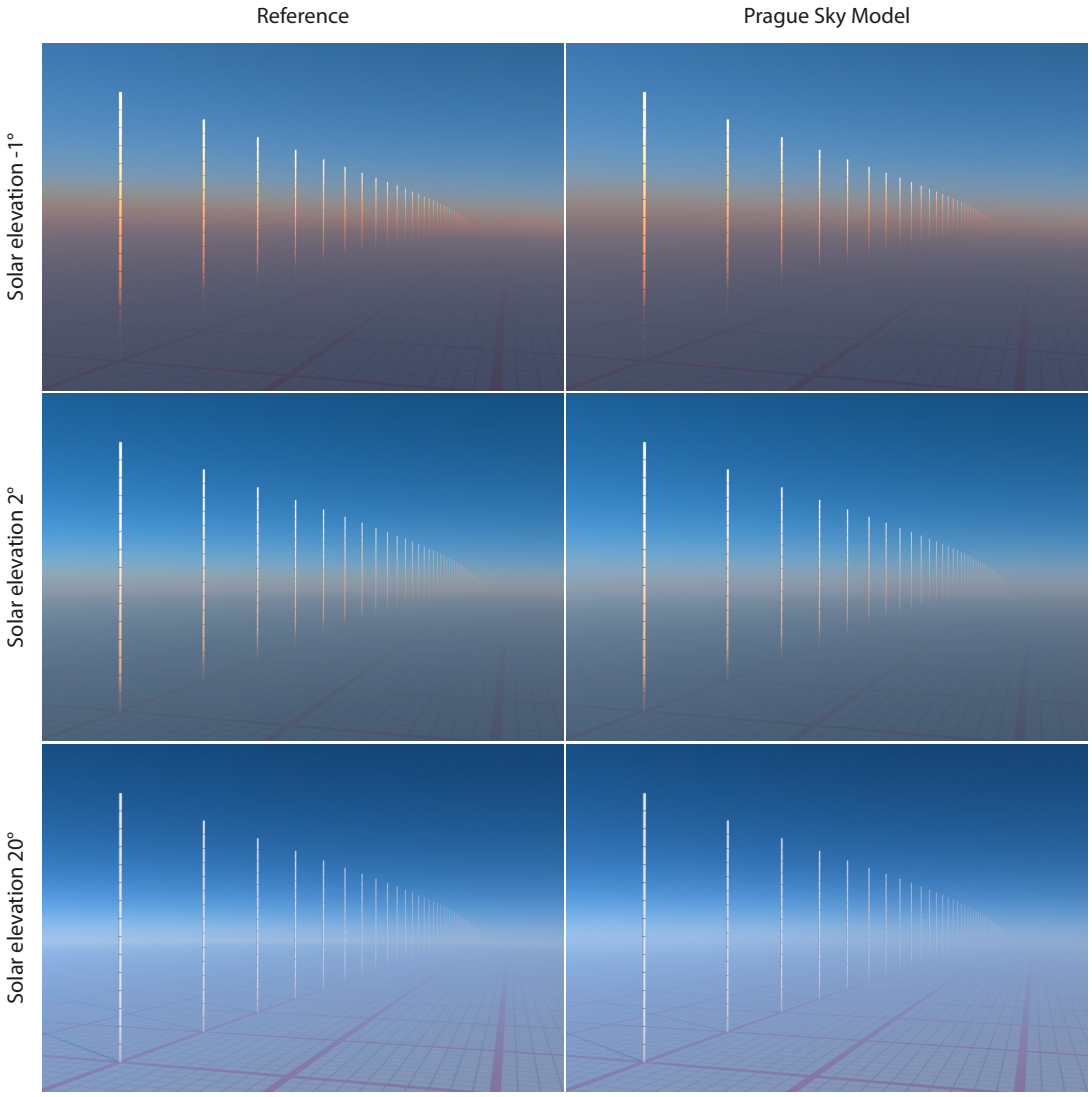


Figure 3.28: A comparison of reference brute force path tracing results from `atmo_sim` (left) to renderings using the Prague Sky Model (right) at solar elevations $-1°$, $2°$ and $20°$ (from top to bottom) in the Columns scene. Observer altitude is 8 km, visibility 59.4 km, the sun is behind the camera.

dataset did not contain solar elevations $2°$ and $20°$, which shows that the Prague Sky Model is capable of interpolations that match the references. Similar to Figure 3.25, the only apparent difference can be seen at the horizon, where the model is more blurred, which is mainly caused by the limited resolution of the fit. However, this blurring mainly affects extreme viewing distances over perfectly flat terrain, and should not lead to artefacts in normal scenes.

One property of a full path tracing solution that is not accounted for in the Prague Sky Model are volumetric shadows. As can be seen in Figure 3.29, these are missing from the image produced by the model. However, a good approximation for these shadows could comparatively easily be computed by simple ray marching, which would still be a lot cheaper than a full path tracing solution. Note that the rendering times in this scene were 10+ hours for the not fully converged reference, but only 4.5 minutes for the model-based rendering.

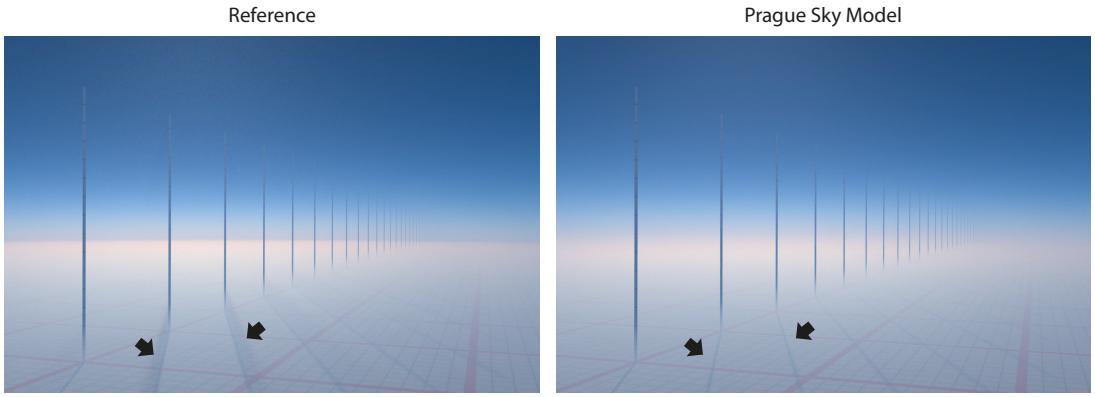


Figure 3.29: The Columns scene rendered for observer altitude 8 km, visibility 59.4 km, with the camera looking towards the sun, which is at 16° elevation, and outside the frame. For this viewing geometry the limitations of just using a pre-computed sky dome radiance model become noticeable. **Left:** A brute force reference rendering done with `atmo_sim`. **Right:** The same scene rendered purely by evaluation of the model. Besides the horizon blur note the absence of volumetric shadows in the ground haze layer (pointed at by the black arrows).

Error analysis

One in eight of all images (every other ground albedo, observer altitude, solar elevation, but all wavelengths and visibilities) were selected for quality control of the obtained fit. This was done via a comparison between the brute force rendered references and renderings that use the fit. This included a manual check for artefacts, but also a systematic SNR analysis. For all images, the minimum SNR was 14.35, the maximum 34.15, and the mean 28.52. The lowest SNR values were obtained for low solar elevations and high observer altitudes, where there is a narrow bright orange wedge on one side of the horizon: in this setting, the added horizon blur causes the most damage. This can be seen in Figure 3.30 which shows one tone-mapped sample of the automatically and systematically generated comparison EXR images which were manually viewed to check for artefacts.

An analysis of the end-to-end error incurred by the whole process from a brute force rendering to a rendering using the model was carried out. It concluded that the following components affect the end result:

- Noise in the brute force rendered reference images: this has some effect, but is limited.
- The direct error incurred by the fitting process: this is the main source of error.
- Error incurred by the dataset compression: these are just the inaccuracies caused by the conversion from double-precision floating-point numbers to half-precision. This introduces some error but the net effect of it is still negligible.
- Noise in the renderings using the fit: this proved to be negligible as well.

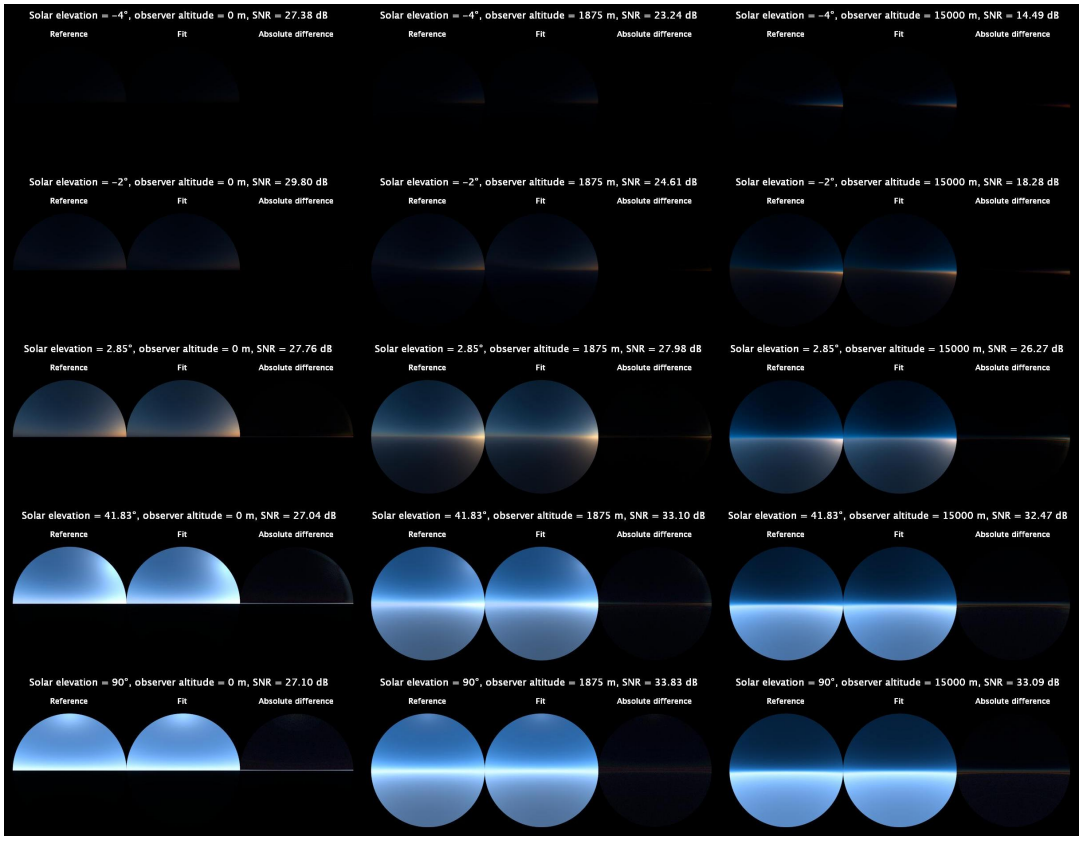


Figure 3.30: A sample of the kind of comparison image which were systematically generated to evaluate the radiance model fit, this one is for visibility 59.4 km. Observer altitudes 0, 1875, 15 000 m (from left to right), solar elevations −4°, −2°, 2.85°, 41.83°, 90° (from top to bottom). Note that the lowest SNR is produced for the high observer altitude and low solar elevation in the upper right corner as discussed in the text.

A systematic analysis of the error incurred by the interpolation between data points which were provided during the fit was also performed. That is, e.g., how far the radiance patterns diverge from the true solution for solar elevations between the ones that were used for the fitting. For a small subset (6) of observer altitudes in the middle of each solar elevation interval, 3 images were compared: brute force rendered (B), fitted (F) from B, and interpolated (I) from fits at interval borders. A ratio RMSE(B, I) / RMSE(B, F) is then 1 at interval borders while in the middle of the intervals it expresses how the fitting error increases because of the interpolation. The maximum of the ratio was 1.52, which was deemed to be acceptable.

In Appendix 3.12.4 plots of the fitting error with respect to each model parameter can be found together with bounds on the interpolation error.

Similar error analysis was done also for the transmittance fit. Maximum absolute error over the transmittance dataset was 0.014. Figure 3.31 shows a sample of reference transmittance images with corresponding fitting results and differences.

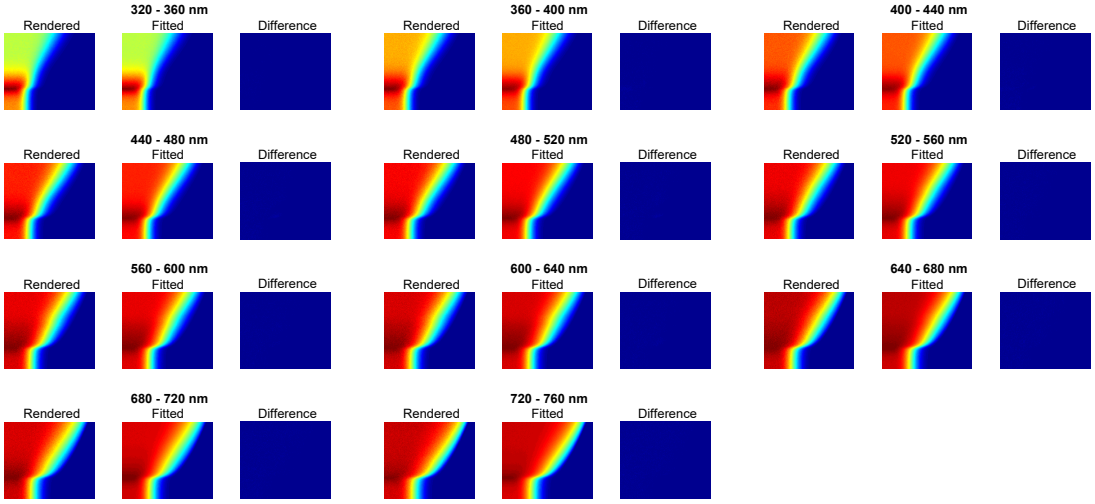


Figure 3.31: A sample of the kind of comparison image which were systematically generated to evaluate the transmittance model fit, this one for observer altitude 2495 m and visibility 59.4 km. Shows false colour reference images with corresponding fitting results and differences for 11 wavelength bins.

3.9.2 Model features

This section demonstrates some of the features of the Prague Sky Model.

Visibility

Results shown in the previous section were all for visibility 59.4 km. Figure 3.32 shows an example of 4 other visibilities out of the entire visibility range from 20 km to 131.8 km provided by the Prague Sky Model.

Post-sunset conditions

One of the benefits of the Prague Sky Model is its support of post-sunset solar elevations. One such example is shown in Figure 3.1, two more are provided in Figure 3.33.

Finite distance in-scattered radiance

Figure 3.34 proves the importance of finite distance in-scattered radiance and transmittance provided by the model. The figure compares the mountain landscape scenes from Figure 3.1 with and without these components and clearly shows they are absolutely crucial for outdoor scene realism.

Performance

The main idea behind fitted sky models is to provide similar quality to brute force rendering at a fraction of time. Figure 3.28 showed that the Prague Sky Model is capable of producing results closely matching brute force rendering. And in deed, it does so in a fraction of time. For the Columns scene shown in the figure at a resolution of 1500×1000, the statistics are as follows: up to 85k spp and 5000 core hours (= 9 days on 24 core CPU) were used for the reference renders, with

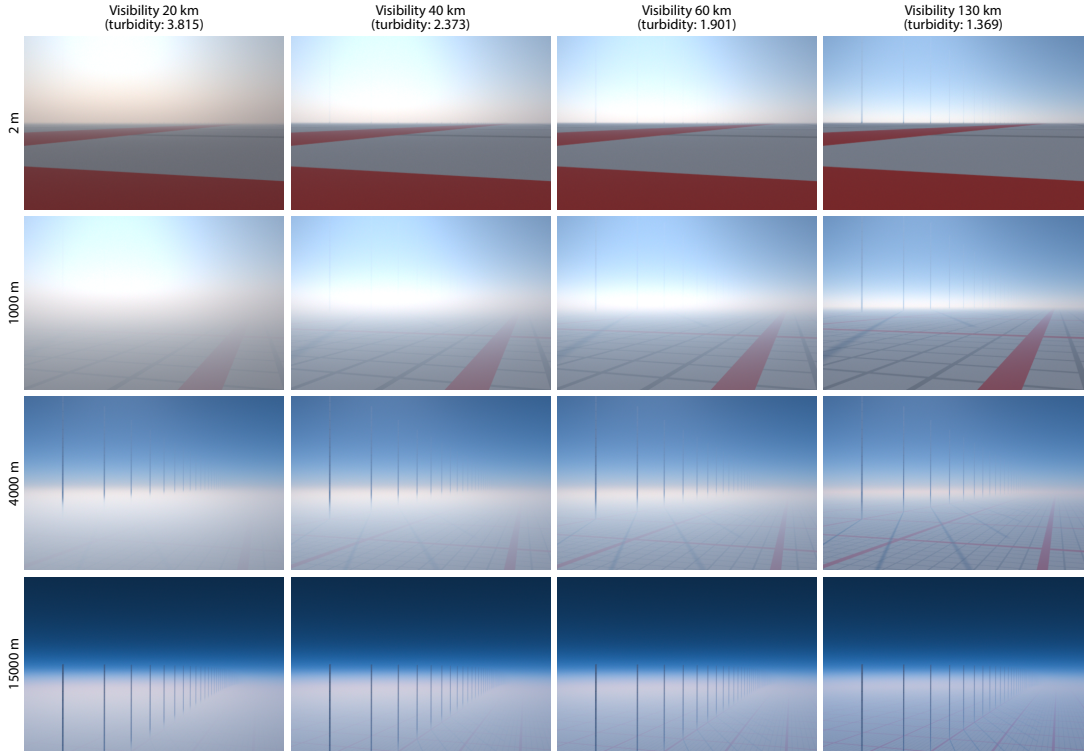


Figure 3.32: An example of 4 different visibilities provided by the Prague Sky Model in the Columns scene. Observer altitudes 2, 1000, 4000 and 15 000 m (from top to bottom), solar elevation 16°. Note how the inversion layer becomes more and more transparent for higher visibility ranges.

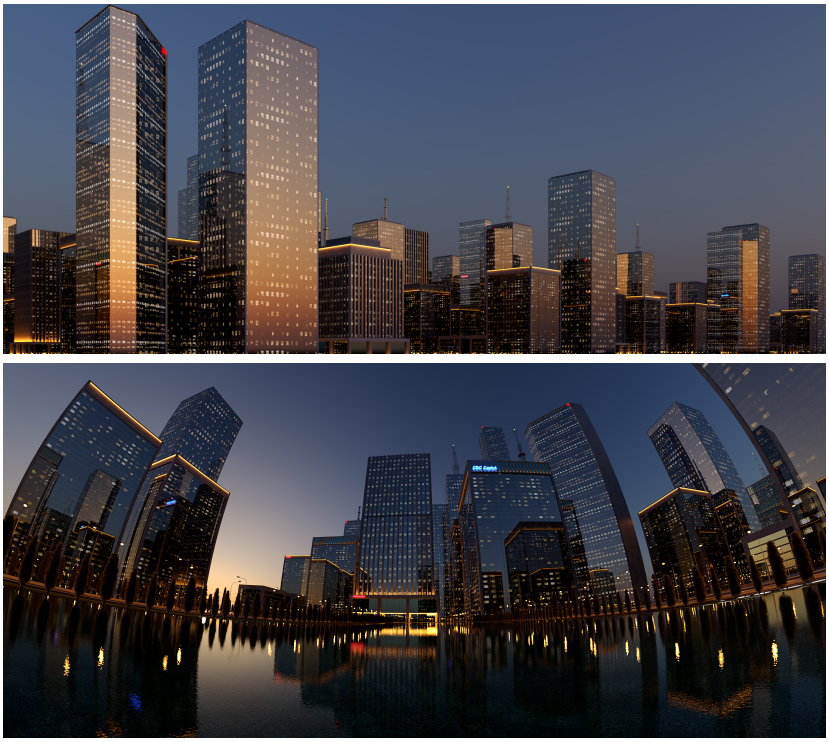


Figure 3.33: Two examples of post-sunset conditions rendered using the Prague Sky Model in the Corona renderer.

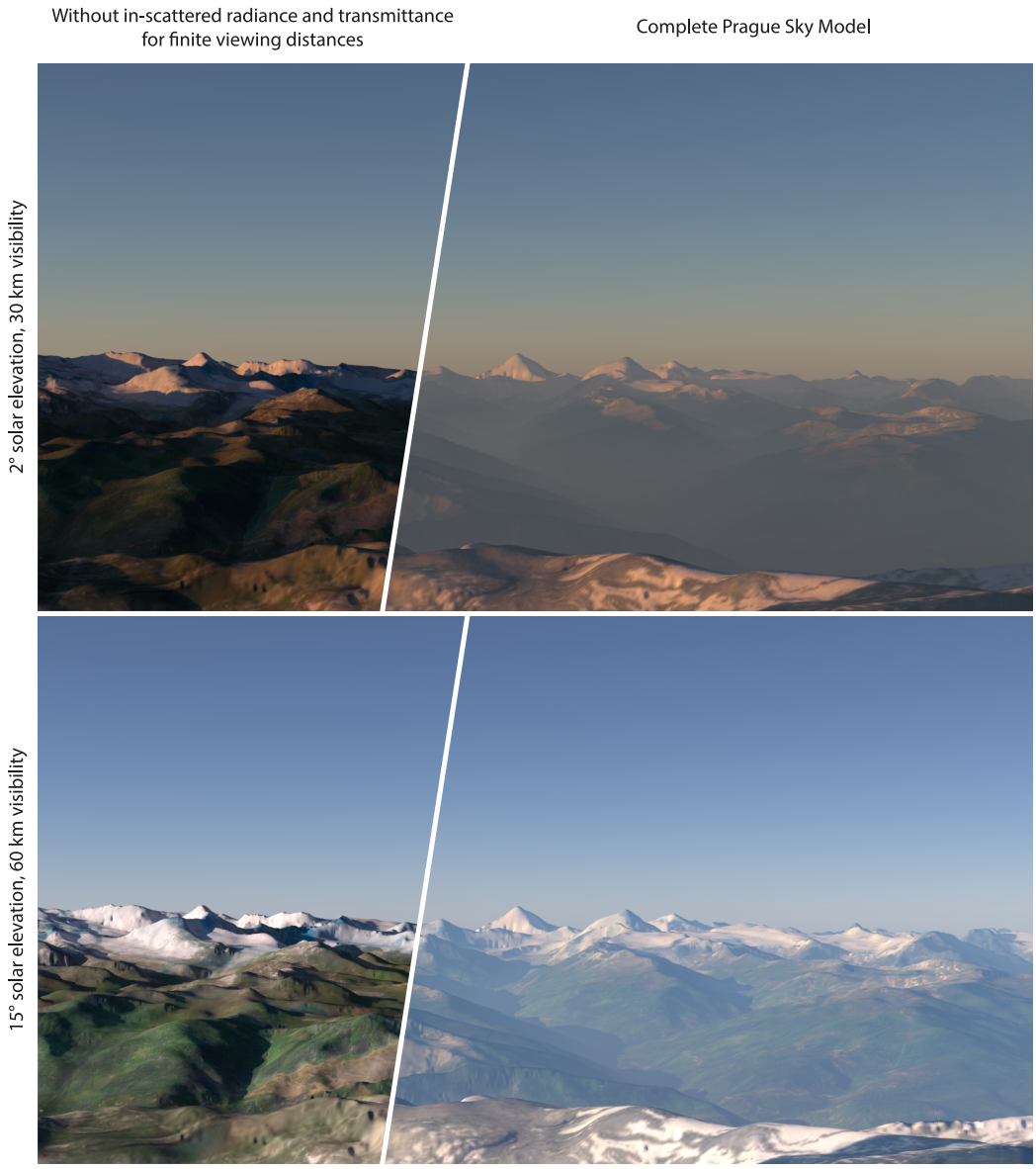


Figure 3.34: The two mountain landscape images from Figure 3.1 with in-scattered radiance and transmittance for finite viewing distances removed to show their importance. Rendered using the Prague Sky Model in the Corona renderer.

the post-sunset case being the most difficult one. By comparison, the renders that used the fitted model used 100 spp and 2 core hours each (= 5 minutes on 24 core CPU).

Polarisation

To show that sky polarisation has a visible impact on renderings and its inclusion in the Prague Sky Model was therefore beneficial, 2 examples are provided. First, Figure 3.35 shows how polarisation affects reflection of the sky in building facades and inter-reflections. Since ART has limited modelling capabilities and cannot use mainstream scene description formats, and also for an equal comparison with the work of Wilkie et al. [2004], the original scene file for one of their test scenes was obtained from the authors for this figure.

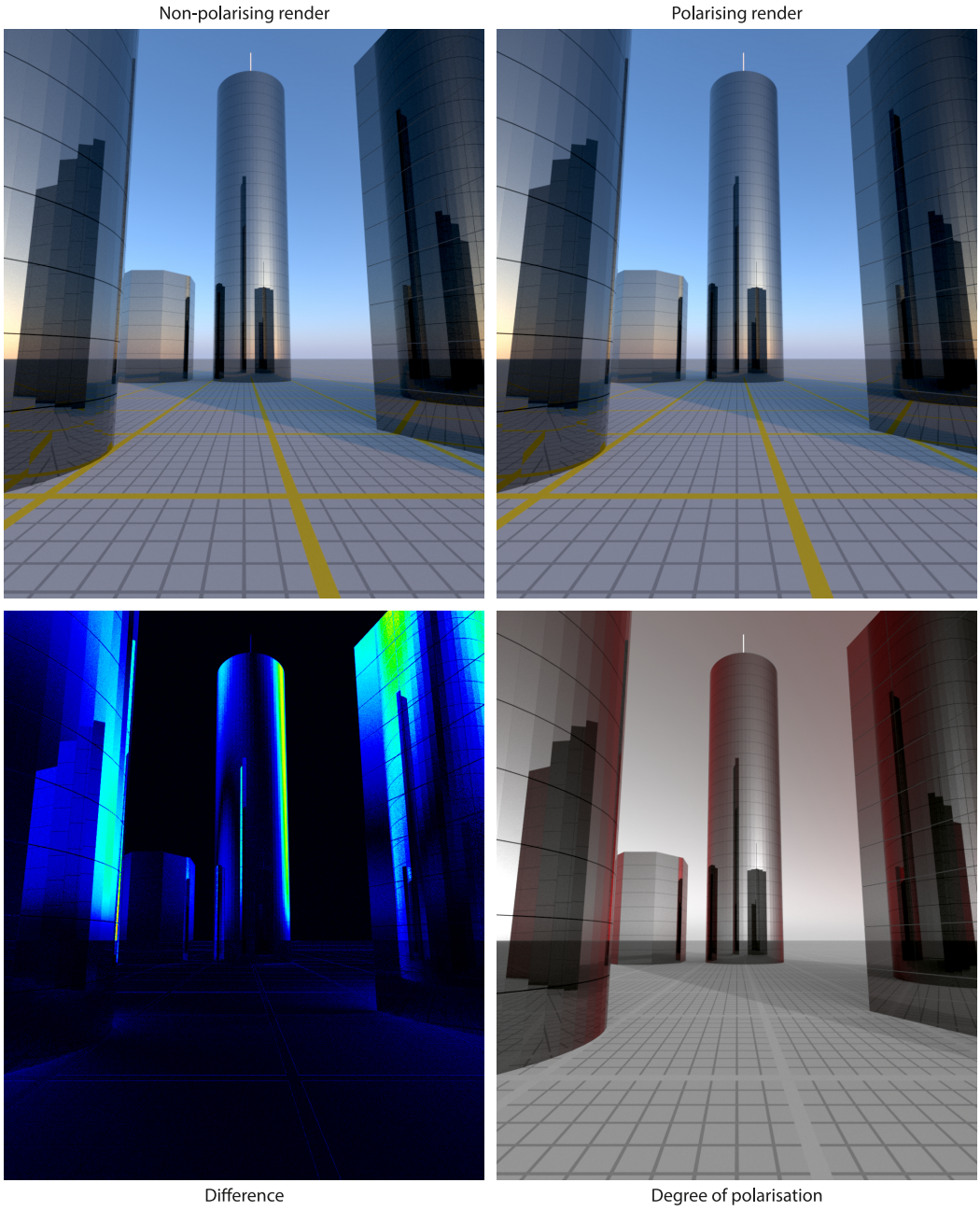


Figure 3.35: The specular architecture scene from Wilkie et al. [2004], re-created using ART [Wilkie, 2018]. Qualitatively similar behaviour as seen in the 2004 publication can be observed. Image rendered with a polarising (top left) and a non-polarising version of the Prague Sky Model (top right). The difference image (bottom left) shows that skylight polarisation not only affects the reflection of the sky in the building facades, but also the inter-reflections of the buildings. The image on the bottom right shows the degree of polarisation as scaled overlay for 550 nm, provided by `polvis` (see Section 3.6.1).

Second, the combination of polarisation with finite distance in-scattered radiance and transmittance allows the Prague Sky Model to simulate removing haze using a polarisation filter. With a properly oriented linear polarisation filter a part of the in-scattered light in a daytime outdoor scene can be removed as shown in Figure 3.36.

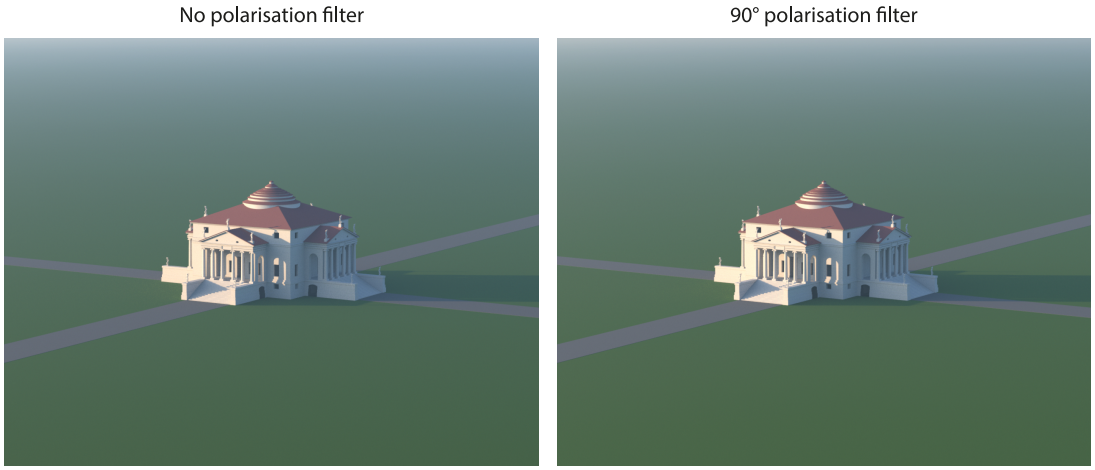


Figure 3.36: Aerial perspective removal with a linear polarisation filter. **Left:** A distant view of a building, seen through a 50% neutral filter. **Right:** The same scene seen through a 90° filter. Even though the haze removal effect obtainable by a polariser is rather subtle, there is a visible slight blue tint in the left image that is not present in the right one.

3.9.3 Comparison to other sky models

As the use of exponential scatterer distributions is a common feature of many other models, the first comparison of the Prague Sky Model is made against an exponential. In Figure 3.37, renderings using the modelled atmosphere based on OPAC profiles (defined in Section 3.5.2) are placed next to three examples of exponential scatterer distributions. The first exponential profile (second column) is constructed to match the OPAC-based profile below the inversion layer. As such it results in much hazier atmosphere. On the other hand, the second profile (third column) matching the model above the inversion layer makes the atmosphere much clearer as the inversion layer is completely missing there. Finally, the third exponential profile (fourth column) is designed to give the same ground level visibility and vertical turbidity as the modelled atmosphere. These results show similarities at lower altitudes, however the exponentials diverge as altitude increases due to the realistic non-exponential scatterer distribution used by the Prague Sky Model.

To verify this observation a direct comparison to two existing sky models that are based on exponential scatterer distributions is provided: the model by Hillaire [2020] and the one by Bruneton [2016]. For this, the source code provided by Hillaire was used which implements both his and Bruneton's model. Its input parameters were set so as the used extinction coefficients were the same as in the Prague Sky Model, and the used exponential aerosol profile resulted in the same ground level visibility and vertical turbidity. As expected, both models then provide output very close to renders of the similarly constructed exponential profile presented in Figure 3.37 (the rightmost column). Therefore, in Figure 3.38 a good match between all three models at 2 m can be observed. But with increasing observer altitude, differences between the OPAC-based atmosphere and the two exponential ones start to manifest themselves.

For completeness, Figure 3.4 shows a sample comparison against the model by

Hošek and Wilkie [2012] although comparing models with so different feature sets is difficult. Even with in-scattered radiance and transmittance for finite viewing distances switched off, the result provided by Prague Sky Model is still more realistic.

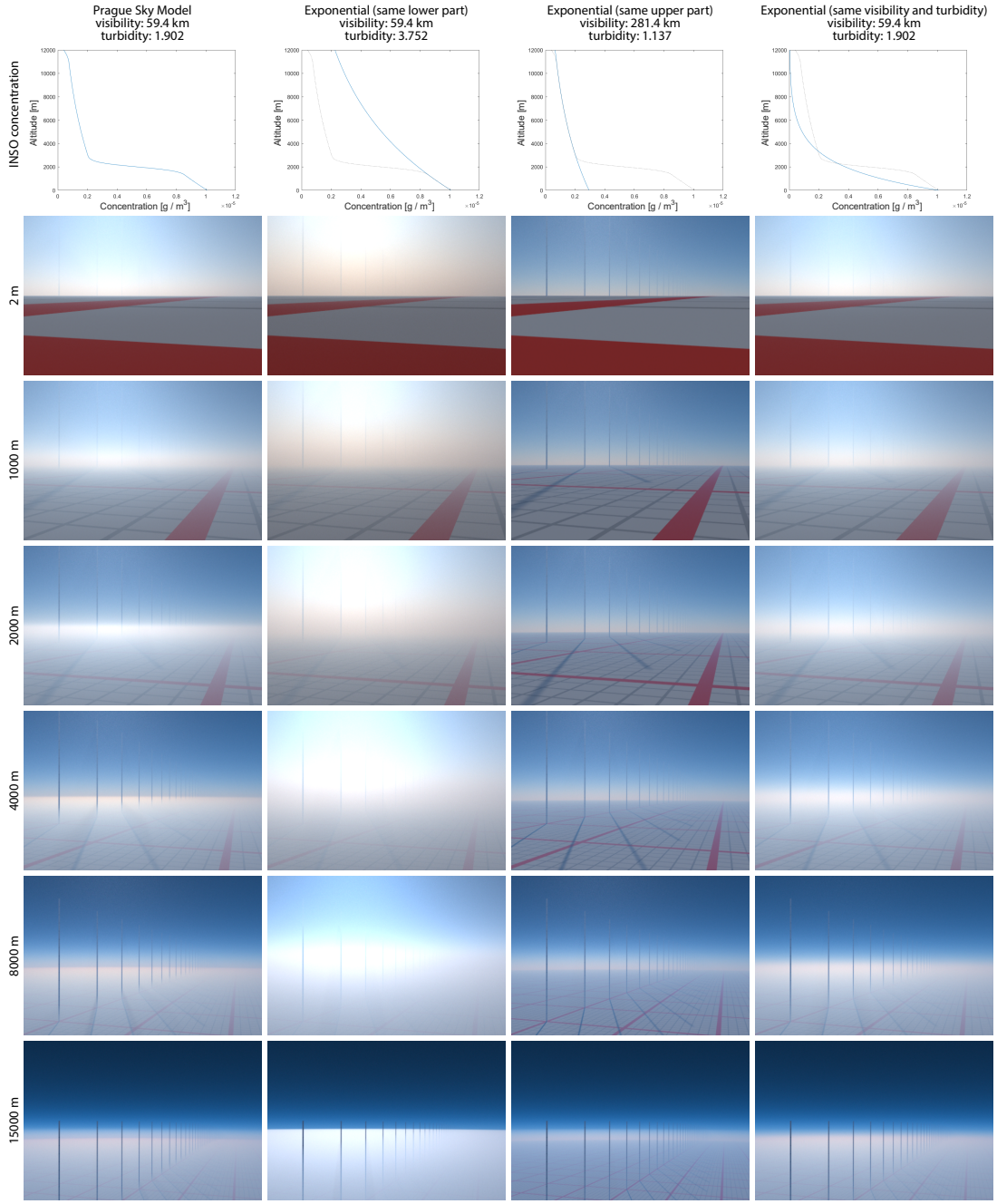


Figure 3.37: A comparison of the OPAC-based atmosphere scatterer profile of the Prague Sky Model versus three examples of purely exponential ones in the Columns scene for solar elevation 16°. The key difference is that no clear sense of observer altitude can manifest itself with an exponential fall-off: in a real clear atmosphere, there is a distinct hazier layer in the first 1–3 km from the ground, with significantly clearer air above it. A purely exponential model is therefore unable to provide realistic views from mountaintops or aircraft, where this feature plays an important role in overall scene appearance.

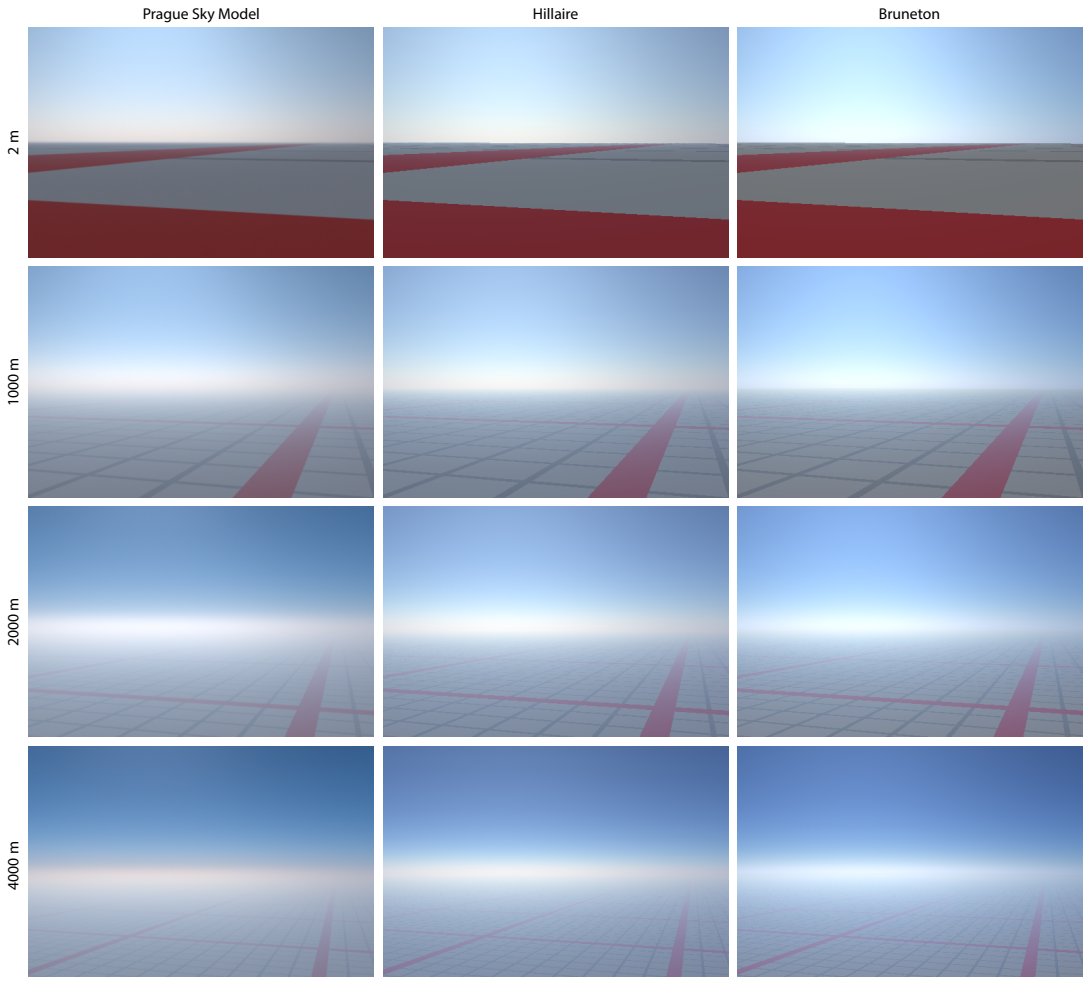


Figure 3.38: A comparison of the Prague Sky Model with two other related works by Hillaire [2020] and Bruneton [2016] in a modified Columns scene (since the used Hillaire's implementation didn't allow easy creation of the columns geometry). Observer altitudes 2, 1000, 2000 and 4000 m (from top to bottom), solar elevation 16°, visibility 59.4 km.

3.9.4 The SWIR extension

Our extended model provides sky radiance, transmittance, and polarisation for zero observer altitude and 55 regularly spaced wavelength channels from 280 nm to 2480 nm. The remaining model parameters, i.e. ground albedo, solar elevation, and visibility, are the same as in the Prague Sky Model. A sample of this output is shown in Figure 3.39. For example, the first channel (280 – 320 nm) shows that although the transmittance is low due to a strong ozone absorption, the sky radiance is relatively high compared to the other channels because of the high extraterrestrial solar radiance there. On the other hand, low solar radiance makes the sky radiance in the last channel (2440 - 2480 nm) very low even though the transmittance is high. And when the decreasing solar radiance meets with the low transmittance due to the water absorption in channel 1360 – 1400 nm, the resulting sky radiance is almost zero. Another set of channels and model parameters is shown in Figure 3.2.

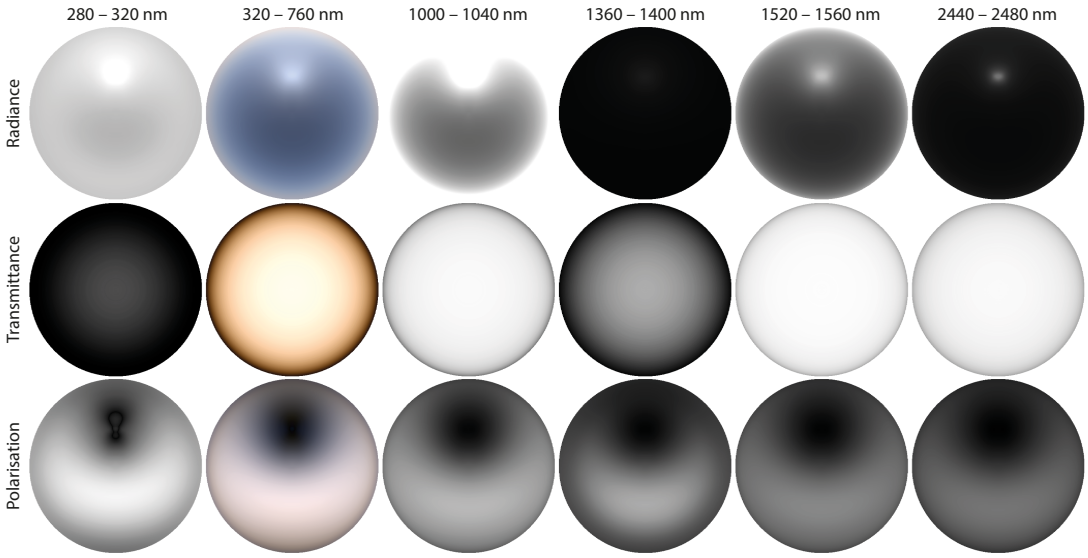


Figure 3.39: An example of outputs provided by our model: sky radiance, transmittance and polarisation for six wavelength bins. All images show up-facing fish-eye view of the sky with the same configuration of ground albedo 0.5, solar elevation 45° and visibility 59.4 km. The second column was obtained by combining corresponding channels into an sRGB image. Each row was tone-mapped independently, but all images in one row share the same exposure, except for the second column.

Fitting error

Similarly to the Prague Sky Model, we also analysed the amount of error introduced by the fitting. Here we discuss errors with respect to the wavelength channels since the behaviour with respect to ground albedo, solar elevation and visibility stays the same as in the Prague Sky Model. Figure 3.40 shows a box plot of the normalised mean absolute errors between the fits and reference images grouped by the wavelength channels. For most channels, the median of the errors stays under 10%, the total average error is 6%. However, there are three cases where the error is significantly larger: in the near-UV channel 280 – 320 nm, between 1360 and 1400 nm, and between 1840 and 1960 nm. All three cases correspond to strong absorption bands caused by ozone and water vapour, which make atmosphere simulation by the path tracer more difficult. More traced paths end up being absorbed, and by the time other channels have already converged, the noise level in these three bands remains high. As a result, the error between the smooth fit and noisy reference image increases. However, as the grey overlay in the figure shows, these regions carry very low energy, making them essentially irrelevant. Similarly, all channels above 2000 nm exhibit larger errors then the channels in the visible range but the unnormalised mean absolute error is practically zero there. Figure 3.41 gives an example of a configuration with the average error.

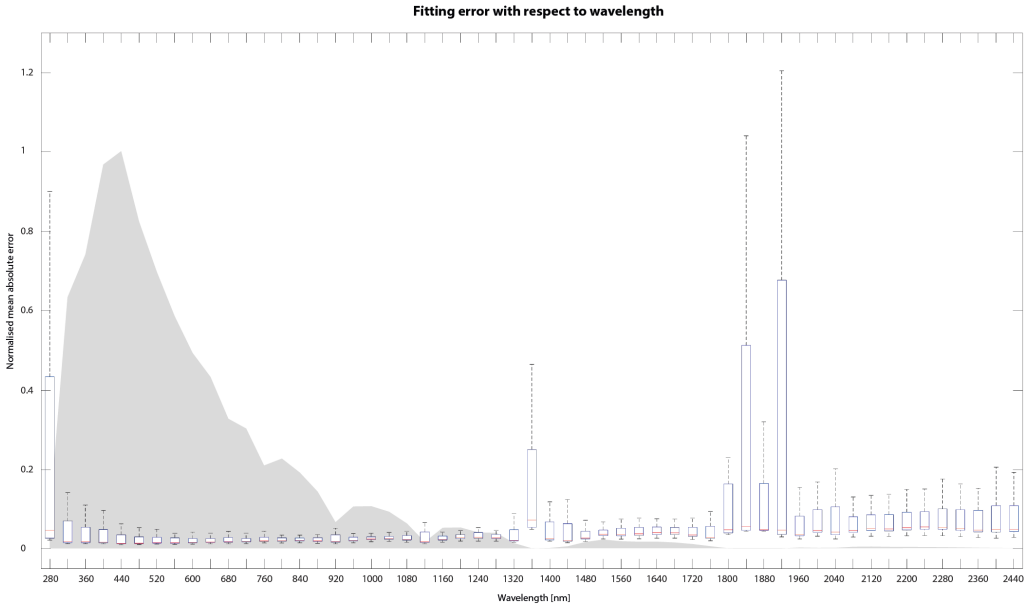


Figure 3.40: A box plot of the mean absolute error between fits and their corresponding reference images, normalised by the average value of each channel, shown for all channels in our model. The red line is the median, the blue box goes from the first to the third quartile, and the whiskers are the minimum and maximum values. The increased error around 280, 1360 and 1840 nm is due to noise in the reference images caused by strong absorption in these bands. The grey shape in the background shows the normalised average pixel value in each channel.

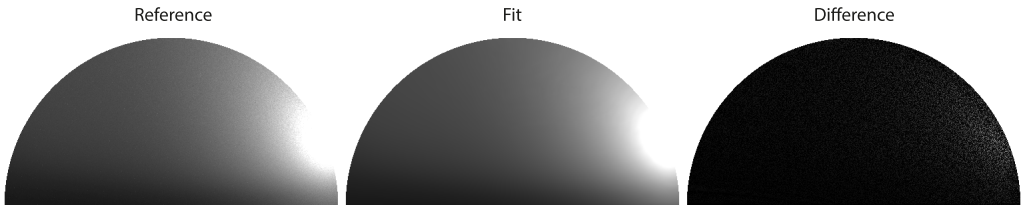


Figure 3.41: An example of configuration with the average fitting error. It shows side-facing fish-eye views corresponding to ground albedo 0, solar elevation 18.64°, visibility 59.4 km and wavelength channel 1320 – 1360 nm, the normalised mean absolute error is 0.059. The error is barely noticeable in the difference image and completely invisible in the fit. The lower halves are black and were omitted.

3.10 Limitations and future work

There are several areas in which the Prague Sky Model could be improved in the future:

1. Most importantly, the model can easily be made to use genuine scattering profile data for high haziness situations instead of extrapolating from OPAC – if such data is found in atmospheric science literature. A key issue here seems to be purely meteorological, in the sense that there is potentially a large number of different fog and haze configurations that can cause low ground level visibility: and merely having a single "view distance" parameter is not enough to differentiate between these.

2. Special provisions can be made to accommodate the sharp circumsolar features that appear if "real" INSO lobes were to be used, instead of the smoothed OPAC ones.

3. Feature sharpness in the horizon area could be increased by fine-tuning the radiance fitting step.

4. More subtle polarisation patterns could be represented if full Mie scattering was used in the brute force simulator: however, this would substantially increase the size of the polarisation fitting, as the optimisation discussed in Section 3.7.4 could no longer be used.

5. Finally, similarly to the SWIR extension of the spectral range, adding even higher observer altitudes or even lower solar elevations might be useful for some applications. Such extensions should be more straightforward than in the case of the spectral range, although it would still increase the dataset size and in case of low solar elevations also slowed down the pre-computation step.

Besides improvements, interesting topics for future research can be found also in novel applications of the Prague Sky Model. For example using the model not only for illuminating a scene but also for importance sampling of the sky.

The SWIR extension

In terms of practical usability of our extended model, the biggest limitation is the size of the final dataset. The hemispherical version presented in this chapter, with its extended set of wavelength channels, requires 550 MB of disk space and twice as much computer memory when loaded for use. Using the same compression technology, a full altitude-resolved model with the same extended channels would consume 12.1 GB of disk space and 24.2 GB of computer memory. However, it has to be noted that use cases for altitude-resolved SWIR sky models seem to be scarce. We proposed a method for omitting unnecessary channels, which can reduce the dataset size if some additional error can be tolerated in a given application. But for any additional extensions of the model a more efficient compression method will have to be found.

Another limitation for further development of the model is the resource-intensive reference dataset generation. This means rendering a large number of images to a low level of noise, which requires a lot of time and computational resources. Recall, that rendering of our hemispherical extended dataset consumed more than 800 thousand core-hours and was done in the course of one week on a scientific supercomputing cluster. The rendering step of the pre-computation runs significantly slower than in the Prague Sky Model because of the additional wavelengths. There is strong absorption in some of the new wavelength bands, which makes rendering in these bands considerably more noisy. However, the slowdown affects the pre-computation step only and rendering a single wavelength channel using our model takes the same amount of time as with the Prague Sky Model. Nevertheless, improving rendering in the difficult channels by, e.g., improving the next event estimation [Hanika et al., 2022] or utilizing some kind of path guiding [Herholz and Dittebrandt, 2022] would be beneficial.

3.11 Conclusion

In this chapter, two fitted sky models were described: the Prague Sky Model and its SWIR extension. The Prague Sky Model is a comprehensive and realistic pre-computed sky dome model for rendering outdoor environments under clear skies. It improves the state of the art in several ways:

1. The model is based on OPAC standard atmospheric constituent data, so the molecular and aerosol distributions represent realistic viewing conditions.

2. The novel in-scattered radiance model provides a full spherical fitting, and is available for observer altitudes up to 15 km altitude. This allows viewpoints above ground level, and downward looking renders. The new in-scattering model can also be evaluated for finite viewing distances.

3. The model extends to solar elevations past sunset, and provides post-sunset features such as proper twilight blue (due to ozone absorption in the high atmosphere), and the shadow of the Earth rising opposite the setting sun.

4. The model provides a matching transmittance function that is view dependent, and also changes with observer altitude.

5. Finally, the model includes also a fitted function for the linear polarisation patterns found in clear skies.

As these components are all derived from the same ground truth dataset, they are consistent, and can be seamlessly used together. To our knowledge, features 1 to 3 are true novelties, and are not available in any existing pre-computed models of sky dome radiance. Feature 4 is only available for the model by Preetham et al. [1999], but not in any later models. Feature 5 was attempted by Wilkie et al. [2004] and Wang et al. [2016], but the Prague Sky Model is the first to provide polarisation information which matches all other components.

While the author was not the primary investigator of the Prague Sky Model, he collaborated on its development and contributed to basically all of its parts: atmosphere composition specification, reference dataset rendering, fitting, and implementation. The author then continued as the primary investigator in further development of the model, in particular he introduced its SWIR extension.

The SWIR extension provides sky radiance, transmittance and polarisation for a much wider spectral range than such models offered before. It covers practically the entire range of solar irradiance at ground level, which makes the extended model useful for predictive simulations of photovoltaic plant yield and thermal building assessments. The author also created a new implementation common to both the Prague Sky Model and SWIR extension making the models more accessible.

While the previous two chapters focused on improving the performance of MC rendering by decreasing the variance of the MC estimators, the Prague Sky Model and SWIR extension present a rather different approach. By pre-computing difficult parts of light transport in the sky, they exclude high computational cost of these parts from rendering completely. Moreover, the pre-computation is done only once and using high quality brute force renderings, therefore any renderer can now achieve realistic sky dome appearance without any atmospheric simulation overhead.

3.12 Appendix

3.12.1 Atmospheric data plots

This section provides plots of all data describing the atmosphere modelled by the Prague Sky Model and its SWIR extension. See Section 3.5 for discussion of their source and usage.

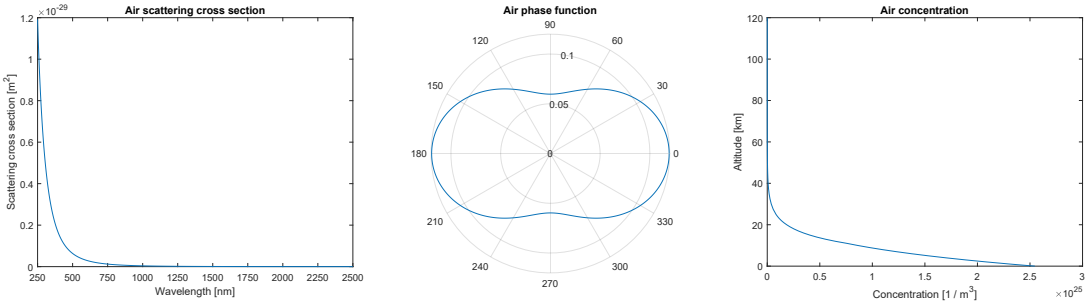


Figure 3.42: Scattering cross section, Rayleigh phase function and vertical particle concentration profile of air ($N_2 + O_2$).

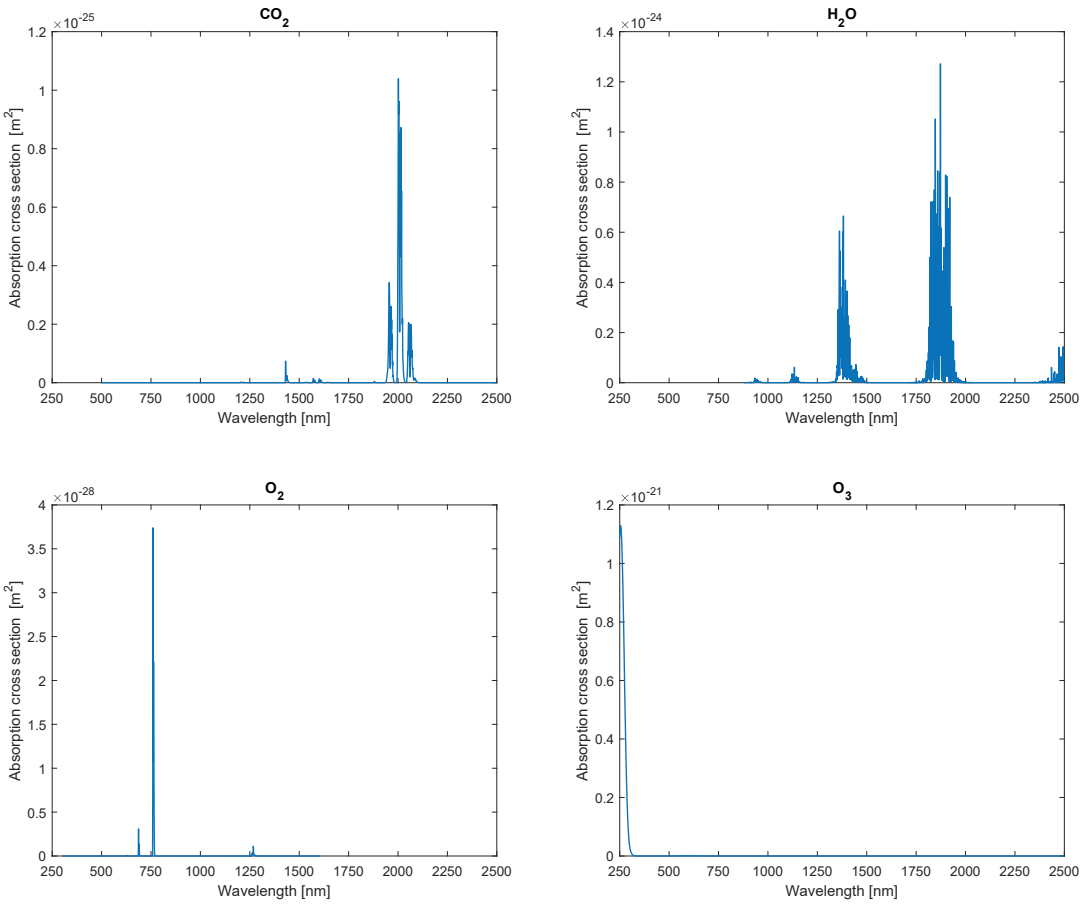


Figure 3.43: Absorption cross sections of CO_2, H_2O, O_2, and O_3.

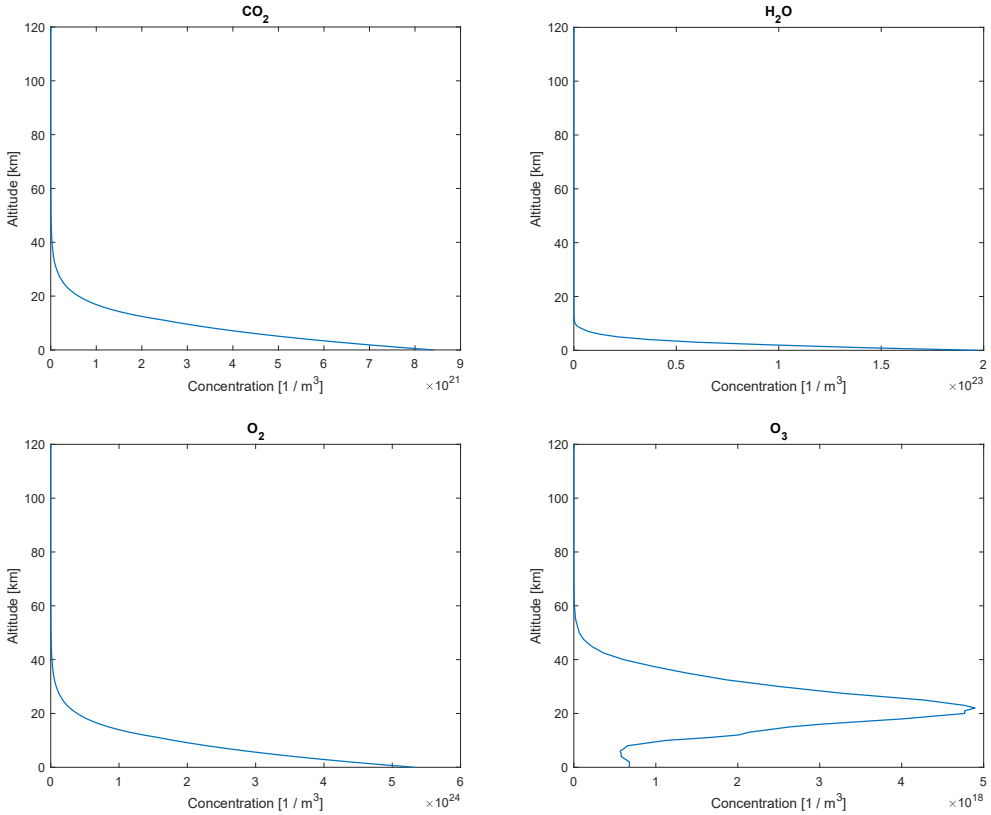


Figure 3.44: Vertical particle concentration profiles of CO_2, H_2O, O_2, and O_3.

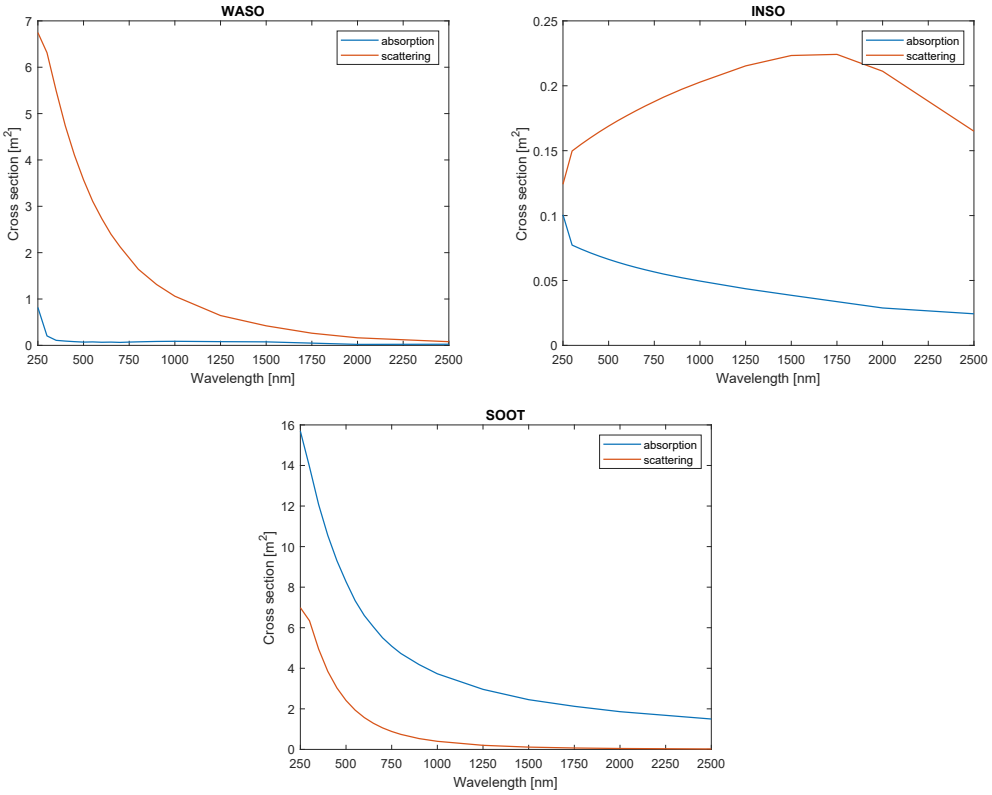


Figure 3.45: Absorption and scattering cross sections of the WASO, INSO and SOOT aerosols.

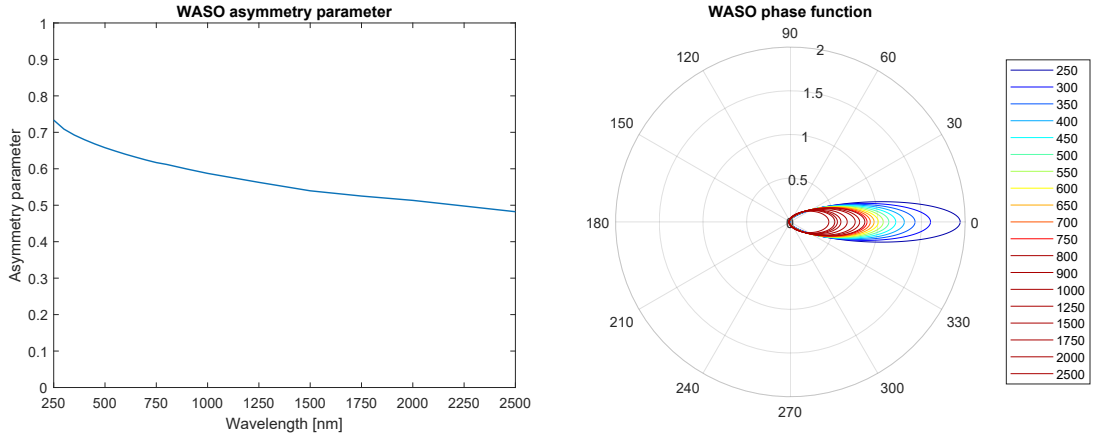


Figure 3.46: The fitted asymmetry parameter and the corresponding Henyey-Greenstein phase function of the WASO aerosols.

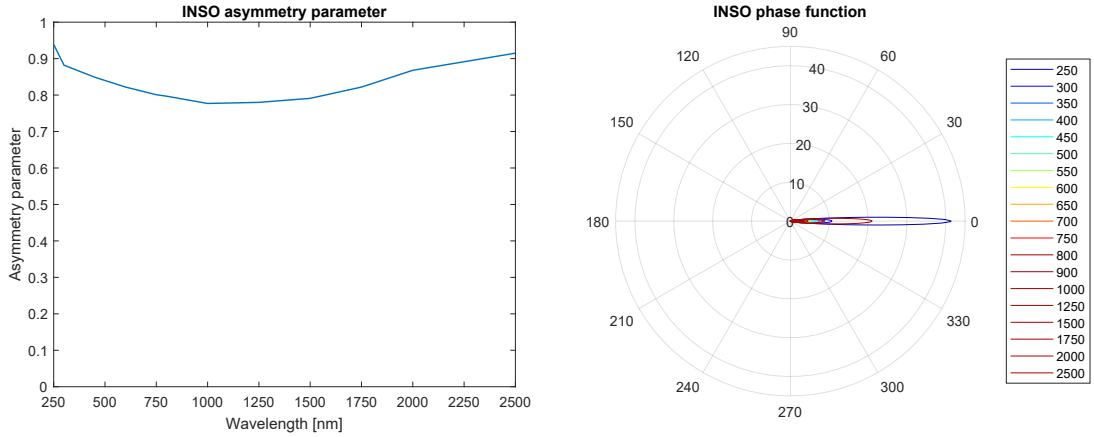


Figure 3.47: The fitted asymmetry parameter and the corresponding Henyey-Greenstein phase function of the INSO aerosols.

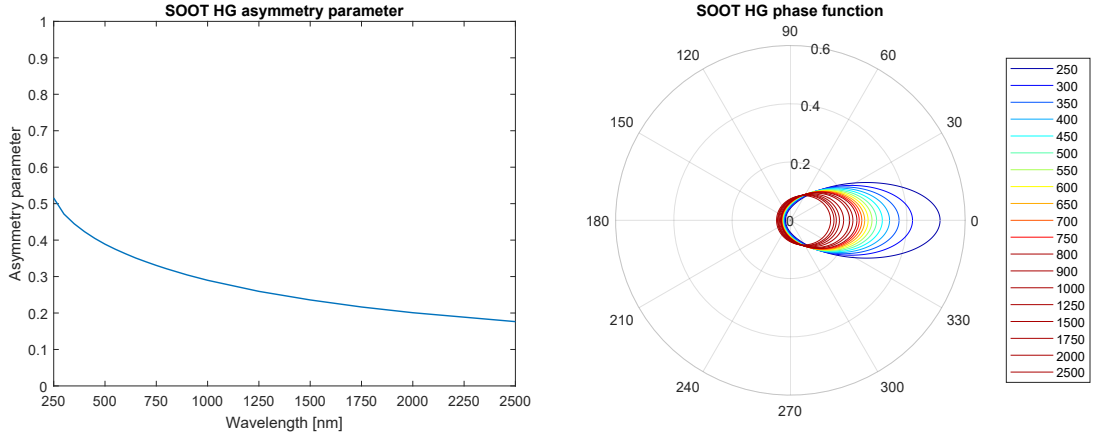


Figure 3.48: The fitted asymmetry parameter and the corresponding Henyey-Greenstein phase function of the SOOT aerosols used in the Prague Sky Model.

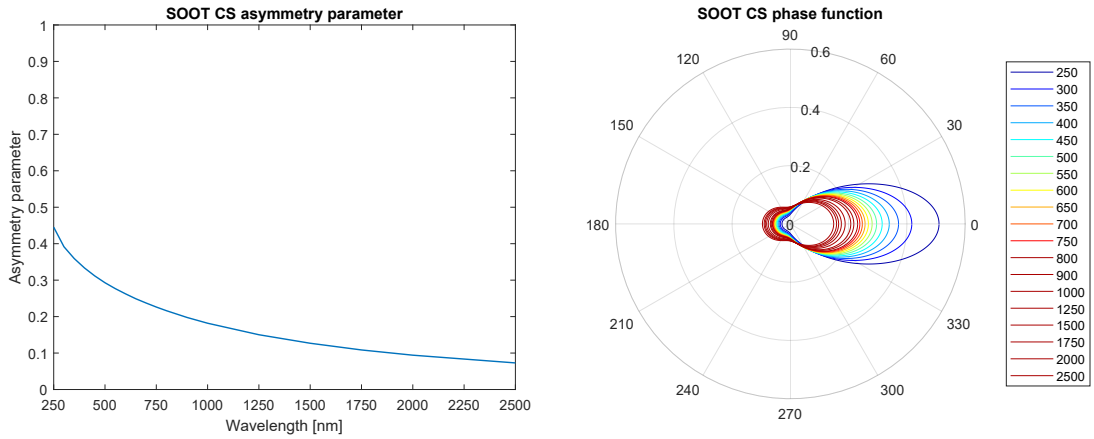


Figure 3.49: The fitted asymmetry parameter and the corresponding Cornette-Shanks phase function of the SOOT aerosols used in the SWIR extension.

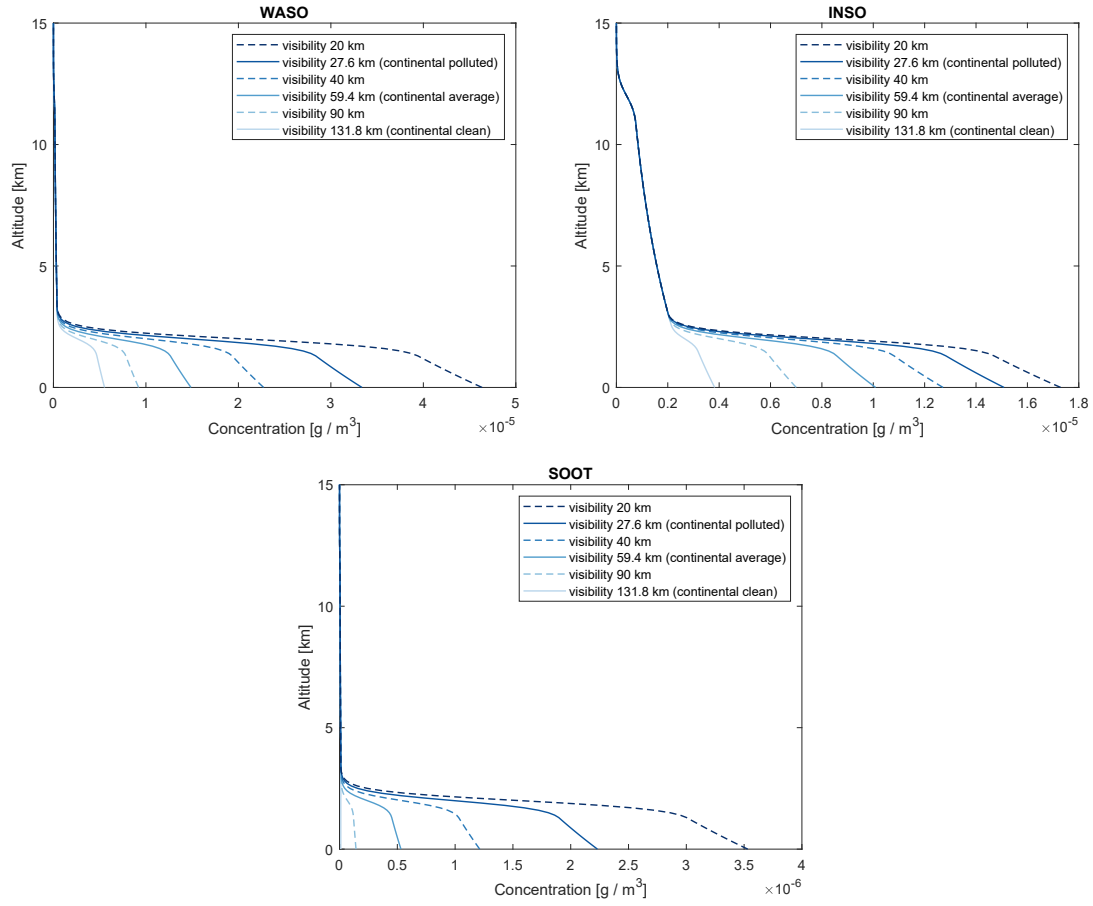


Figure 3.50: Vertical particle concentration profiles of the WASO, INSO and SOOT aerosols for different visibilities.

3.12.2 Validation

In order to verify that the atmosphere composition used by the Prague Sky Model and its SWIR extension is physically plausible and that the used brute force renderer `atmo_sim` works correctly, its output was validated against three different sources: data measured by Kider et al. [2014], images rendered in `libradtran` [Emde et al., 2016], and Alexander Wilkie's own empirical observations.

Validation against Kider

Kider et al. [2014] provide a systematically collected dataset which includes spectral sky dome radiance measurements taken in the course of several days for 81 sample positions in the sky. Figure 3.51 compares this measurements for one particular date and time (27. 05. 2013, 09:30) with results simulated by `atmo_sim`. Since the measurements do not contain exact atmospheric parameters at the time of capture, input parameters for `atmo_sim` were obtained experimentally: ground albedo 0, observer altitude 0 m, solar elevation 41.08° and visibility 59.4 km (which corresponds to OPAC continental average aerosol composition) provided a very good match to the measurements. The figure also proves importance of the absorbers added by the SWIR extension (described in Section 3.5.1).

Figure 3.52 shows a different comparison of the same data. Instead of comparing the measurements with the simulation with respect to wavelengths, a comparison in the image space is shown. The slightly higher error in the immediate circumsolar region is caused by using the INSO asymmetry parameter specified by OPAC instead of a precise fit, as discussed in Section 3.5.2. Nevertheless, the simulation matches the measurements closely (compare with similar error plots in the work of Kider et al. [2014]) which proves it to be highly physically plausible. Note that the error of the SWIR extension is necessarily higher since errors from more wavelengths are included.

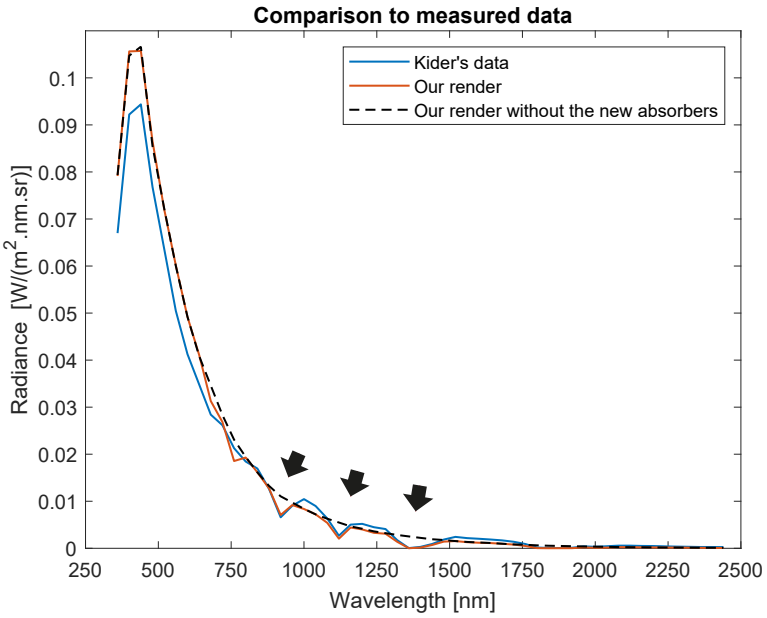


Figure 3.51: A comparison of the modelled atmosphere to real sky dome measurements by Kider et al. [2014]. **Blue:** Radiance spectrum from the Kider dataset for 27. 05. 2013, 09:30, averaged over all provided sample points. **Red:** Radiance spectrum rendered by `atmo_sim` for a similar atmospheric configuration, averaged over the same sample points. **Black dashed:** Radiance spectrum rendered by `atmo_sim` but for an atmosphere without the CO_2, H_2O, O_2 absorbers added by the SWIR extension. Note how the additional absorbers yield much better match to the real data at wavelengths marked by the black arrows.

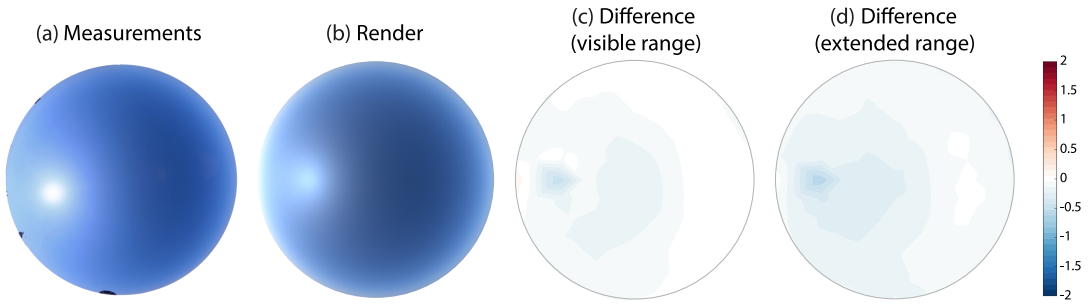


Figure 3.52: A comparison of the modelled atmosphere to real sky dome measurements by Kider et al. [2014]. (a) A tone-mapped up-facing fish-eye photo from the Kider HDR dataset for 27. 05. 2013, 09:30. (b) A tone-mapped render obtained with `atmo_sim` for ground albedo 0, observer altitude 0 m, solar elevation 41.08° and visibility 59.4 km. (c) Relative error plot between the spectral data corresponding to (a) and (b), limited to the spectral range of the Prague Sky Model. (d) Same as (c) but includes the entire spectral range of the SWIR extension. Note that the plots use the same colour scheme and scale as Kider et al. and are therefore comparable with similar plots in their work.

Validation against `libradtran`

This section provides a validation of the results computed by `atmo_sim` against those obtained by `libradtran`, a scientific software package for radiative transfer calculations within an atmosphere [Emde et al., 2016]. For the comparisons, `libradtran`'s Monte Carlo radiative transfer solver MYSTIC [Mayer, 2009] is used, which traces photons at given wavelengths through the atmosphere. MYSTIC is run individually for every single pixel at selected wavelengths 420 nm, 540 nm and 620 nm, which roughly corresponds to blue, green and red, respectively. To get comparable outputs from `atmo_sim`, spectral images with 10 nm wide wavelength bands are rendered and corresponding single-wavelength data are extracted using the `tonemap` tool from the ART toolchain. The comparison images show radiances captured by a panoramic 180° × 180° camera with 90° towards the zenith (top half) and 90° towards the ground (bottom half).

Atmosphere without aerosols The first validation was done for a simple atmosphere without any aerosols (i.e., containing only molecules described in Section 3.5.1), the result is shown in Figure 3.53. The validation was performed with a diffuse ground albedo 0.2 for two different observer altitudes (0 km and 10 km) and solar elevations (5° and 45°). The chosen altitudes correspond to viewing the sky from the ground (0 km) and from a commercial airplane (with typical flight altitudes around 10 km). The elevation of 5° was chosen to validate the O_3 absorption, as it is most noticeable at low sun elevations. The difference images are computed by dividing the radiances simulated by `atmo_sim` by the radiances from MYSTIC, which shows which areas are brighter and darker, respectively.

The average difference at solar elevation 45° is 1.0025 with an average signal correlation of 0.99978. The error is uniformly distributed over the images with no apparent patterns.

At 5°, the average difference is 1.0002 but with a higher variance than at 45°. The average signal correlation is 0.99960. Notice that `atmo_sim` is darker at higher wavelengths, which is most likely due to a slightly different O_3 absorption curve, as this does not happen when O_3 is removed from the atmosphere.

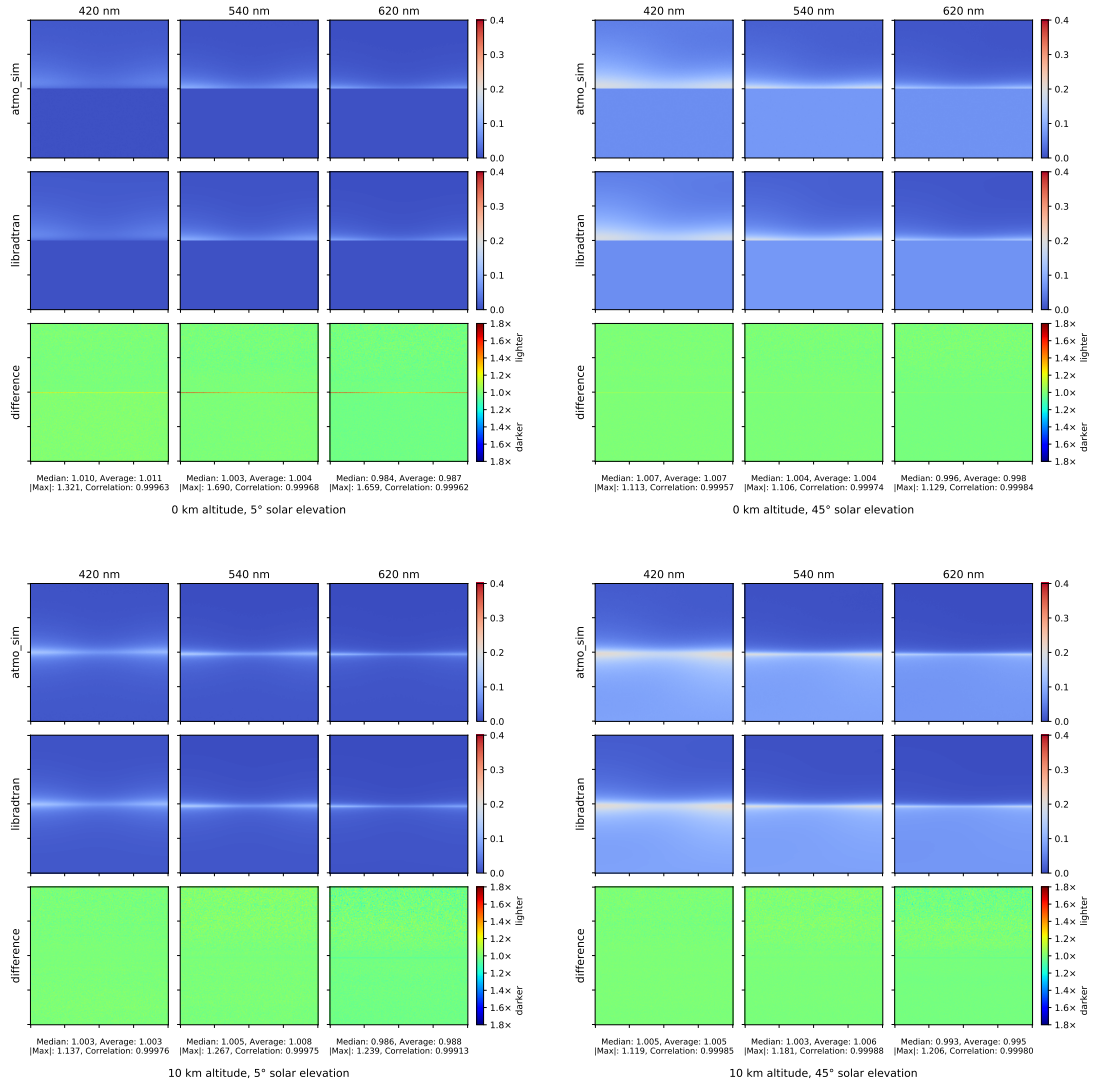


Figure 3.53: Atmospheres without any aerosols at 0 km and 10 km altitudes (top, bottom) with solar elevations 5° and 45° (left, right). Notice that the radiances simulated by `atmo_sim` are comparable to `libradtran` without any noticeable error patterns. With increasing wavelengths, the modelled atmosphere produces slightly darker images, which only happens with an O_3 layer and is most likely due to a different absorption cross section. The error lines at horizons are discussed in the text. Notation and scale: The top two rows of every comparison show radiances with a colour scale in $\mathrm{W} \cdot \mathrm{sr}^{-1} \cdot \mathrm{m}^{-2}$ per wavelengths 420 nm, 540 nm and 620 nm separately in the three columns. The difference images are acquired by dividing `atmo_sim` radiances by `libradtran` radiances per pixel. The colour scale is normalised to show brighter and darker areas with an equivalent weight. The median, average and maximum absolute errors are computed directly from the per-pixel divided radiances. The signal correlation coefficients are computed from the radiances flattened to a 1D array.

Finally, notice that in some comparisons there is a 1 pixel horizontal stripe of large differences between `atmo_sim` and MYSTIC. The stripe is always located at the planet edge, i.e., at the horizon for altitudes of 0 km, or slightly below the horizon for 10 km. These artefacts are most likely caused by sub-pixel sampling and jittering in `atmo_sim`, when for a pixel on the planet edge, some of the photon paths hit the planet and some do not. On the other hand, in the script that evaluates MYSTIC, the radiance is always simulated in the middle of a pixel, hence it is always either above or below the planet edge with no jittering and randomness.

Atmosphere with aerosols The second validation was done for a complete atmosphere including the aerosols, Figure 3.54 shows the result. As discussed in Section 3.5.2, `atmo_sim` uses analytical Henyey-Greenstein and Cornette-Shanks phase functions instead of precisely sampled Mie phase functions provided by `libradtran`. The fitted asymmetry parameter g for the WASO and SOOT leads to very close matches to the sampled Mie lobes, while a more "blurry" asymmetry parameter provided by OPAC is used for the INSO particles. The effect of this simplification is that the circumsolar region has a considerably less peaky distribution of energy right next to the solar disc. This, in turn, makes the resulting function easier to fit, and `atmo_sim` rendering converges faster. However, it is worth noting that the more blurry asymmetry parameter is not per se *unrealistic* – it just deviates from what the U.S. Standard Atmosphere datasets should contain, in that the more blurry parameter corresponds to different particles being present, instead of the actual INSO ones. The remainder of the atmosphere remains exactly as specified. Validations against MYSTIC were again performed in the same way as in the previous case with no aerosols.

As expected, due to the INSO phase function simplification discussed in the previous paragraph, the largest differences can be seen in the immediate circumsolar region, which is especially noticeable at 0 km and 45°. The inner parts of the solar glow are darker in `atmo_sim`, and the outer parts are lighter. The real INSO particles are very strongly forward-scattering, but it was better to avoid using such an extreme phase function in `atmo_sim`. So the more blurry estimate provided by OPAC – possibly because they also had, at some point, a reason to avoid the very narrow real INSO lobes – came in very handy for this purpose.

To verify that all the observable differences are indeed due to the INSO scattering implementation in `atmo_sim`, another experiment was run with only INSO aerosols present, where `libradtran` was forced to use the same Henyey-Greenstein phase function with the same asymmetry parameter as in `atmo_sim`. As can be seen in Figure 3.55, this completely eliminates all the noticeable differences and results in a perfect match across all the wavelengths. Hence, the noticeable differences between `atmo_sim` and `libradtran` are caused by using aerosol phase functions that are only approximated by the Henyey-Greenstein formula, and by using the blurry asymmetry parameter for the INSO particles.

Conclusion To summarize, it can be seen that the radiances simulated by `atmo_sim` which was used for generating the reference datasets are good matches to results obtained with a well-established research-grade atmospheric library `libradtran` and its Monte Carlo radiative transfer solver MYSTIC. Atmospheres

without any aerosols yield almost exactly the same results, which means that Rayleigh scattering and absorption are simulated correctly. The addition of aerosols yields differences caused by the analytical Henyey-Greenstein phase function approximation, which is especially noticeable around the sun as its light spreads more. However, even then, the outer solar glow is only less than 2 times as bright than in `libradtran`, which can be considered to be a perfectly valid approximation.

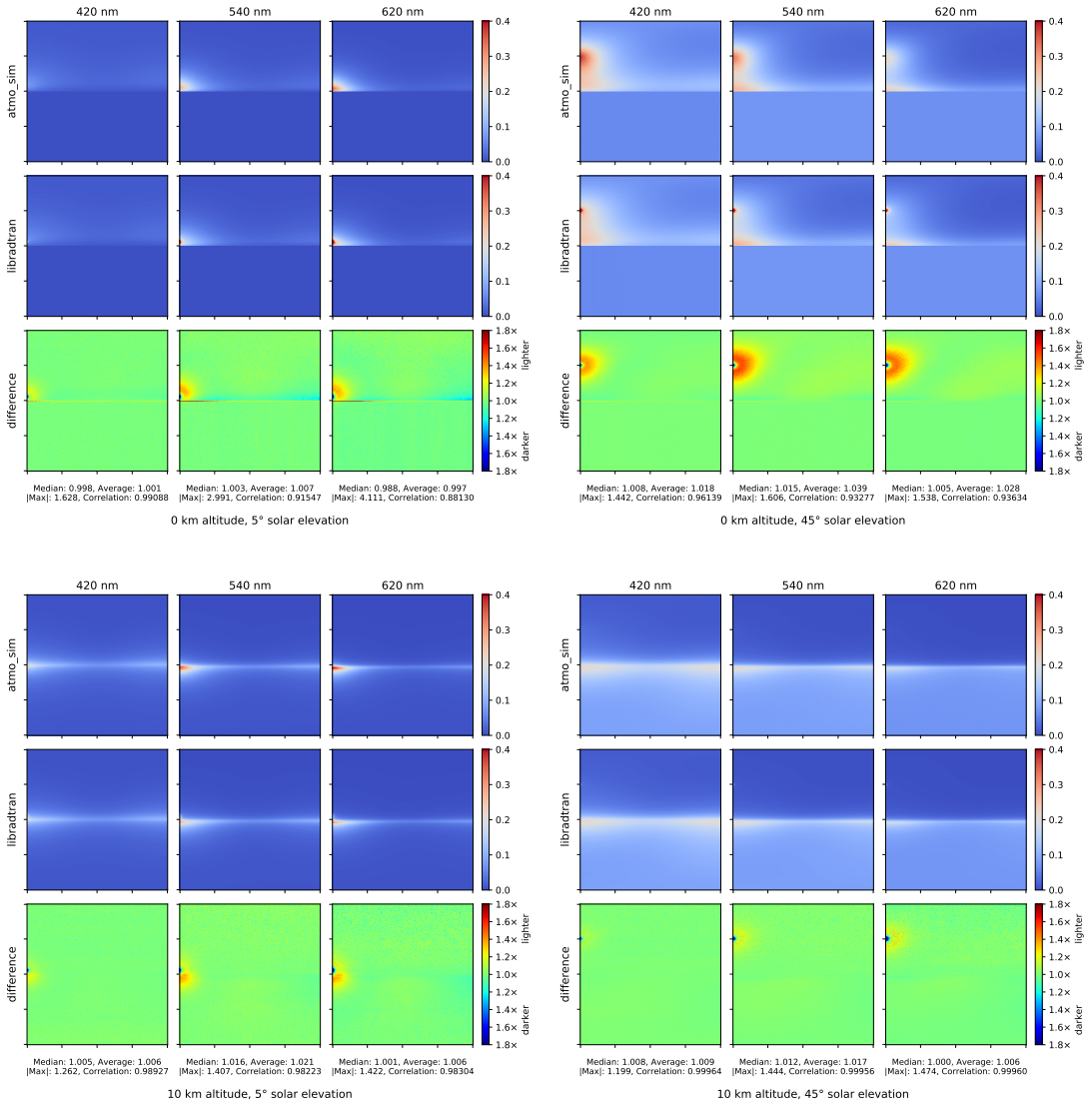


Figure 3.54: Similar to Figure 3.53, but this time with an atmosphere containing also the INSO, WASO and SOOT aerosols from OPAC (corresponding to visibility 59.4 km, i.e., using the continental average particle concentration). Notice that the colour scales are the same as in Figure 3.55 and 3.53, which allows one to see that errors appeared because of differently modelled aerosol phase functions, which we discuss in the text. The errors are more pronounced around the solar disk due to the used INSO phase function being less forward scattering. Notation and scale: See caption of Figure 3.53, where the image notations and scales are explained.

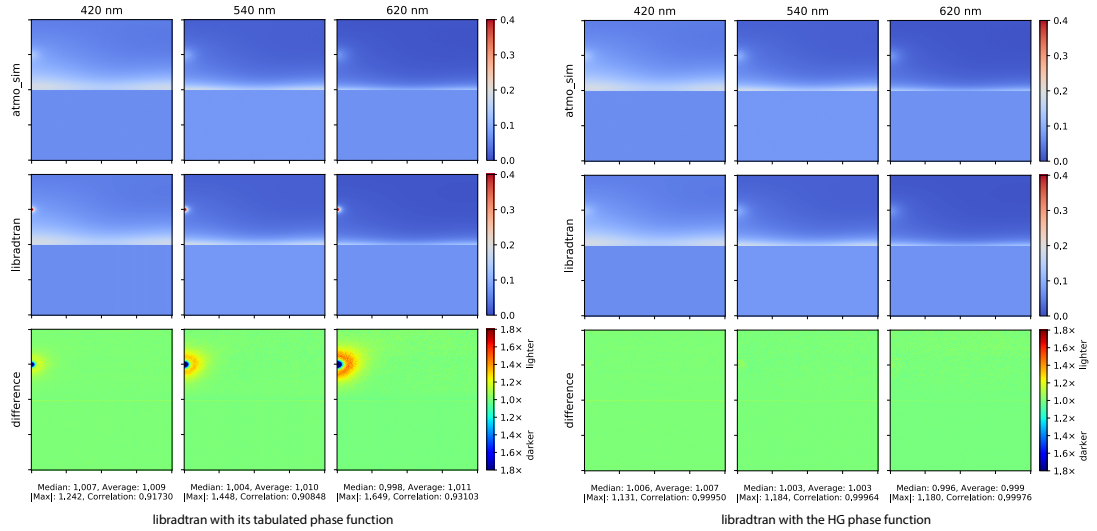


Figure 3.55: A comparison of the Henyey-Greenstein (HG) phase function used in `atmo_sim` against a tabulated phase function used in `libradtran`, at 0 km altitude with 45° solar elevation. The atmosphere contains only molecules and INSO aerosols. **Left:** `atmo_sim` uses HG phase function (see text for details), `libradtran` uses their tabulated phase function. Notice that the scattering in `atmo_sim` is less forward, so the sun energy is blurred in a wider area, which makes the inner part darker and outer part brighter. **Right:** Both use the same HG phase function, the difference disappears. Notation and scale: See caption of Figure 3.53, where the image notations and scales are explained.

Empirical validation

As discussed in Section 3.5.2, the modelled atmosphere uses scatterer particle concentration profiles provided by OPAC which exhibit a distinct lower haze layer. Here, anecdotal real-life imagery of what a similar configuration looks from higher observer altitudes is provided. Figure 3.56 shows photographs taken on a fairly typical clear autumn day in Central Europe during high pressure weather and comparable render produced by `atmo_sim`. In this region, the presence of a marked, hazy inversion layer that can be seen in these images is typical for not just autumn days, but generally high pressure scenarios where the atmospheric layering is so stable that no cloud-forming convection can start. On such days, clear, cloud-free skies can be expected from dawn to dusk: in other words, exactly the conditions that a clear sky model attempts to represent. The only change during such a day is usually a gradual rise of the inversion layer during the course of the day, and a more or less pronounced increase in turbidity: both are due to residual convection within the inversion layer. Even though the atmospheric conditions captured in the photos are different then those modelled by the Prague Sky Model, the simulated result still provides a good match.

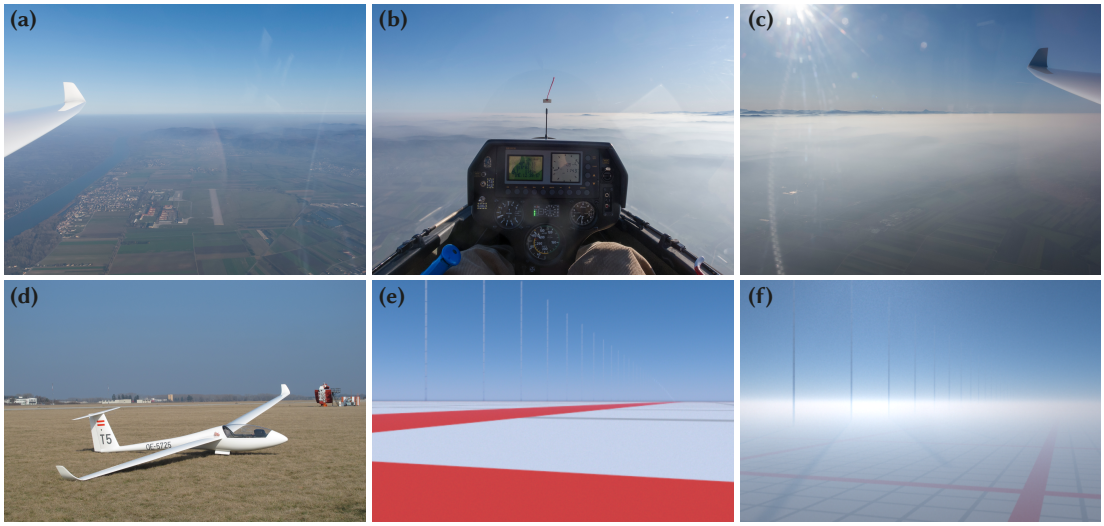


Figure 3.56: A low inversion layer on a clear autumn day in Central Europe. Photos (a), (b), (c) were taken seconds apart from on board of a glider that was flying at about 1000 m above ground level, photo (d) was taken approximately 30 minutes after the in-flight images (all photos courtesy of Alexander Wilkie). The in-flight images show a distinct inversion layer transition and exceptionally clear viewing conditions above it: the mountaintops in image (c) are between 100 km and 150 km away. Note that the haze layer with its tops at around 500 m above ground level was homogeneous across the entire region: the strong forward scattering coming from the solar direction, and the comparative transparency in the antisolar direction, moved with the aircraft viewpoint - the haze was not actually denser in the direction of the sun. On the other hand, no atmospheric layering is visible from the ground in image (d): the inversion layer does not produce any noticeable brighter stripe along the horizon. However, noticeable ground haze for horizontal viewing directions can be observed: the trees and buildings at the edge of the airfield are less than 2 km away. Images (e), (f) show simulations in `atmo_sim` for 2 m and 2 km observer altitudes in the Columns scene, respectively. Note that the photos capture an example of a fairly sharp inversion layer transition at a lower altitude than the OPAC atmospheres used in the model. Therefore, images (e), (f) do not attempt to precisely match photos (d), (c), respectively, they rather demonstrate similar features.

3.12.3 Fitting in-scattered radiance

This section describes the entire process of fitting the radiance component of the Prague Sky Model in detail. The content of this section, both text and figures, is taken with minimum modifications from the paper of Wilkie et al. [2021] (Section 1 in the Supplemental document), majority of it also appeared in the doctoral thesis by Hošek [2019] (on pages 58 – 68). This section does not contain any contribution of the author, it is not necessary for a general overview of the Prague Sky Model, but it is crucial for understanding the fitting process and its reproducibility.

Choice of mathematical approach

We first extensively experimented with techniques similar to that of the Perez et al. [1993] model and its descendants. The approach of these models is to assume the sky dome radiance patterns to consist of separable features – e.g., a gradient between the zenith and the horizon plus a radial bright patch around the Sun. In these models, radiance is calculated as a function of ray direction (given as a pair of angles, a "solar" and a "zenith" angle) and a small number of configuration factors: numbers that control the strength of each individual feature, and which are found by means of non-linear optimisation. This approach works as long as the number of features is kept low: but that obviously limits the range of radiance patterns the model can reliably reproduce.

For our purposes, we need an expression that works for a full sphere instead of just the upper hemisphere: in particular, it has to be able to handle the discontinuity which is present at the horizon in most radiance configurations. It also has to have terms that approximate the features of the sky well, including two phenomena specific to twilight skies: post sunset, the Earth casts a shadow onto the atmosphere, which produces a wedge of darker colour at the horizon. Above it, there is a second wedge of brighter pinkish back-scattered light (called "Belt of Venus" or "anti-twilight arch"): Figure 3.18 shows how these features develop as the sun goes beneath the horizon.

Theoretically, a suitable mix of features could be devised by educated guess and trial and error. However, even if we managed to find such features, the fitting process becomes slower with each new parameter in a non-linear fashion, plus more memory consuming and prone to getting stuck in local minima. Extensive experimentation showed the old feature-based approach is simply not suitable for radiance fitting on the full sphere any more: there are too many features in fully spherical radiance patterns which cannot be cleanly separated. This even applies to models which attempted to separate just the polarisation patterns: Kreuter and Blumthaler [2013] also only managed to work with the upper hemisphere.

This is why we opted for an entirely new approach: we obtain the radiance pattern of the sky as a sum of outer products of single variable functions. The functions themselves are free-form, tabulated and were obtained by Canonical Polyadic Decomposition (CPD) [Kolda and Bader, 2009], a process very similar to SVD low rank approximation. This approach can be thought of as a specialised compression scheme, however it is also essentially a decomposition of the radiance pattern into an optimal orthogonal set of "features".

The methods that we describe in the next sections all rely on tensor and matrix decompositions. An alternative choice could have been to use neural networks, similar to [Satilmis et al., 2016; Hold-Geoffroy et al., 2019; Zhang et al., 2019]. However, while learning approaches do have merits, reliability of reconstruction is not one of them. Additionally, they tend to incur a higher runtime overhead than our model.

Input parametrisation

It is desirable to choose a parametrisation in which the features are as axis-aligned as possible, as that makes the input matrix easy to decompose into separable matrices by CPD / SVD. For a given solar elevation η, the natural parametrisation

of a sky dome model is the view direction, represented as a unit vector **v**. For better separability we transform it into a set of angles (see Figure 3.57).

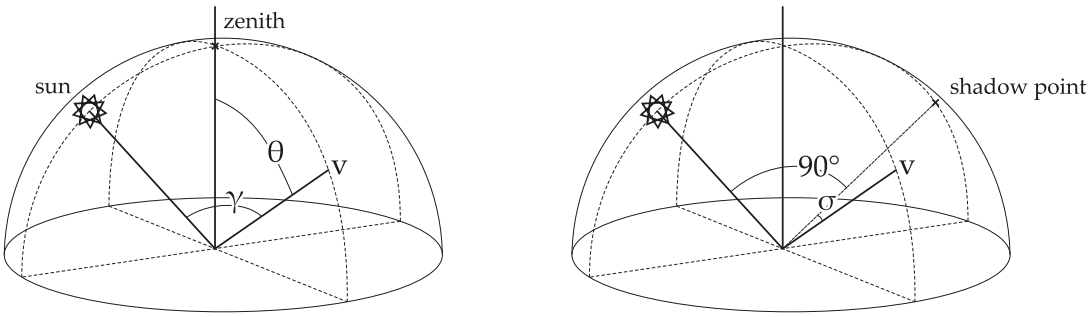


Figure 3.57: The angles used in the model.

The first one is the solar angle γ. The solar angle is the angle formed by the view direction and the direction towards the center of the solar disc. This makes the gradient of the solar glow roughly parallel to the first axis – *roughly* because the solar glow does not actually form a perfect circle around the sun, but tends to extend further to the sides and downward.

We also need the model to describe Earth's shadow. So when fitting images of post-sunset skies, we chose the shadow angle σ to be second input angle / axis. This angle is formed by the view ray and a *shadow point* – an imaginary point lying at 90° away from the center of the solar disc in the direction of the zenith. The shadow line is perpendicular to the solar angle gradient and parallel to the shadow angle gradient, which makes it aligned with both axes.

For regular daytime skies, no shadow is visible, and using the shadow angle as an input parameter provides no benefit since its axis is not parallel or perpendicular to any feature visible on the sky. We instead use the zenith angle θ as the second parameter for these configurations. This is the angle formed by the view direction and the direction towards the zenith. This makes the horizon perpendicular to the axis defined by the horizon angle.

The model switches between these two modes at solar elevation 0°, when the zenith point aligns exactly with the shadow point, which makes the transition seamless. To present this approach in a unified manner, we introduce an angle α, which we for lack of a better name call simply the *zenith/shadow angle*, which is equivalent to the zenith angle for solar elevations greater than 0° and equivalent to the shadow angle otherwise. Note that this makes the horizon not aligned with any axis in post-sunset skies, which makes fitting of the horizon tricky. The image emphasis process described later was developed to make horizon fitting more accurate in these conditions.

High altitude angle correction

Our model consists of a finite number of fitted sky dome configurations, and intermediate states have to be interpolated in a way that generates plausible sky dome appearance. It is a key feature of the CPD / SVD separation that this is actually possible - at least for some sky dome features.

A case that works is the circumsolar glow: there, taking the fitted data for a specific solar elevation (e.g., $\eta = 20°$), and using it to generate a different

elevation (e.g., $\eta = 30°$) yields usable results: the $\mathbf{v} \rightarrow (\gamma, \alpha)$ re-projection process warps the image correctly, and the solar glow gets moved to the right place.

This unfortunately does not work for the horizon, as that changes in a manner that is too complex for simple re-projection to handle. It only appears as a line at $\theta = 90°$ for altitude = 0: and as can be seen in Figure 3.16, it moves downwards for higher observer altitudes, and turns into a curve. If we had data for just two observer altitudes, e.g., 100 m, and 100 km, interpolation between these states would contain two blended horizons, instead of a single one.

We fix this issue via the way the θ and σ angles (and thereby also α) are calculated. If the view direction $\mathbf{v}$ is tangent to the Earth's surface, its θ will always be 90°, regardless of altitude.

The un-corrected way of calculating the angles is as follows: the directions towards the solar, zenith and shadow points are given as unit vectors $\mathbf{s}$, $\mathbf{z}$ and $\mathbf{u}$. Assuming that the z-axis points upwards, $\mathbf{z} = (0, 0, 1)$, the angles can be calculated as follows:

$$\begin{aligned} \gamma &= \cos^{-1}(\mathbf{v} \cdot \mathbf{s}) \\ \theta &= \cos^{-1}(\mathbf{v} \cdot \mathbf{z}) \\ \sigma &= \cos^{-1}(\mathbf{v} \cdot \mathbf{u}) \end{aligned} \tag{3.13}$$

With an observer altitude above ground, the tangent from the camera origin towards the horizon is not perpendicular to the zenith direction $\mathbf{z}$. To correct that, we project the point of tangency p_t onto the line from Earth's center to the observer to obtain the *virtual ray origin*, p_o (see Figure 3.58). We denote $\mathbf{v}'$ the direction from p_o to p_t. The direction $\mathbf{v}'$ can be expressed in terms of the original direction $\mathbf{v}$, camera altitude alt and Earth radius r as:

$$\mathbf{v}' = \text{normalize}(\mathbf{v} - \mathbf{corr}) \tag{3.14}$$

with the *correction vector* $\mathbf{corr}$ being defined as:

$$\begin{aligned} \mathbf{corr} &= (0, 0, \frac{c}{t}) \\ c &= r + alt - \frac{r^2}{r + alt} \\ t &= \sqrt{(r + alt)^2 - r^2} \end{aligned} \tag{3.15}$$

This correction to the view direction is applied in the model for the purpose of calculating the zenith and shadow angles.

The core function

The function which evaluates the in-scattered radiance is a function of two parameters γ, α. The function is internally represented as an outer product of two single parameter functions:

$$\mathbb{F}(\gamma, \alpha) = \sum_{i=1}^{n} \mathbb{F}^{(i)}_{solar}(\gamma) \otimes \mathbb{F}^{(i)}_{zenith/shadow}(\alpha) \tag{3.16}$$

The functions $\mathbb{F}_{solar}$ and $\mathbb{F}_{zenith/shadow}$ are tabulated and provided as part of the model. The tabulated functions are obtained by re-projecting the fish-eye image

into the (γ, α) space, essentially producing a 2D look-up table of $\mathbb{F}(\gamma, \alpha)$, and then decomposing the look-up table into outer vector products using CPD. The process is described in detail later in this section.

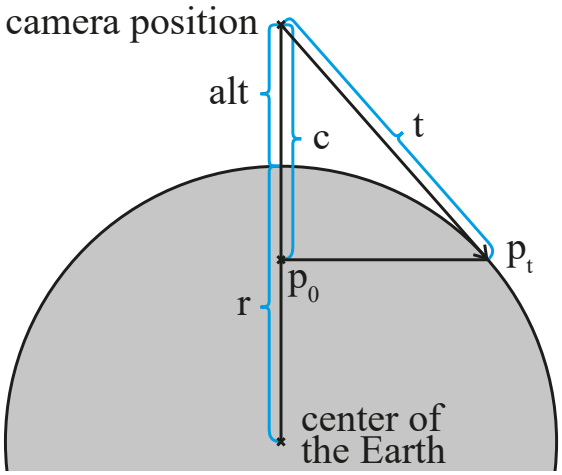


Figure 3.58: The horizon correction geometry. r - Earth radius, alt - camera altitude, c - correction length, t - tangent length, p_t - point of tangency, p_o - virtual ray origin.

Improved horizon fitting with image emphasis

Recall that in order to make the decomposition perform properly, the angle α was defined so that the horizon is a horizontal line in the re-projected image. This only holds for daytime solar elevations: post sunset the horizon is not axis-aligned any more, so the composition performs poorly in those cases. To deal with this, we remove the sharp horizon from the images prior to transforming them into (γ, α) space, and only then perform the fitting which then works satisfactorily on a blurred horizon that is not axis aligned. This process, which we call *pre-emphasis*, is reverted when using the model: the *de-emphasis* we perform then returns the sharp horizon transition to its place. We can accurately do this, as the location of the horizon is analytically known.

The pre-emphasis process works as follows: We denote I_{orig} the original input image and $I_{orig}(x)$ its value at pixel position x. The input image I_{orig} is cut into two parts very close to the horizon line (we chose a value of $30'$ above the horizon). The top/above horizon part is left intact. The bottom / below horizon part is deleted and infilled using the standard `regionfill` algorithm of Matlab version 2019b, which smoothly interpolates inward. This essentially removes the sharp horizon transition and leaves the bottom part of the image a completely featureless smooth gradient with brightness equivalent to that of the sky dome area just above the horizon. We denote this newly infilled image as I_{infill}. We also note a single value κ denoting the ratio of the average brightness of the original below-horizon area relative to the brightness of the newly infilled area.

By linearly interpolating between I_{orig} and I_{infill}, we create a guide image I_{guide}. The control value for the interpolation is a function of κ and the pixel position. When κ is high, meaning that the below-horizon part of the image was already bright enough in the original, we mostly leave the original intact, except

for the very bottom part near the nadir, which we always replace by the infill, because it's generally always noisy. Formally:

$$\begin{aligned} I_{guide}(x) &= \text{lerp}\left(I_{orig}(x), I_{infill}(x), V_c(x)\right) \\ V_c(x) &= \text{sat}\left(\text{sat}\left(\frac{\theta(x) - \pi/2}{\pi/2}\right) + \text{sat}\left(\frac{\kappa - 0.5}{-0.2}\right)\right) \end{aligned} \tag{3.17}$$

where:

- V_c is the control value of the linear interpolation.
- sat is the *saturate* function that clamps the value to $[0, 1]$.
- $\theta(x)$ is the zenith angle value at pixel location x.

The control value V_c consists of two terms: the first one makes sure that the original image is always gradually replaced by the infilled image, starting at $\theta = 90°$ (at the horizon) and progressing towards $\theta = 180°$ (the nadir). The second term makes sure that if the bottom part of the original image was too dark, it is replaced completely, the effect being gradually applied depending on κ, starting at $\kappa = 0.5$ and finishing at full strength at $\kappa = 0.3$.

Having the guide image ready, we calculate a pixel-wise ratio between the guide and the original image:

$$I_{ratio}(x) = \frac{I_{orig}(x)}{I_{guide}(x)} \tag{3.18}$$

The ratio image is then used to define the de-emphasis, a function of zenith angle θ:

$$E(z) = \underset{x;\ \theta(x)=z}{\text{mean}} I_{ratio}(x) \tag{3.19}$$

The function $E(z)$ is tabulated and becomes part of the model data. Next, the pre-emphasized image is calculated by applying the inverse of the de-emphasis function:

$$I_{preemph}(x) = \frac{I_{orig}(x)}{E(\theta(x))} \tag{3.20}$$

This pre-emphasized image is then used as an input for CPD.

The fitting process

We now have all the components required to perform the fitting. To recapitulate the whole fitting process:

1. We start with I_{orig}. This is the reference rendering of the sky dome produced by the path tracer.
2. Pre-emphasis is performed on I_{orig}, yielding a pre-emphasized image $I_{preemph}$ and a de-emphasis function E.
3. $I_{preemph}$ is re-projected into (γ, α) space, essentially producing a 2D look-up table of $\mathbb{F}(\gamma, \alpha)$.

4. The re-projected image is partially infilled and filtered. More on that later in this section.

5. A CPD decomposition is performed, yielding pairs of one-dimensional tabulated functions $\mathbb{F}^{(i)}_{solar}$ and $\mathbb{F}^{(i)}_{zenith/shadow}$.

The final products of the fitting are:

- The de-emphasis function E
- The tabulated functions $\mathbb{F}^{(i)}_{solar}$ and $\mathbb{F}^{(i)}_{zenith/shadow}$.

These constitute all the data required to render the sky dome using the analytical model.

We have chosen the dimensions of the (γ, α) re-projected image to be 361×361 (i.e., $0.5°$/pixel since the valid values of γ and α are $0°$ - $180°$). This image is computed by transforming each pair of γ, α values into the reference rendering followed by bilinear filtering to avoid artefacts. This re-projected image is then decomposed into an outer vector product using the CPD low rank approximation algorithm. Note that in theory, we could extend this process by unwrapping the input into a three or even higher dimensional look-up table, e.g., parametrised by (γ, θ, σ). CPD is a tensor decomposition algorithm and would deal with the resulting tensor natively. The problem of this approach is that the valid combinations of angles form a 2D manifold inside this 3D space – in other words the tensor is mostly undefined, which makes the decomposition unstable.

Even in 2D, the issue of undefined values requires us to infill parts the re-projected image. Not all combinations of angles are valid, e.g., in a sky where the sun is at the horizon there is no direction that would correspond to both γ and α being 0. The valid combinations form a parallelogram, see Figure 3.59. The CPD algorithm deals natively with undefined values, however there is no guarantee what the undefined part is going to look like in the resulting approximation. When discussing the high altitude angle correction, we claimed that data from one sky dome configuration can be re-used, e.g., for other solar elevations if it is suitably re-projected. This is true, but a potential issue arises due to the changing shape of the parallelogram of valid combinations: upon re-projection, we might attempt to read from an undefined part of the (γ, α) image.

To fix this, the re-projected image has to be partially infilled. The valid area of the tensor is dilated, and the missing data is again filled using Matlab's `regionfill` algorithm. The amount of dilation is the minimal amount required to cover the intermediate values between the provided fittings (see Table 3.2).

After infilling, the tensor also has to be filtered, as CPD decomposition not only retains noise present in the input, but moves it to (γ, α) space: so instead of grain-like Monte Carlo noise, there are ringing artefacts. To be usable, the final fitting has to be completely devoid of any such artefacts: even the slightest unevenness is *very* apparent in renderings, especially as they are usually dissimilar across spectral channels, and show up as rainbow effects. So the reference renderings are filtered (using the `wdenoise` wavelet noise reduction algorithm found in Matlab version 2019b), and after the image is re-projected to a tensor and infilled, it is again filtered, this time using a gaussian blur with the standard deviation 1 both in the solar and zenith/shadow axis (recall that the resolution of the tensor is $0.5°$ per pixel).

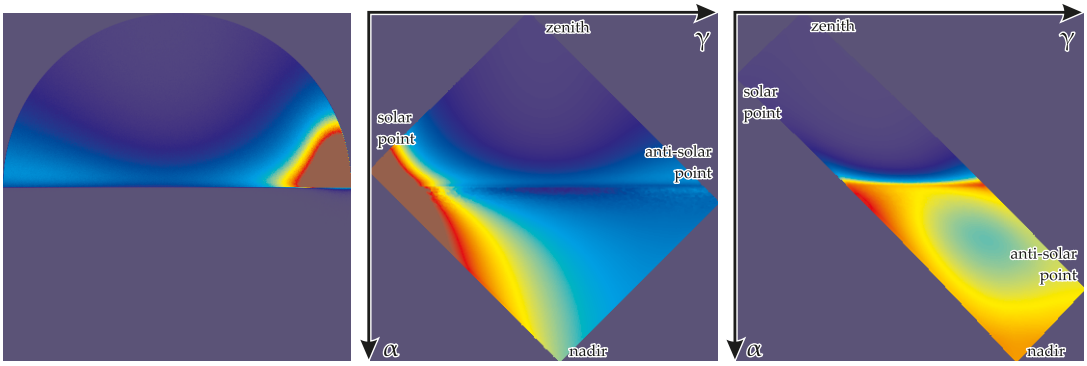


Figure 3.59: **Left:** A single spectral channel of the input fish-eye image in false colour. Solar elevation 8°, observer altitude 15 m, so the image is almost completely zero below the horizon line. **Middle:** The same image transformed to (γ, α) space: the valid combinations of γ and α form a parallelogram. The below-horizon part of the image is amplified by the pre-emphasis process. **Right:** The (γ, α) remapping of a different sky dome, solar elevation 55°: this illustrates the changing shape of the parallelogram of valid values.

Regarding the decomposition itself, we have chosen to use decomposition rank $n = 9$, i.e. the decomposition produces nine sets of vectors. In all images we have tested, $n = 9$ produces a decomposition that explains $>$ 99.5 % of the variance in the tensor.

The tensor decomposition produces vectors of length 361 (0.5° increments). These do not have to be distributed in their entirety in the final model: they essentially represent tabulated functions, and the samples do not have to be placed uniformly. We have chosen to sample $\mathbb{F}_{solar}$ densely at lower angles (areas directly surrounding the sun) and sparsely around the anti-solar point, giving us satisfactory results at 275 samples. Similarly, $\mathbb{F}_{zenith/shadow}$ is sampled sparsely around the zenith and nadir and densely around the horizon, giving us 205 samples. The same approach has been used for the de-emphasis function, which is sampled more densely around the horizon, giving us 118 samples.

The evaluation process

The complete radiance function is evaluated as follows: for a given viewing direction the angles γ, α and θ are computed first. Then the 9 pairs of tabulated functions $\mathbb{F}^{(i)}_{solar}(\gamma)$, $\mathbb{F}^{(i)}_{zenith/shadow}(\alpha)$ are looked up and combined according to (3.16). Finally, the result is multiplied by looked-up value of the de-emphasis function $E(\theta)$.

3.12.4 Error plots

This section provides box plots of the normalised mean absolute errors with respect to every parameter of the Prague Sky Model. In these plots, the red line is the median, the blue box goes from the first to the third quartile, and the whiskers are the minimum and maximum values. The labelled values represent parameter values for which reference images were computed in the brute force rendered dataset, and the errors shown there are between the fit and those images. In between the labelled values, the interpolation error had to be estimated, as intermediate reference images were generally not available. The estimate is the difference between the two neighbouring fits: while this is a loose bound on the true interpolation error, it can best be interpreted as "how wrong could one get if one did not interpolate at all", and not as the actual interpolation error. If this difference-based estimate was low, it would mean there is no point in using the non-trivial image interpolation scheme proposed in this chapter, because normal pixel-wise interpolation would already work well. This, in turn, means that the sometimes quite large interpolation error seen there is not automatically a bad thing: if it were too low, the proposed interpolation scheme would be pointless.

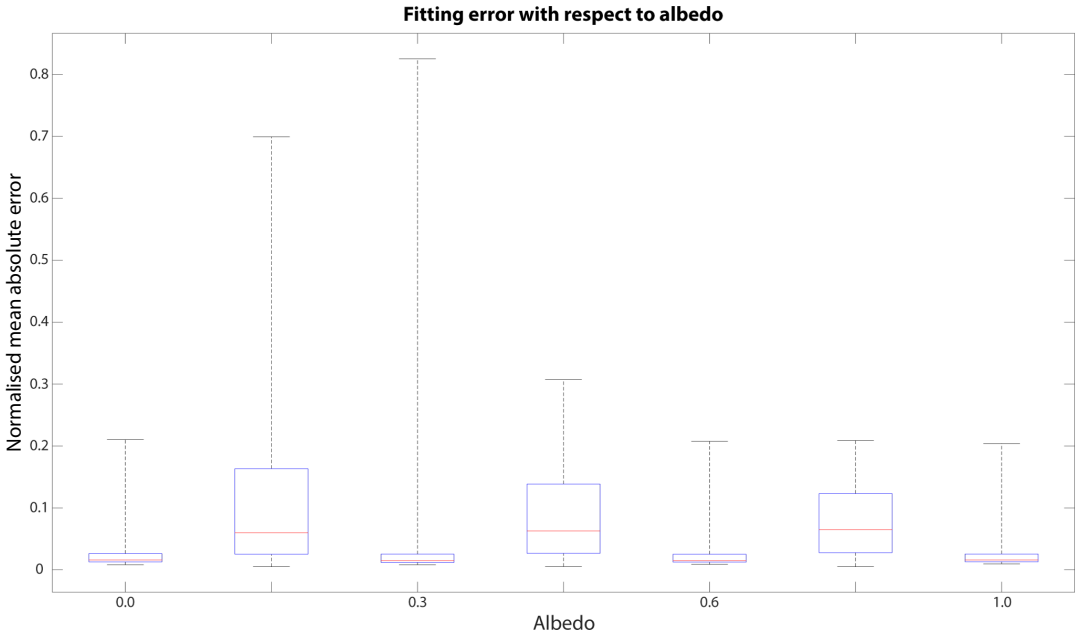


Figure 3.60: A box plot of the normalised mean absolute errors for ground albedos covered by the Prague Sky Model.

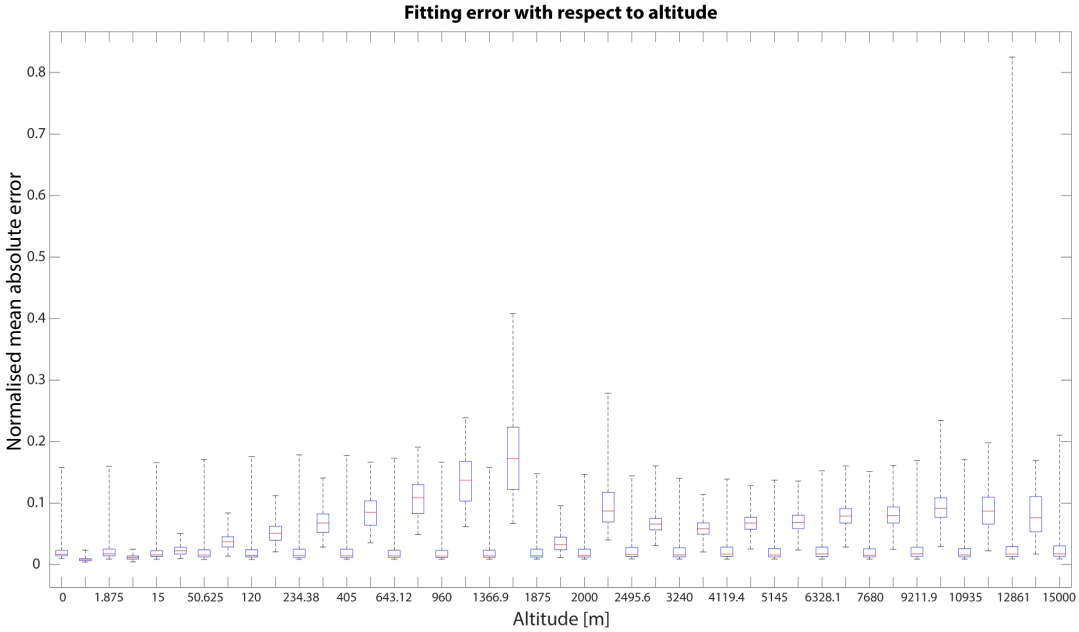


Figure 3.61: A box plot of the normalised mean absolute errors for observer altitudes covered by the Prague Sky Model.

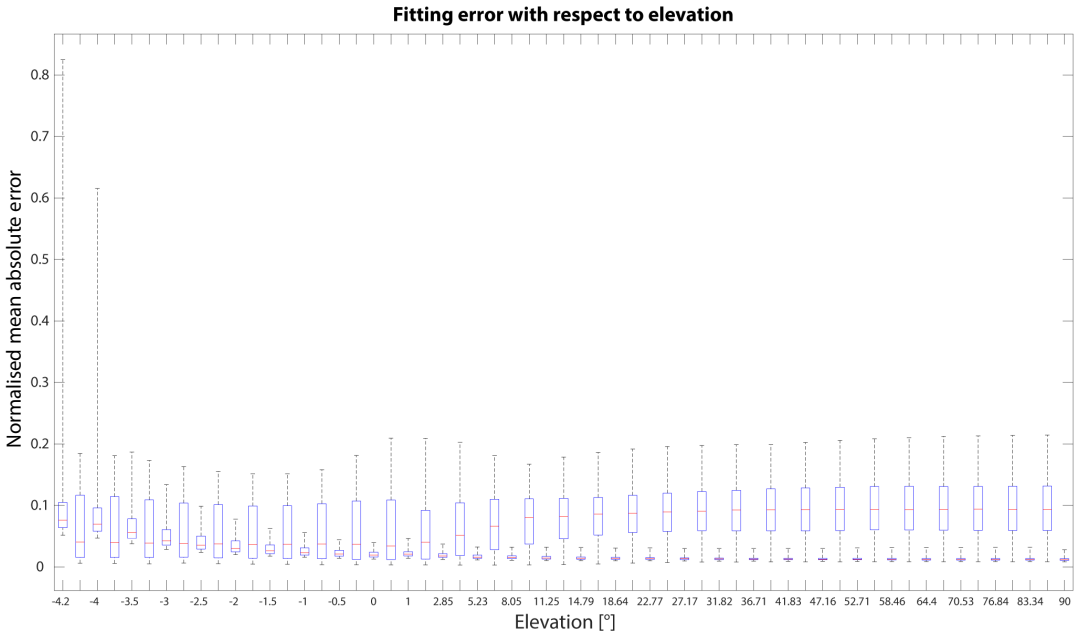


Figure 3.62: A box plot of the normalised mean absolute errors for solar elevations covered by the Prague Sky Model. Note how the error on the labelled values increases with decreasing solar elevation due to the higher noise levels in these images.

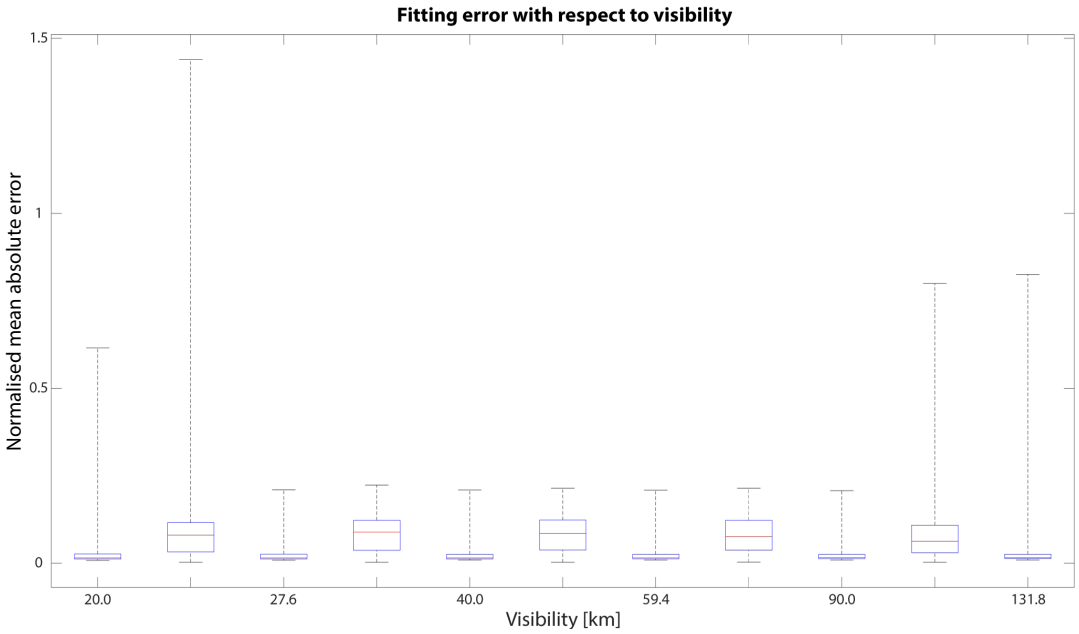


Figure 3.63: A box plot of the normalised mean absolute errors for visibilities covered by the Prague Sky Model.

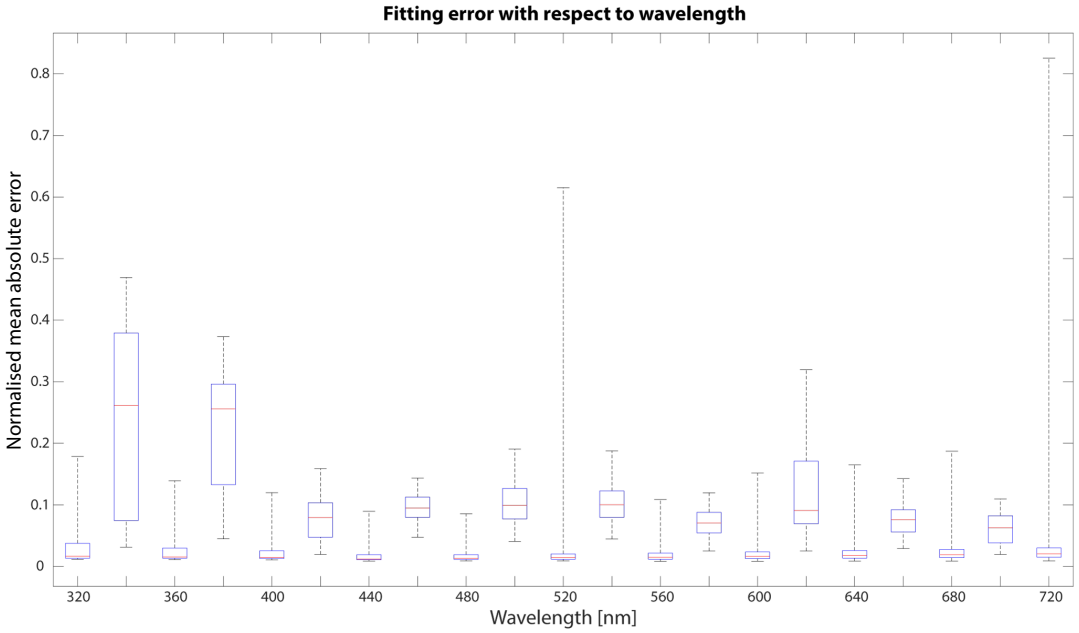


Figure 3.64: A box plot of the normalised mean absolute errors for wavelengths covered by the Prague Sky Model.

3.12.5 Wavelength compression

Error	Wavelength channels to omit
2.5%	49, 50
5.0%	48, 50, 52
7.5%	15, 34, 35, 48, 49, 50, 51, 53
10.0%	15, 19, 34, 35, 48, 49, 50, 51, 53
12.5%	15, 19, 33, 35, 37, 47, 49, 50, 51, 53
15.0%	14, 16, 19, 33, 35, 37, 47, 48, 49, 50, 52, 53
17.5%	14, 16, 19, 21, 33, 34, 35, 36, 47, 48, 49, 50, 51, 53
20.0%	14, 16, 19, 21, 24, 25, 33, 34, 35, 36, 47, 48, 49, 50, 51, 53
22.5%	14, 15, 19, 20, 24, 25, 31, 33, 34, 35, 36, 47, 48, 49, 50, 51, 53, 54
25.0%	14, 15, 18, 20, 24, 25, 31, 33, 34, 35, 36, 38, 47, 48, 49, 50, 51, 52, 54
27.5%	14, 15, 18, 20, 24, 25, 31, 33, 34, 35, 36, 38, 47, 48, 49, 50, 51, 52, 54
30.0%	13, 14, 15, 18, 20, 24, 25, 31, 33, 34, 35, 36, 38, 47, 48, 49, 50, 51, 52, 54

Table 3.3: A list of wavelength channels that can be omitted for various error thresholds from the SWIR extension in order to decrease the fitted dataset size.

Conclusion

Given the importance of Monte Carlo (MC) integration not only to image synthesis but also to many other scientific fields, a vast body of research exists that focuses on mitigating effects of its inherent problem – variance. Out of the many possible approaches to variance reduction, this thesis presented three, each significantly different from the others to demonstrate variability of this problematic. The result are three advanced methods increasing efficiency of MC integration in rendering.

First, we used the standard approach of importance sampling, i.e., finding a sampling technique as close to being proportional to the integrand as possible. Motivated by practical needs of a production path tracer, we focused on direct illumination calculation, its speed and robustness. We proposed an adaptive solution for direct illumination sampling based on a novel statistical model of direct illumination and learning its parameters from previous samples using Bayesian regression. The method is unbiased, scalable, virtually free of any preprocessing, and robust even in early stages of calculation.

In the second approach, we addressed the situation when finding a single sampling technique that would be a good match for the entire integrand is infeasible and a combination of multiple techniques has to be used. We investigated the commonly used multiple importance sampling (MIS) framework and found a room for improvement in its weighting functions. We derived optimal weighting functions that provably minimize the variance of MIS estimators and perform even better than predicted by existing variance bounds. The new weights also open the way for novel design considerations for selecting appropriate sampling techniques in integration problems and we proposed several examples of those.

Finally, the third approach considered types of light transport that are difficult to simulate using any sampling techniques but can be separated from the rest and pre-computed. We focused on rendering of the sky as an ideal example and reviewed the Prague Sky Model, a feature-rich clear sky model created by fitting a large set of pre-computed reference images of the sky. It advances the state of the art of sky models in almost every aspect and allows any renderer to achieve realistic sky appearance without any atmospheric simulation overhead. We described our contribution to development of this method and presented our own extension covering the full spectral range of terrestrial solar irradiance, enabling usage of such pre-computed models for purposes other than renderings intended to mimic the perception of human observers, such as thermal analysis, and photovoltaic plant yield simulations.

We believe that the three presented methods significantly improve rendering efficiency and quality, and contribute valuable insights to the field of MC integration in image synthesis. We would also like to emphasize that the methods are not purely theoretical. In fact, the first and third method were both integrated in the Corona renderer and have been successfully used there to this day demonstrating also their practical utility.

Bibliography

Gail Anderson, Shepard Clough, F. Kneizys, J. Chetwynd, and Eric Shettle. AFGL atmospheric constituent profiles (0–120km). *Environmental research papers*, 954, 1986.

James Arvo. Stratified sampling of spherical triangles. In *Proceedings of SIGGRAPH 1995*, pages 437–438, 1995.

Gilles Aubert and Pierre Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*. Springer, 2nd edition, 2006.

Niels Billen, Björn Engelen, Ares Lagae, and Philip Dutré. Probabilistic visibility evaluation for direct illumination. *Computer Graphics Forum*, 32(4):39–47, 2013.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York Inc., 2006.

Barry A. Bodhaine, Norman B. Wood, Ellsworth G. Dutton, and James R. Slusser. On Rayleigh optical depth calculations. *Journal of Atmospheric and Oceanic Technology*, 16(11):1854–1861, 1999.

Malik Boughida and Tamy Boubekeur. Bayesian collaborative denoising for monte carlo rendering. *Computer Graphics Forum*, 36(4):137–153, 2017.

Jonathan Brouillat, Christian Bouville, Brad Loos, Charles Hansen, and Kadi Bouatouch. A bayesian monte carlo approach to global illumination. *Computer Graphics Forum*, 28(8):2315–2329, 2009.

Eric Bruneton. A qualitative and quantitative evaluation of 8 clear sky models. *IEEE transactions on visualization and computer graphics*, 23(12):2641–2655, 2016.

Eric Bruneton and Fabrice Neyret. Precomputed atmospheric scattering. *Computer graphics forum*, 27(4):1079–1086, 2008.

Brian C. Budge, John C. Anderson, and Kenneth I. Joy. Caustic forecasting: Unbiased estimation of caustic lighting for global illumination. *Computer Graphics Forum*, 27(7):1963–1970, 2008.

Norbert Bus, Nabil H. Mustafa, and Venceslas Biri. IlluminationCut. *Computer Graphics Forum*, 34(2):561–573, 2015.

Olivier Cappé, Arnaud Guillin, Jean-Michel Marin, and Christian P. Robert. Population Monte Carlo. *Journal of Computational and Graphical Statistics*, 13(4):907–929, 2004.

Olivier Cappé, Randal Douc, Arnaud Guillin, Jean-Michel Marin, and Christian P. Robert. Adaptive importance sampling in general mixture classes. *Statistics and Computing*, 18(4):447–459, 2008.

Chaos Czech a.s. Corona Renderer, 2023. http://corona-renderer.com/.

Petrik Clarberg and Tomas Akenine-Möller. Exploiting visibility correlation in direct illumination. *Computer Graphics Forum*, 27(4):1125–1136, 2008.

William M. Cornette and Joseph G. Shanks. Physically reasonable analytic expression for the single-scattering phase function. *Applied Optics*, 31(16):3152–3160, 1992.

Jean-Marie Cornuet, Jean-Michel Marin, Antonietta Mira, and Christian P. Robert. Adaptive multiple importance sampling. *Scandinavian Journal of Statistics*, 39(4):798–812, 2009.

Michael Donikian, Bruce Walter, Kavita Bala, Sebastian Fernandez, and Donald P. Greenberg. Accurate direct illumination using iterative adaptive sampling. *IEEE Transactions on Visualization and Computer Graphics*, 12(3): 353–363, 2006.

Randal Douc and Arnaud Guillin. Minimum variance importance sampling via population monte carlo. *ESAIM: Probability and Statistics*, 11:427–447, 2007.

Víctor Elvira, Luca Martino, David Luengo, and Mónica F. Bugallo. Generalized multiple importance sampling. arXiv:1511.03095, 2015.

Víctor Elvira, Luca Martino, David Luengo, and Mónica F. Bugallo. Heretical multiple importance sampling. *IEEE Signal Processing Letters*, 23(10), 2016.

Claudia Emde, Robert Buras-Schnell, Arve Kylling, Bernhard Mayer, Josef Gasteiger, Ulrich Hamann, Jonas Kylling, Bettina Richter, Christian Pause, Timothy Dowling, and Luca Bugliaro. The libRadtran software package for radiative transfer calculations (version 2.0.1). *Geoscientific Model Development*, 9(5):1647–1672, 2016.

Shaohua Fan, Stephen Chenney, Bo Hu, Kam Wah Tsui, and Yu Chi Lai. Optimizing control variate estimators for rendering. *Computer Graphics Forum*, 25 (3):351–357, 2006.

Shaohua Fan, Yu-Chi Lai, Stephen Chenney, and Charles Dyer. Population monte carlo sampler for rendering. Technical Report 1613, Department of Computer Sciences, University of Wisconsin-Madison, 2007.

Sebastian Fernandez, Kavita Bala, and Donald P. Greenberg. Local illumination environments for direct lighting acceleration. In *Proceedings of the 13th Eurographics workshop on Rendering*, pages 7–14, 2002.

Manuel N. Gamito. Solid angle sampling of disk and cylinder lights. *Computer Graphics Forum*, 35(4):25–36, 2016.

Iliyan Georgiev, Jaroslav Křivánek, Stefan Popov, and Philipp Slusallek. Importance caching for complex illumination. *Computer Graphics Forum*, 31(2pt3): 701–710, 2012a.

Iliyan Georgiev, Jaroslav Křivánek, Tomáš Davidovič, and Philipp Slusallek. Light transport simulation with vertex connection and merging. *ACM Transactions on Graphics*, 31(6):192:1–192:10, 2012b.

Paul Glasserman. *Monte Carlo method in financial engineering*. Springer-Verlag, New York, USA, 2003.

I.E. Gordon, L.S. Rothman, R.J. Hargreaves, R. Hashemi, E.V. Karlovets, F.M. Skinner, E.K. Conway, C. Hill, R.V. Kochanov, Y. Tan, P. Wcisło, A.A. Finenko, K. Nelson, P.F. Bernath, M. Birk, V. Boudon, A. Campargue, K.V. Chance, A. Coustenis, B.J. Drouin, J.–M. Flaud, R.R. Gamache, J.T. Hodges, D. Jacquemart, E.J. Mlawer, A.V. Nikitin, V.I. Perevalov, M. Rotger, J. Tennyson, G.C. Toon, H. Tran, V.G. Tyuterev, E.M. Adkins, A. Baker, A. Barbe, E. Canè, A.G. Császár, A. Dudaryonok, O. Egorov, A.J. Fleisher, H. Fleurbaey, A. Foltynowicz, T. Furtenbacher, J.J. Harrison, J.–M. Hartmann, V.–M. Horneman, X. Huang, T. Karman, J. Karns, S. Kassi, I. Kleiner, V. Kofman, F. Kwabia–Tchana, N.N. Lavrentieva, T.J. Lee, D.A. Long, A.A. Lukashevskaya, O.M. Lyulin, V.Yu. Makhnev, W. Matt, S.T. Massie, M. Melosso, S.N. Mikhailenko, D. Mondelain, H.S.P. Müller, O.V. Naumenko, A. Perrin, O.L. Polyansky, E. Raddaoui, P.L. Raston, Z.D. Reed, M. Rey, C. Richard, R. Tóbiás, I. Sadiek, D.W. Schwenke, E. Starikova, K. Sung, F. Tamassia, S.A. Tashkun, J. Vander Auwera, I.A. Vasilenko, A.A. Vigasin, G.L. Villanueva, B. Vispoel, G. Wagner, A. Yachmenev, and S.N. Yurchenko. The HITRAN2020 molecular spectroscopic database. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 277, 2022.

Victor Gorshelev, Anna Serdyuchenko, Mark Weber, W. Chehade, and John P. Burrows. High spectral resolution ozone absorption cross-sections – Part 1: Measurements, data analysis and comparison with previous measurements around 293 K. *Atmospheric Measurement Techniques*, 7(2):609–624, 2014.

Eloi Grau, Jean-Philippe Gastellu-Etchegorry, Ferran Gascon, Jeremy Rubio, and Aurore Brut. Earth-atmosphere radiative transfer in DART model. In *WHISPERS*, pages 1–4, 2009.

Adrien Gruson, Mickaël Ribardière, Martin Šik, Jiří Vorba, Rémi Cozot, Kadi Bouatouch, and Jaroslav Křivánek. A spatial target function for metropolis photon tracing. *ACM Transactions on Graphics*, 36(4), 2016.

Christian A. Gueymard. The SMARTS spectral irradiance model after 25 years: New developments and validation of reference spectra. *Solar Energy*, 187:233–253, 2019.

David Guimera, Diego Gutierrez, and Adrián Jarabo. A Physically-Based Spatio-Temporal Sky Model. In *Spanish Computer Graphics Conference*, 2018.

Jörg Haber, Marcus Magnor, and Hans-Peter Seidel. Physically-based simulation of twilight phenomena. *ACM Transactions on Graphics*, 24:1353–1373, 2005.

Toshiya Hachisuka, Jacopo Pantaleoni, and Henrik Wann Jensen. A path space extension for robust light transport simulation. *ACM Transactions on Graphics*, 31(6):191:1–191:10, 2012.

Toshiya Hachisuka, Anton S. Kaplanyan, and Carsten Dachsbacher. Multiplexed metropolis light transport. *ACM Transactions on Graphics*, 33(4): 100:1–100:10, 2014.

Johannes Hanika, Andrea Weidlich, and Marc Droske. Once-more scattered next event estimation for volume rendering. *Computer Graphics Forum*, 41(4), 2022.

Miles Hansard. Fast synthesis of atmospheric image effects. In *European Conference on Visual Media Production*, 2019.

David Hart, Philip Dutré, and Donald P. Greenberg. Direct illumination with lazy visibility evaluation. In *Proceedings of SIGGRAPH 1999*, pages 147–154, 1999.

Vlastimil Havran and Mateu Sbert. Optimal combination of techniques in multiple importance sampling. In *Proceedings of the 13th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, pages 141–150, 2014.

Hera Y. He and Art B. Owen. Optimal mixture weights in multiple importance sampling. arXiv:1411.3954, 2014.

L. G. Henyey and J. L. Greenstein. Diffuse radiation in the Galaxy. *Astrophysical Journal*, 93:70–83, 1941.

Sebastian Herholz and Addis Dittebrandt. Intel® Open Path Guiding Library, 2022. `http://www.openpgl.org`.

Sebastian Herholz, Oskar Elek, Jiří Vorba, Hendrik Lensch, and Jaroslav Křivánek. Product importance sampling for light transport path guiding. *Computer Graphics Forum*, 35(4):67–77, 2016.

Michael Hess, Peter Koepke, and I. Schult. Optical properties of aerosols and clouds: The software package OPAC. *Bulletin of the American Meteorological Society*, 79(5):831–844, 1998.

Heinrich Hey and Werner Purgathofer. Importance sampling with hemispherical particle footprints. In *Proceedings of the 18th Spring Conference on Computer Graphics*, page 107, 2002.

Sébastien Hillaire. A scalable and production ready sky and atmosphere rendering technique. *Computer Graphics Forum*, 39(4):13–22, 2020.

Yannick Hold-Geoffroy, Akshaya Athawale, and Jean-François Lalonde. Deep sky modeling for single image outdoor lighting estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6927–6935, 2019.

Helmuth Horvath. On the applicability of the Koschmieder visibility formula. *Atmospheric Environment (1967)*, 5(3):177–184, 1971.

Lukáš Hošek. Atmospheric Rendering. *Dissertation at the Department of Software and Computer Science Education of Charles University*, 2019.

Lukáš Hošek and Alexander Wilkie. An analytic model for full spectral sky-dome radiance. *ACM Transactions on Graphics*, 31(4):95, 2012.

Lukáš Hošek and Alexander Wilkie. Adding a solar radiance function to the Hošek-Wilkie skylight model. *IEEE Computer Graphics and Applications*, 33 (3):44–52, 2013.

E. O. Hulburt. Explanation of the brightness and color of the sky, particularly the twilight sky. *Journal of the Optical Society of America*, 43(2):113–118, 1953.

Henrik Wann Jensen. Importance driven path tracing using the photon map. *Rendering Techniques*, 95:326–335, 1995.

Henrik Wann Jensen and Niels Jørgen Christensen. Efficiently rendering shadows using the photon map. In *Proceedings of Compugraphics 1995*, pages 285–291, 1995.

Malvin H. Kalos and Paula A. Whitlock. *Monte Carlo Methods*. Wiley-VCH, 2nd edition, 2008.

Csaba Kelemen, László Szirmay-Kalos, György Antal, and Ferenc Csonka. A simple and robust mutation strategy for the metropolis light transport algorithm. *Computer Graphics Forum*, 21(3):531–540, 2002.

Alexander Keller, Luca Fascione, Marcos Fajardo, Iliyan Georgiev, Per Christensen, Johannes Hanika, Christian Eisenacher, and Greg Nichols. The path tracing revolution in the movie industry. In *ACM SIGGRAPH 2015 Courses*, 2015.

Joseph T. Kider, Jr., Daniel Knowlton, Jeremy Newlin, Yining Karl Li, and Donald P. Greenberg. A framework for the experimental comparison of solar and skydome illumination. *ACM Transactions on Graphics*, 33(6):180:1–180:12, 2014.

Tamara Kolda and Brett Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

Ivo Kondapaneni, Petr Vévoda, Pascal Grittmann, Tomáš Skřivan, Philipp Slusallek, and Jaroslav Křivánek. Optimal multiple importance sampling. *ACM Transactions on Graphics*, 38(4), 2019.

A. Kreuter and M. Blumthaler. Feasibility of polarized all-sky imaging for aerosol characterization. *Atmospheric Measurement Techniques*, 6(7):1845–1854, 2013.

Peter Kutz. Sky Renderer project blog. `http://skyrenderer.blogspot.com`, 2013. Accessed: 2015-12-31.

Jaroslav Křivánek, Iliyan Georgiev, Toshiya Hachisuka, Petr Vévoda, Martin Šik, Derek Nowrouzezahrai, and Wojciech Jarosz. Unifying points, beams, and paths in volumetric light transport simulation. *ACM Transactions on Graphics*, 33(4):103:1–103:13, 2014.

Eric P. Lafortune and Yves D. Willems. A 5d tree to reduce the variance of monte carlo ray tracing. *Rendering Techniques*, pages 11–20, 1995.

Yu-Chi Lai, Shao Hua Fan, Stephen Chenney, and Charcle Dyer. Photorealistic image rendering with population monte carlo energy redistribution. In *Proceedings of Eurographics Symposium on Rendering*, 2007.

Stephen S. Lavenberg, Thomas L. Moeller, and Peter D. Welch. Statistical Results on Control Variables with Application to Queueing Network Simulation. *Operations Research*, 30(1):182–202, 1982.

Marc Lebrun, Antoni Buades, and Jean-Michel Morel. A nonlocal bayesian image denoising algorithm. *SIAM Journal on Imaging Sciences*, 6(3):1665–1688, 2013.

Raymond L. Lee, Wolfgang Meyer, and Goetz Hoeppe. Atmospheric ozone and colors of the Antarctic twilight sky. *Applied Optics*, 50(28):162–171, 2011.

Heqi Lu, Romain Pacanowski, and Xavier Granier. Second-order approximation for variance reduction in multiple importance sampling. *Computer Graphics Forum*, 32(7):131–136, 2013.

Ricardo Marques, Christian Bouville, Mickaël Ribardiere, Luís Paulo Santos, and Kadi Bouatouch. A spherical gaussian framework for Bayesian Monte Carlo rendering of glossy surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 19(10):1619–1632, 2013.

Luca Martino, Víctor Elvira, David Luengo, and Jukka Corander. An adaptive population importance sampler: Learning from uncertainty. *IEEE Transactions on Signal Processing*, 63(16):4422–4437, 2015.

Bernhard Mayer. Radiative transfer in the cloudy atmosphere. *EPJ Web of Conferences*, 1:75–99, 2009.

Bailey Miller, Iliyan Georgiev, and Wojciech Jarosz. A null-scattering path integral formulation of light transport. *ACM Transactions on Graphics*, 38(4): 1–13, 2019.

Hans Mueller. The foundations of optics. In *Proceedings of the Winter Meeting of the Optical Society of America*, page 661, 1948.

Thomas Müller, Markus Gross, and Jan Novák. Practical Path Guiding for Efficient Light-Transport Simulation. *Eurographics Symposium on Rendering*, 36(4), 2017.

Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Transactions on Graphics*, 38(6), 2019.

Tomoyuki Nishita, Takao Sirai, Katsumi Tadamura, and Eihachiro Nakamae. Display of the earth taking into account atmospheric scattering. In *Proceedings of SIGGRAPH 1993*, pages 175–182, 1993.

Tomoyuki Nishita, Yoshinori Dobashi, and Eihachiro Nakamae. Display of clouds taking into account multiple anisotropic scattering and sky light. In *Proceedings of SIGGRAPH 1996*, pages 379–386, New York, NY, USA, 1996.

Sean O'Neil. Accurate atmospheric scattering. GPU Gems 2, 2005.

Art Owen and Yi Zhou. Safe and effective importance sampling. *Journal of the American Statistical Association*, 95(449):135–143, 2000.

Anthony Pajot, Loic Barthe, Mathias Paulin, and Pierre Poulin. Representativity for robust and adaptive multiple importance sampling. *IEEE Transactions on Visualization and Computer Graphics*, 17(8):1108–1121, 2011.

Jacopo Pantaleoni and Eric Heitz. Notes on optimal approximations for importance sampling. arXiv:1707.08358, 2017.

Eric Paquette, Pierre Poulin, and George Drettakis. A light hierarchy for fast rendering of scenes with many lights. *Computer Graphics Forum*, 17(3):63–74, 1998.

Vincent Pegoraro, Carson Brownlee, Peter S. Shirley, and Steven G. Parker. Towards interactive global illumination effects via sequential Monte Carlo adaptation. *IEEE/EG Symposium on Interactive Ray Tracing 2008*, pages 107–114, 2008.

Raul Perez, R. Seals, and J. Michalsky. All-weather model for sky luminance distribution–preliminary configuration and validation. *Solar Energy*, 50(3):235 – 245, 1993.

Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering, From Theory to Implementation.* Morgan Kaufmann Publishers Inc., 3rd edition, 2016.

Stefan Popov, Iliyan Georgiev, Philipp Slusallek, and Carsten Dachsbacher. Adaptive quantization visibility caching. *Computer Graphics Forum*, 32(2): 399–408, 2013.

Stefan Popov, Ravi Ramamoorthi, Fredo Durand, and George Drettakis. Probabilistic connections for bidirectional path tracing. *Computer Graphics Forum*, 34(4):75–86, 2015.

Arcot J. Preetham, Peter Shirley, and Brian Smits. A practical analytic model for daylight. In *Proceedings of SIGGRAPH 1999*, pages 91–100, 1999.

Carl Edward Rasmussen and Zoubin Ghahramani. Bayesian monte carlo. *Advances in Neural Information Processing Systems*, pages 489–496, 2002.

Fabrice Rousselle, Wojciech Jarosz, and Jan Novák. Image-space control variates for rendering. *ACM Transactions on Graphics*, 35(6), 2016.

Reuven Y. Rubinstein and Ruth Marcus. Efficiency of Multivariate Control Variates in Monte Carlo Simulation. *Operations Research*, 33(3):661–677, 1985.

Pinar Satilmis, Thomas Bashford-Rogers, Alan Chalmers, and Kurt Debattista. A machine-learning-driven sky model. *IEEE Computer Graphics and Applications*, 37(1):80–91, 2016.

Mateu Sbert and Vlastimil Havran. Adaptive multiple importance sampling for general functions. *The Visual Computer*, 33(6-8):845–855, 2017.

Mateu Sbert, Vlastimil Havran, and Laszlo Szirmay-Kalos. Variance analysis of multi-sample and one-sample multiple importance sampling. *Computer Graphics Forum*, 35(7):451–460, 2016.

Mateu Sbert, Vlastimil Havran, and Laszlo Szirmay-Kalos. Multiple importance sampling revisited: breaking the bounds. *EURASIP Journal on Advances in Signal Processing*, 2018(1):15, 2018.

Peter Shirley, Changyaw Wang, and Kurt Zimmerman. Monte Carlo techniques for direct lighting calculations. *ACM Transactions on Graphics*, 15(1):1–36, 1996.

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

Hendrik Christoffel van de Hulst. *Light scattering by small particles*. Structure of matter series. Dover Publications, 1957.

Eric Veach. Robust Monte Carlo methods for light transport simulation. *Dissertation at the Department of Computer Science of Stanford University*, 1997.

Eric Veach and Leonidas J. Guibas. Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of SIGGRAPH 1995*, pages 419–428, 1995.

Sekhar Venkatraman and James R. Wilson. The efficiency of control variates in multiresponse simulation. *Operations Research Letters*, 5(1):37–42, 1986.

E.F. Vermote, D. Tanre, J.L. Deuze, M. Herman, and J.-J. Morcette. Second simulation of the satellite signal in the solar spectrum, 6S: an overview. *IEEE Transactions on Geoscience and Remote Sensing*, 35(3):675–686, 1997.

Petr Vévoda, Ivo Kondapaneni, and Jaroslav Křivánek. Bayesian online regression for adaptive direct illumination sampling. *ACM Transactions on Graphics*, 37(4):125:1–125:12, 2018.

Petr Vévoda, Thomas Bashford-Rogers, Monika Kolářová, and Alexander Wilkie. A wide spectral range sky radiance model. *Computer Graphics Forum*, 41(7): 291–298, 2022.

Jiří Vorba, Ondřej Karlík, Martin Šik, Tobias Ritschel, and Jaroslav Křivánek. On-line learning of parametric mixture models for light transport simulation. *ACM Transactions on Graphics*, 33(4):101:1–101:11, 2014.

Martin Šik, Hisanari Otsu, Toshiya Hachisuka, and Jaroslav Křivánek. Robust light transport simulation via metropolised bidirectional estimators. *ACM Transactions on Graphics*, 35(6):245:1–245:12, 2016.

Ingo Wald and Carsten Benthin. Interactive global illumination in complex and highly occluded environments. *Eurograhics Symposium on Rendering*, pages 1–9, 2003.

Bruce Walter, Sebastian Fernandez, Adam Arbree, Kavita Bala, Michael Donikian, and Donald P Greenberg. Lightcuts: a scalable approach to illumination. *ACM Transactions on Graphics*, 24(3):1098–1107, 2005.

Bruce Walter, Adam Arbree, Kavita Bala, and Donald P. Greenberg. Multidimensional lightcuts. *ACM Transactions on Graphics*, 25(3):1081, 2006.

Rui Wang and Oskar Akerlund. Bidirectional importance sampling for unstructured direct illumination. *Computer Graphics Forum*, 28(2):269–278, 2009.

Xin Wang, Jun Gao, Zhiguo Fan, and Nicholas W Roberts. An analytical model for the celestial distribution of polarized light, accounting for polarization singularities, wavelength and atmospheric turbidity. *Journal of Optics*, 18(6): 065601, 2016.

Gregory J. Ward. Adaptive shadow testing for ray tracing. In *Proceedings of the Second Eurographics Workshop on Rendering*, pages 11–20, 1994.

C. Wehrli. Extraterrestrial solar spectrum. In *Publication no. 615*. Physikalisch-Meteorologisches Observatorium + World Radiation Center, Davos Dorf, Switzerland, 1985.

Alexander Wilkie. The Advanced Rendering Toolkit, 2018. `http://cgg.mff.cuni.cz/ART`.

Alexander Wilkie and Andrea Weidlich. Polarised light in computer graphics. In *SIGGRAPH Asia 2012 Courses*, 2012.

Alexander Wilkie, Christiane Ulbricht, Robert F. Tobler, Georg Zotti, and Werner Purgathofer. An analytical model for skylight polarization. *Rendering Techniques*, pages 387–398, 2004.

Alexander Wilkie, Sehera Nawaz, Marc Droske, Andrea Weidlich, and Johannes Hanika. Hero wavelength spectral sampling. *Computer Graphics Forum*, 33: 123–131, 2014.

Alexander Wilkie, Petr Vévoda, Thomas Bashford-Rogers, Lukáš Hošek, Tomáš Iser, Monika Kolářová, Tobias Rittig, and Jaroslav Křivánek. A fitted radiance and attenuation model for realistic atmospheres. *ACM Transactions on Graphics*, 40(4), 2021.

Yu Ting Wu and Yung Yu Chuang. VisibilityCluster: Average directional visibility for many-light rendering. *IEEE Transactions on Visualization and Computer Graphics*, 19(9):1566–1578, 2013.

Jinsong Zhang, Kalyan Sunkavalli, Yannick Hold-Geoffroy, Sunil Hadap, Jonathan Eisenman, and Jean-François Lalonde. All-weather deep outdoor lighting estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10158–10166, 2019.

List of abbreviations

BRDF	bidirectional reflectance distribution function
CPD	canonical polyadic decomposition
CV	control variate
DI	direct illumination
GI	global illumination
HG	Henyey-Greenstein
INSO	water-insoluble particles
MAP	maximum a posteriori
MC	Monte Carlo
MIS	multiple importance sampling
ML	maximum likelihood
MSE	mean squared error
OPAC	optical properties of aerosols and clouds (name of a database)
PDF	probability density function
RMSE	root mean squared error
SOOT	black carbon particles
SNR	signal-to-noise ratio
SVD	singular value decomposition
SWIR	short-wavelength infrared
WASO	water-soluble particles

List of publications

This thesis is based on the following publications:

1. **Petr Vévoda**, Ivo Kondapaneni, and Jaroslav Křivánek. Bayesian online regression for adaptive direct illumination sampling. *ACM Transactions on Graphics*, 37(4):125:1–125:12, 2018.

    - The author shares the first authorship with Ivo Kondapaneni.
    - The author created an initial version of the complete method (including scalability, on-line learning, distance falloff modelling, control variates, prior design) and then collaborated with Ivo Kondapaneni on the Bayessian formulation. The author was also responsible for all the implementation work.

2. Ivo Kondapaneni, **Petr Vévoda**, Pascal Grittmann, Tomáš Skřivan, Philipp Slusallek, and Jaroslav Křivánek. Optimal multiple importance sampling. *ACM Transactions on Graphics*, 38(4), 2019.

    - The author shares the first authorship with Ivo Kondapaneni.
    - The author discovered the limitation of the balance heuristic variance bounds and designed all the applications of the optimal weights including the new sampling techniques. The author was also responsible for most of the implementation work.

3. Alexander Wilkie, **Petr Vévoda**, Thomas Bashford-Rogers, Lukáš Hošek, Tomáš Iser, Monika Kolářová, Tobias Rittig, and Jaroslav Křivánek. A fitted radiance and attenuation model for realistic atmospheres. *ACM Transactions on Graphics*, 40(4), 2021.

    - The author shares the first authorship with Alexander Wilkie.
    - The author contributed to the atmosphere composition specification (switch to OPAC profiles, their smoothing and interpolation), reference dataset rendering (`atmo_sim` optimizations, parameter values selection optimization, rendering execution and dataset assembly), fitting (fitting algorithm improvements, rendering execution and dataset assembly, dataset compression), and implementation (Corona and Standalone).

4. **Petr Vévoda**, Thomas Bashford-Rogers, Monika Kolářová, and Alexander Wilkie. A wide spectral range sky radiance model. *Computer Graphics Forum*, 41(7):291–298, 2022.

    - The author was the primary investigator.