Importance Sampling of Realistic Light Sources

Heqi Lu

THÈSE

présentée à

L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

par **LU Heqi**

Pour obtenir le grade de

DOCTEUR

SPÉCIALITÉ : INFORMATIQUE

Echantillonnage d'importance des sources de lumières réalistes

---

Soutenue le : 27/02/2014

Après avis des rapporteurs :

Prof. PAULIN Mathias	Université Paul Sabatier
Dr. HOLZSCHUCH Nicolas	Inria Grenoble-Rhne-Alpes

Devant la commission d'examen composée de :

Prof. SCHLICK Christophe	Université De Bordeaux ..	Prèsident
Prof. PAULIN Mathias	Université Paul Sabatier ..	Rapporteur
Dr. HOLZSCHUCH Nicolas	Inria Grenoble-Rhne-Alpes	Rapporteur
Prof. PENG Qunsheng	Zhejiang University	Examinateur
Prof. GRANIER Xavier	Institut d'Optique	Directeur de Thèse
Dr. PACANOWSKI Romain	CNRS	Co-Directeur de Thèse

– 2014 –

Importance Sampling of Realistic Light Sources

A thesis submitted for the degree of

Doctor of Philosophy

in:

Computer Science

at:

Computer Department

Bordeaux University

by:

Heqi Lu

supervised by:

Prof. Xavier Granier

Dr. Romain Pacanowski

Feb. 27, 2014

I would like to dedicate this thesis to my loving family and many friends. A special feeling of gratitude to my loving parents whose words of encouragement and push for tenacity ring in my ears. My wife and my son who build a warm home that is an undeniably bedrock. My friends who supported me throughout the process. I appreciate all they have done.

Acknowledgements

First and foremost, I would like to express the deepest appreciation to Professor Xavier Granier, my supervisor for his guidance, understanding, patience, and more important, his friendship and for all I have learned from him as well as for his continuous help and support not only on this thesis, but also in daily life. I would also like to thank him for being an open person to different ideas, and for encouraging and helping me to shape my interest.

Besides, I would like to gratefully thank my co-superior, Romain Pacanowski whose advices and insight were invaluable to me. For all I learned from him, and for providing the help in some stages of this thesis.

In addition, I would also like to thank all my colleges, former colleges, secretaries and former secretaries for their input, valuable discussions, accessibility, and especially for constructing a very friendly work environment.

Furthermore, I would like to thank all the reviewers for my thesis and all the jury for suggestions and questions. Also, thanks to the jury for coming to my PhD defense from different areas to Bordeaux. In particular, I would like to thank Professor Qunshen Peng who came from China for participating the defense specially.

Moreover, I would like to thank université de bordeaux, Inria, LaBRI and all the fundings that supported me during this PhD. My scholarship is funded by the Région Aquitaine and the research was supported by the ALTA project. Without them, this thesis cannot be finished.

Finally, I would like to sincerely thank my wife Caixia Chen. Her support, encouragement, quiet patience and unwavering love were undeniably the bedrock upon which the past four years. Her tolerance of my occasional vulgar moods is a testament in itself of her unyielding devotion and love. I thank my son who brings me happiness, hope and sense of responsibility. Moreover, I also thank my parents who silently support me thousands of miles away.

Abstract

Realistic images can be rendered by simulating light transport with Monte Carlo techniques. The possibility to use realistic light sources for synthesizing images greatly contributes to their physical realism. Among existing models, the ones based on environment maps and light fields are attractive due to their ability to capture faithfully the far-field and near-field effects as well as their possibility of being acquired directly. Since acquired light sources have arbitrary frequencies and possibly high dimension (4D), using such light sources for realistic rendering leads to performance problems.

In this thesis, we focus on how to balance the accuracy of the representation and the efficiency of the simulation. Our work relies on generating high quality samples from the input light sources for unbiased Monte Carlo estimation. In this thesis, we introduce three novel methods.

The first one is to generate high quality samples efficiently from dynamic environment maps that are changing over time. We achieve this by introducing a GPU approach that generates light samples according to an approximation of the form factor and combines the samples from BRDF sampling for each pixel of a frame. Our method is accurate and efficient. Indeed, with only 256 samples per pixel, we achieve high quality results in real time at 1024×768 resolution. The second one is an adaptive sampling strategy for light field light sources (4D), we generate high quality samples efficiently by restricting conservatively the sampling area without reducing accuracy. With a GPU implementation and without any visibility computations, we achieve high quality results with 200 samples per pixel in real time at 1024×768 resolution. The performance is still interactive as long as the visibility is computed using our shadow map technique. We also provide a fully unbiased approach by replacing the visibility test with a offline CPU approach. Since light-based importance sampling is not very effective when the underlying material of the geometry is specular, we introduce a new balancing technique for Multiple Importance Sampling. This allows us to combine other sampling techniques with our light-based importance sampling. By minimizing the variance based on a second-order approximation, we are able to find good balancing between different sampling techniques without any prior knowledge. Our method is effective, since we actually reduce in average the variance for all of our test scenes with different light sources, visibility complexity, and materials. Our method is also efficient, by the fact that the overhead of our "black-box" approach is constant and represents 1% of the whole rendering process.

Résumé

On peut atteindre des images réalistes par la simulation du transport lumineuse avec des méthodes de Monte-Carlo. La possibilité d'utiliser des sources de lumière réalistes pour synthétiser les images contribue grandement à leur réalisme physique. Parmi les modèles existants, ceux basés sur des cartes d'environnement ou des champs lumineuse sont attrayants en raison de leur capacité à capter fidèlement les effets de champs lointain et de champs proche, aussi bien que leur possibilité d'être acquis directement. Parce que ces sources lumineuses acquises ont des fréquences arbitraires et sont éventuellement de grande dimension (4D), leur utilisation pour un rendu réaliste conduit à des problèmes de performance.

Dans ce manuscrit, je me concentre sur la façon d'équilibrer la précision de la représentation et de l'efficacité de la simulation. Mon travail repose sur la génération des échantillons de haute qualité à partir des sources de lumière par des estimateurs de Monte-Carlo non-biaisés. Dans ce manuscrit, nous présentons trois nouvelles méthodes.

La première consiste à générer des échantillons de haute qualité de manière efficace à partir de cartes d'environnement dynamiques (i.e. qui changent au cours du temps). Nous y parvenons en adoptant une approche GPU qui génère des échantillons de lumière grâce à une approximation du facteur de forme et qui combine ces échantillons avec ceux issus de la BRDF pour chaque pixel d'une image. Notre méthode est précise et efficace. En effet, avec seulement 256 échantillons par pixel, nous obtenons des résultats de haute qualité en temps réel pour une résolution de 1024×768. La seconde est une stratégie d'échantillonnage adaptatif pour des sources représente comme un "light field". Nous générons des échantillons de haute qualité de manière efficace en limitant de manière conservative la zone d'échantillonnage sans réduire la précision. Avec une mise en œuvre sur GPU et sans aucun calcul de visibilité, nous obtenons des résultats de haute qualité avec 200 échantillons pour chaque pixel, en temps réel et pour une résolution de 1024×768. Le rendu est encore être interactif, tant que la visibilité est calculée en utilisant notre nouvelle technique de carte d'ombre (shadow map). Nous proposons également une approche totalement non-biaisée en remplaçant le test de visibilité avec une approche CPU. Parce que l'échantillonnage d'importance à base de lumière n'est pas très efficace lorsque le matériau sous-jacent de la géométrie est spéculaire, nous introduisons une nouvelle technique d'équilibrage pour de l'échantillonnage multiple (Multiple Importance Sampling). Cela nous permet de combiner d'autres techniques d'échantillonnage avec le notre basé sur la lumière. En minimisant la variance selon une approximation de second ordre, nous sommes en mesure de trouver une bonne représentation entre les différentes techniques d'échantillonnage sans aucune connaissance préalable. Notre méthode est pertinance, puisque nous réduisons effectivement en moyenne la variance pour toutes nos scènes de test avec différentes sources de lumière, complexités de visibilité et de matériaux. Notre méthode est aussi efficace par le fait que le surcoût de notre approche «boîte noire» est constant et représente 1% du processus de rendu dans son ensemble.

Brief Introduction

Realistic rendering techniques try to reproduce realistic images for the virtual world. Such techniques are useful for movie, CAD, simulation and any other requirements of realism. One possible technique is to simulate the light transport directly. Since light sources contribute to the illumination significantly, using realistic light sources for rendering will definitively increase the realism. It would be very interesting if we can use real world light sources to illuminate virtual world objects directly in real time. In this way, we would be able to build an interaction between real world and virtual world. Therefore, all stages must be done in real time. However, the real-time simulation of realistic illumination has been one of the main challenges in Computer Graphics because of its high complexity and expensive computation cost.

Realistic illumination is complex and has arbitrary variations. In order to achieve realistic rendering, the use of realistic geometry, realistic materials and realistic light sources with physically-based simulation are required. These realistic elements have arbitrary complexity. A straightforward way to represent them is to capture them from the real world directly. However, it is very difficult to use them for rendering since their inherent complexity. Although, there are many methods to achieve plausible realistic rendering results, they are not reliable enough for some applications that require high accuracy such as simulation and CAD. Reliable results are yet necessary to compute the ground truth, and are very helpful to verify the approximations made by plausible rendering techniques. Therefore, using elements as realistic as possible to achieve reliable rendering results is quite natural. In this thesis, we focus on using realistic light sources.

Due to the complexity, it is very hard to represent captured realistic light sources with classical light models. This problem makes the captured data very hard to manage and very hard to compute. In order to use captured light sources, previously, researchers have done a lot of effort to reduce the computations cost with approximations. They use computational friendly basis functions to represent captured light data and because of the approximated representation, some details lost. As a result, at the rendering stage, the performance is increased. However, as a drawback, it is very hard to represent high frequency with these basis functions. One solution is to perform pre-computation to represent data with better basis functions that have the capability of representing all frequencies. However, this pre-computation remains very costly and the projection of the data to the basis functions is difficult to achieve. Therefore, achieving all stages in real time is very difficult. The motivation of this thesis is that even the pre-computation should be done in real time.

The simulation of light transport is computationally expensive. Rays are emitted

from light sources, then scattered between different geometries with different materials and possibly many times to finally reach an observer or a sensorcamera. In order to generate realistic images, we have to compute the energy that reaches at each pixel of the camera or the sensor, by gathering rays coming from the light sources. Since, there is generally no analytical solutions to compute this, researchers have focused on numerical methods (e.g., finite element methods and Monte Carlo methods) to estimate the energy reaching at each pixel.

Monte Carlo methods have been very famous for 30 years, since they are independent from the complexity of the scene geometry. This feature is very important for estimating realistic illumination, since light propagation may introduce high dimension problems as long as the rays scatter between different surfaces many times. The key point for using Monte Carlo Methods is to use good samples for the estimation. The probability of generating good samples should be correlated to the energy that reaches at a pixel. If they are proportional, only one sample is enough to get a converged result for a pixel. Obviously, the more samples needed for estimation, the more costly of the computation will be. Therefore, the key point to achieve real-time realistic rendering with realistic light sources is to use preferably only good samples for estimation. However, it is very difficult to find good samples, since we do not know the exact relationship between the light coming from light sources and the energy that reaches at the sensor (or the observer). If we knew, the energy could be computed with an analytical solution directly. Sometimes, we may only approximate the relationship between them, and set the probability of generating samples as coherent as possible to the final energy that reaches the sensor (or the observer). However, how to approximate efficiently this relationship are still open questions. The problem is even bigger using complex realistic light sources.

Therefore, our motivation is to find good samples and generate these samples efficiently for realistic rendering with Monte Carlo estimation. In order to do all stages in real time automatically, the generation of the samples and the estimation of the illumination should be both in real time. In this thesis, we focus on how to balance the accuracy of the representation and the efficiency of the simulation. This work relies on generating high quality samples from the input light sources for unbiased Monte Carlo estimation. We introduce new sampling techniques for physically-based rendering with time-varying environment lighting (2D data) and light field light sources (4D data). The results show that physically accurate rendering with realistic light sources can be achieved in real time.

The first contribution is a new method that can achieve real-time physically-based rendering with far-field lighting that changes over-time. With this method, if we can capture the far-field lighting with some dedicated devices in real time, all stages of the realistic illumination can be done in real time and thus the interaction between the two world could

be built. The main problem of using time-varying environment lighting is that we have to generate light samples on the environment map at each frame. Therefore, the generation of the light samples must be in real time. In order to achieve this goal we use the representation of the environment lighting directly without any basis functions. The raw data on environment maps are represented by a constant box basis (pixel shape) which coefficients are the pixel color. However this raw representation has a large set of data, therefore it is not computationally efficient. In order to use this kind of light source efficiently, we have to try to use as small data as possible. Moreover, for the purpose of accuracy, we have to preserve the representation as much as possible. So the key solution is to find important data in real time. We achieve this by introducing a GPU approach that generates light samples according to an approximation of the product of the incoming light and cosine. Obviously, a good sample should result in high value of the product where incoming light and cosine are both high. However, it is very difficult to find these good samples, since the cosine factor and the incoming energy are independent. In general, there is no analytical solution for the product of these two. In order to approximate the product more accurately, we introduce a pseudo-product that approximates the cosine of each sample by the cosine of each cube map face that the sample belongs to. The solution is performed in two steps and assumes the environment lighting is stored in a cube map. For each frame and online. In the first step, we generate many samples on the current cube map according to the incoming light energy. In the second step, we select a sub set of the samples according to the weights of each cube map face. The weights are computed according to the products between the face energy and the face cosine for all faces. Furthermore, we also generate some samples according to the material properties. Therefore, the cosine, the material and the incoming light are all taken into account to generate the samples. Our new method is able to generate good samples in real time. Indeed, with only 256 samples per pixel, we achieve high quality results in real time at 1024×768 resolution.

Environment lighting is a far-field lighting that has directional variations (2D) only. In order to obtain more general realistic light sources for near-field lighting effects, we investigated the use of light field light sources (4D) for real-time rendering. **The second contribution** is an adaptive sampling strategy for these near-field lighting (4D). Previously, researchers have tried to use the light source directly under the concept of 4D reconstruction. They first generate samples in 2D domain for sample positions, and then generate light directions for each sample position. This 4D sampling technique is too costly to be real-time without introducing large approximations. In order to reduce the sampling cost, we introduced a conservative 2D sampling technique that can generate 4D samples. Intuitively, the users care more about the final image than the reconstruction of light sources. So instead of generating samples directly from the light sources, we introduce a method that

generates samples dynamically for each shaded pixels on the final image. It is achieved by restricting conservatively the sampling area without reducing accuracy. With a GPU implementation and without any visibility computations, high quality results are achieved with 200 samples per pixel in real time at 1024×768 resolution. This new method is real time and physically accurate without any approximations. The performance is still interactive as long as the visibility is computed using our shadow map technique. However, the visibility is approximated since we are using shadow maps. We also provide a fully unbiased approach by replacing the visibility test with a offline CPU approach. This offline approach does not introduce any approximations and thus can be used as a reference solution for future studies.

The third contribution is an adaptive multiple importance sampling (MIS) technique. As we all know, the realistic illumination is also impacted a lot by the geometry and its material. BRDF-based importance sampling is very effective when the material is very glossy or mirror whereas light-based importance sampling is very efficient when the light source has high frequency. For the purpose of generating good samples for different illumination, we have to combine different sampling techniques together. One technique is called MIS. MIS combines different sampling techniques with weighted sum. If we know the weights, we are able to use MIS for generating good samples. The weights should depend on the contribution of different sampling techniques. In order to compute this, we have to know the property of the sampling techniques and the characteristics of the scene. However, this knowledge is generally missing. Researchers prefer to treat different sampling techniques equally with heuristic strategies under the assumption that the different sampling techniques have the same contribution to the estimation in all samples (default balancing). Of course, this default balancing is not universal. Therefore, in this thesis, we introduce a new method that can adaptively compute a better balancing for each pixel. We achieve this by minimizing the variance using a second-order Taylor approximation with a few set of samples. This detection step acquires knowledge about the sampling and the scene and is able to estimate a better balancing for each pixel. Then we send more samples to estimate the radiance with a better balancing learned from the detection step. Indeed, we are able to find good balancing between different sampling techniques without any prior knowledge. An important advantage of this method is that we can estimate at the same time a better balancing and the radiance with the same set of samples. Therefore, the overhead of the detection of the balancing is very small and the computation cost is neglectable. The results show that the method is effective, since we reduce in average the variance for all of our test scenes with different light sources, visibility complexity, and materials. It is also efficient, since the overhead of our detection step is constant and represents only 1% of the whole rendering step.

Brève Introduction

Les techniques de restitution d'images réalistes essaient de reproduire des images réalistes pour le monde virtuel. Elles sont utiles dans l'industrie du cinéma, en CAO, en simulation ou de manière plus générale, dès que du réalisme est requis. Une façon de procéder est de simuler directement le transport lumineux. Comme les sources de lumière contribuent significativement au transport lumineux, utiliser des sources de lumière réalistes pour le rendu augmente nécessairement le réalisme de l'image restituée. Il serait intéressant de pouvoir utiliser en temps réel des sources de lumière du monde réel afin d'éclairer, en continu et de manière directe, des objets virtuels. Pour ce faire, toutes les étapes se doivent d'être effectuées en temps réel. Néanmoins, la simulation temps réel du transport lumineux en synthèse d'images reste un problème majeur et ce dû à sa complexité et son coût élevé de calcul.

L'éclairage réaliste est complexe et possède des variations arbitraires. Afin d'obtenir un rendu d'image réaliste, l'utilisation de géométrie, de matériaux réalistes et de sources de lumière réalistes ainsi qu'une simulation reposant sur les lois de la physique sont requis. Ces éléments réalistes possèdent une complexité arbitraire. Une manière directe de les représenter est de les mesurer à partir du monde réel. Cependant, il reste difficile de les utiliser pour le rendu dû à leur complexité intrinsèque. Bien qu'il existe de nombreuses méthodes pour obtenir un rendu plausible, elles restent insuffisantes pour les applications qui requièrent une grande précision telles en simulation ou en CAO. Des résultats fiables sont pourtant nécessaires afin d'obtenir une solution de référence à laquelle les approximations, faites pour un rendu plausible, peuvent être comparées. Dès lors, utiliser des éléments aussi réalistes que possibles semblent un choix naturel afin d'obtenir des rendus les plus réalistes. Dans cette thèse, nous nous intéressons à l'utilisation des sources de lumière réalistes.

A cause de la complexité, il est très difficile de représenter avec des modèles classiques les sources de lumière acquises. Ceci induit un problème complexe de gestion et de temps de calcul avec des données acquises. Afin d'utiliser des sources de lumière mesurées, les chercheurs ont consacré énormément d'effort pour réduire les temps de calculs en introduisant des approximations. Ils utilisent des fonctions de bases afin de représenter les sources de lumière qui ont l'avantage d'être efficace en terme de calcul, mais comme elles introduisent des approximations des détails sont perdus. Ainsi, à l'étape de rendu, les performances sont au rendez-vous mais il est difficile de représenter les hautes fréquences avec ce type d'approche. Une solution est d'effectuer un pré-calcul afin de représenter les données à l'aide de bases de fonctions plus efficaces qui ont la capacité de représenter toutes

les fréquences. Néanmoins, ce pré-calcul reste très coûteux et la projection des données dans la base de fonctions est difficile à réaliser. Ainsi, réaliser toutes les étapes en temps réel reste très difficile. La motivation de cette thèse est de faire en sorte que même l'étape de pré-calcul soit réalisée en temps réel.

La simulation du transport lumineux est coûteuse en termes de calculs. Les rayons sont émis depuis les sources de lumière, réfléchis une ou plusieurs fois entre les différentes géométries, qui possèdent elles-mêmes différents matériaux, avant de finalement atteindre un observateur ou un capteur (par exemple un CCD). Afin de générer des images réalistes, nous devons calculer l'énergie qui atteint un pixel du capteur en accumulant les rayons en provenance des sources de lumière. Comme il n'existe pas en général de solutions analytiques à ce calcul, les chercheurs se sont concentrés sur l'utilisation de méthodes numériques (e.g., éléments finis ou Monte Carlo) afin d'estimer l'énergie qui arrive au pixel.

Les méthodes de type Monte Carlo ont été populaires durant les trente dernières années parce qu'elles sont indépendantes de la géométrie de la scène. Cette propriété est très importante afin d'estimer un éclairement réaliste puisque la propagation de la lumière peut introduire des problèmes de haute dimension au fur et à mesure que les rayons se réfléchissent entre les différentes surfaces. L'idée centrale lors de l'utilisation des méthodes de Monte Carlo est d'utiliser des échantillons de bonne qualité pour l'estimation. La probabilité de générer de tels échantillons doit être corrélée à l'énergie qui atteint un pixel. Si elles sont proportionnelles, un seul échantillon est suffisant pour obtenir un résultat convergé pour un pixel donné. Evidemment, plus un grand nombre d'échantillons sont requis pour l'estimation, plus les calculs seront coûteux. Ainsi, il est nécessaire pour un rendu temps réel avec des sources lumières réalistes, d'utiliser au maximum des échantillons de bonne qualité pour l'estimation. En revanche, il est très difficile d'obtenir des échantillons de bonne qualité puisque nous ne connaissons pas exactement la relation entre la lumière, provenant des sources de lumière, et l'énergie qui atteint le capteur (ou l'observateur). Si nous en avions la connaissance, l'énergie pourrait être calculée directement avec une solution analytique. Parfois, nous ne pouvons qu'approximer la relation entre ces dernières, et assigner une probabilité, de générer des échantillons de manière aussi cohérente que possible, à l'énergie que recevra un pixel donné. Comment approximer cette relation de manière efficace reste une question ouverte. Le problème est même encore plus grand lors de l'utilisation de sources de lumière réalistes et complexes.

Ainsi, notre motivation est de trouver des échantillons de bonne qualité et les générer de manière efficace à l'aide de méthodes de type Monte Carlo afin d'obtenir un rendu réaliste. Afin d'effectuer automatiquement toutes les étapes en temps réel, la génération des échantillons et l'estimation de l'éclairement doivent toutes deux être accomplies en temps réel.

Dans cette thèse, nous nous concentrons sur comment pondérer la précision de la représentation avec l'efficacité de la simulation. Ce travail repose sur la génération d'échantillons de haute qualité pour les sources de lumière afin d'obtenir une estimation de type Monte Carlo non biaisée. Nous introduisons des nouvelles techniques d'échantillonnage pour un rendu physiquement réaliste avec des cartes d'environnement (données 2D) qui varient temporellement ainsi que des sources de type light-field (données 4D). Les résultats montrent que l'on peut obtenir avec des sources de lumière réalistes un rendu physiquement réaliste et ce en temps réel.

La première contribution est une nouvelle méthode qui permet d'obtenir un rendu physiquement réaliste et temps réel avec de l'éclairement en champ lointain variant avec le temps. Avec cette méthode, si l'on peut acquérir en temps réel l'éclairement lointain avec un capteur dédié, toutes les étapes de l'éclairement réaliste peuvent être accomplies en temps réel et alors l'interaction entre les deux mondes (réel et virtuel) sera construite. Le problème principal de l'utilisation d'éclairement variant temporellement est que nous devons générer des échantillons pour les sources de lumière à image rendue. Ainsi, cette génération doit elle aussi être effectuée en temps réel. Afin d'atteindre ce but nous utilisons directement la représentation de l'éclairement sans utiliser des bases de fonctions. Les données brutes des cartes d'environnement sont représentées par des bases de fonctions de type boite dont les coefficients sont les couleurs du pixel. Cependant, la représentation brute possède un grand nombre de données la rendant inefficace en terme de calculs. Afin d'utiliser ce type de sources de lumière aussi efficacement que possible, nous devons utiliser le moins de données possible. De plus, afin de garder de la précision, nous devons préserver la représentation autant que possible.

La solution recherchée est donc de trouver les données les plus importantes en temps réel. Nous réalisons cela en introduisant une approche GPU qui génère les échantillons des sources de lumière en fonction d'une approximation du produit de l'éclairement incident et du facteur cosinus. Naturellement, un «bon»échantillon doit générer une forte valeur du produit pour lequel l'éclairement et le facteur cosinus sont tous deux élevés. Cependant, il est très difficile de trouver ces bons échantillons puisque l'énergie incidente et le facteur cosinus sont indépendants. En général, il n'y existe pas de solution analytique pour le produit de ces deux quantités. Afin d'approximer le produit, nous introduisons un pseudo-produit qui représente le facteur cosinus de chaque échantillon par le cosinus de chaque face du cube visible depuis l'échantillon.

Lors de la première étape, nous générons un grand nombre d'échantillons sur la carte d'environnement cubique (*cube map*) en fonction de l'énergie de cette dernière. Lors de la second étape, nous sélectionnons un sous-ensemble de ces échantillons en fonction du poids de chaque face de la *cube map*. Ces poids sont calculés pour toutes les faces en

fonction du pseudo-produit entre l'énergie de la face le facteur cosinus de la face. De plus, nous générons aussi des échantillons en fonction du matériau. Ainsi, le facteur cosinus, le matériau et la lumière incidente sont tous pris en compte afin de générer les échantillons. Notre nouvelle méthode est capable de générer des échantillons de bonne qualité en temps réel. En effet, avec seulement 256 échantillons par pixel, nous obtenons des résultats temps réel de haute qualité pour des images de 1024 par 768 pixels.

L'éclairement à l'aide de carte d'environnement est un éclairement en champ lointain qui ne représente seulement que des variations directionnelles (2D). Afin d'obtenir des sources lumières plus réalistes pour l'éclairement en champ proche, nous nous intéressons à l'utilisation temps réel des sources de lumière de type *light-fields* (4D). **La seconde contribution** est une stratégie d'échantillonnage préférentiel pour ce type de sources 4D. Précédemment, les chercheurs ont essayé d'utiliser de manière directe ces sources de lumière en utilisant le concept de reconstruction 4D. Ils génèrent d'abord des échantillons dans le domaine 2D afin d'obtenir les positions spatiales de chaque échantillon, puis ils génèrent les directions pour chacune de ces positions. Cet échantillonnage 4D est trop coûteux pour être réalisé en temps réel sans introduire de fortes approximations. Afin de réduire le coût d'échantillonnage, nous introduisons une technique conservative d'échantillonnage 2D qui génère des échantillons 4D. Intuitivement, les utilisateurs attachent plus d'importance à l'image finale qu'à la reconstruction des sources de lumière. Ainsi, au lieu de générer des échantillons directement depuis les sources de lumière, nous introduisons une méthode qui génère dynamiquement les échantillons pour chaque pixel de l'image finale. Ceci est accompli en restreignant, de manière conservative, la zone d'échantillonnage et ce sans réduction de la précision. A l'aide d'une implémentation GPU et en omettant les calculs de visibilité, des résultats de haute qualité sont obtenus en temps réel avec 200 échantillons par pixel pour une image de 1024 par 768 pixels. Cette nouvelle méthode est temps réel et physiquement correcte sans approximation. Tant que les calculs de visibilité sont effectués avec notre technique de carte d'ombre (*shadow map*), les performances demeurent interactives. Cependant, l'utilisation de *shadow map* introduit une approximation dans le calcul de visibilité. Nous fournissons aussi une approche sans biais en remplaçant le calcul de visibilité par shadow map en utilisant le tracé de rayon. Cette approche CPU n'introduit aucune approximation et peut donc être utilisée comme solution de référence pour les études futures.

La troisième contribution est une technique d'échantillonnage préférentiel multiple (*Multiple ImportanceSampling*). Comme nous le savons tous, l'éclairement réaliste est aussi fortement impacté par la géométrie son matériau sous-jacent. L'échantillonnage préférentiel reposant sur les matériaux (BRDF) est très efficace lorsque le matériau est très brillant ou miroir alors que l'échantillonnage préférentiel reposant sur les sources de lumière est

quant à lui très efficace quand ces dernières possèdent de hautes fréquences. Pour générer des échantillons de bonne qualité avec différents types d'éclairement, nous devons combiner différentes techniques d'échantillonnage préférentiel. Une façon de procéder est appelé M.I.S. Le M.I.S. combine différente technique à l'aide d'une somme pondérée. Si nous connaissons cette pondération nous pouvons utiliser le MIS pour générer des échantillons de bonne qualité. La pondération doit dépendre de la contribution de chacune des techniques d'échantillonnage. Afin de computer cela, nous devons connaître les propriétés des techniques d'échantillonnage ainsi que les caractéristiques de la scène, information généralement manquante. Les chercheurs préfèrent traiter les différentes techniques d'échantillonnage de manière égales en faisant l'hypothèse que les différentes techniques auront la même contribution pour l'estimation de tous les échantillons (pondération par défaut: *default balancing*). Naturellement cette pondération par défaut n'est pas universelle. Ainsi, dans cette thèse, nous introduisons une nouvelle méthode qui calcule de manière adaptative une meilleure pondération pour chaque pixel. Nous accomplissons cela par une minimisation de la variance en effectuant une approximation de Taylor au second degré et en utilisant un faible nombre d'échantillons. Cette étape de détection accumule de l'information sur l'échantillonnage et la scène et est capable d'estimer une meilleure pondération d'échantillonnage pour chaque pixel. Ensuite, nous raffinons le résultat en envoyant plus d'échantillons, afin d'estimer la radiance, et en utilisant la pondération de l'étape précédente. En effet, nous sommes capable de trouver une bonne pondération des techniques d'échantillonnage sans aucune connaissance à priori. Un avantage majeur de la méthode introduite est que nous pouvons estimer, en même temps, une meilleure pondération ainsi que la radiance finale avec le même ensemble d'échantillons. Ainsi, le surcoût associé à l'étape de détection de la pondération est très faible. Les résultats montrent que la méthode est efficace puisque nous réduisons en moyenne la variance pour toutes nos scènes de tests avec des sources de lumière, de la visibilité et des matériaux très variés. La méthode est aussi efficace puisque le surcout induit par l'étape de détection est constant et représente seulement 1% du temps total de rendu.

CONTENTS

Chapter 1

Introduction and Motivation

Figure 1.1: *Realistic rendering with measured light sources.*

Nowadays, realistic rendering has already a big impact on a wide range of applications, such as simulation, movies, and video games. It uses digital models to perform computations and rendering 2D frames in virtual world (e.g., computer), for the sake of representing realistic illumination that we perceive from real world. However, there is a big gap between virtual world and real world, the former usually uses digital models and the later always follows physical laws.

In order to narrow this gap, researchers have done a lot of work to find reasonable digital models that match physical laws. Due to the increase of parallel computing power, research trends are more and more moving to physically-based rendering techniques [Pharr & Humphreys 2004; Ritschel *et al.* 2012], such as natural parallelization of algorithms: path tracing [Kajiya 1986; Lafortune & Willems 1993; Veach & Guibas 1997] and photon mapping [Hachisuka *et al.* 2008; Jensen 1996a; Kaplanyan & Dachsbacher 2013a]. The most important things for these light transport and simulation techniques are the roots from where the light comes: light sources. To achieve realistic rendering, light sources should be as realistic as possible. However, most light sources used for real-time applications are represented by empirical models, which are limited to perform simple and very approximated lighting. These limitations introduce drawbacks. Firstly, artists have to do a lot of work to design plausible realistic light sources with these empirical models. Secondly, researchers have to do a lot of research on how to avoid artifacts and how to achieve visually plausible

results. Thirdly, programmers have to do a lot of work to develop and optimize different techniques to achieve good performance for different levels of visual quality.

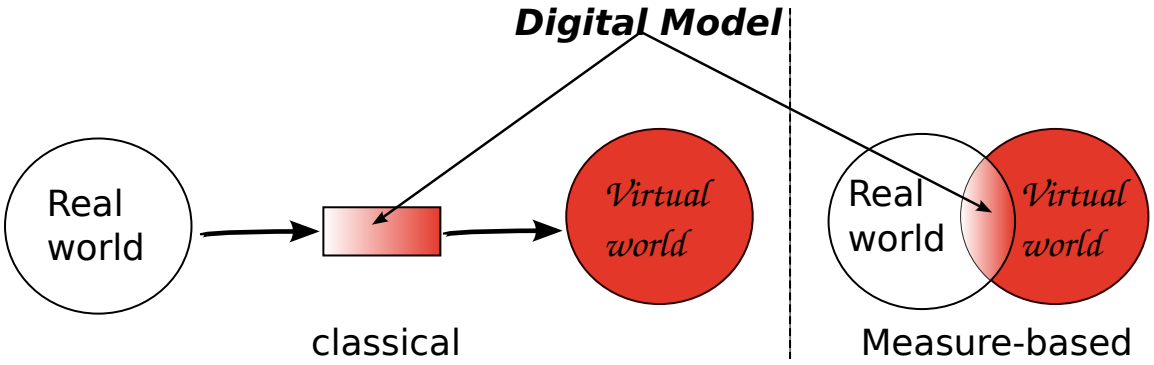


Figure 1.2: *Two different digital models built between real world and virtual world. Classical digital models are empirical models which are constructed by learning the laws of real world as well as matching the properties of virtual world such as computation and storage friendly. The digital model, the gradual changing rectangle has no union parts of the two worlds, since it is constructed by the knowledge and experience from the two worlds. On the contrary, measure-based digital models are built on the union part of the two worlds. These models are built from the real world data which is captured without knowing the physical law behind (i.e. How the data was created), and they may not computation nor storage friendly. However, these models actually reduce the gap since they have equivalent representations in the two worlds.*

The reason why these empirical models cause these problems, is because that the gap between real world and virtual world is too large. The empirical models based on empirical observations are not physically accurate ([Verbeck & Greenberg 1984]). As point by previous work [Ashdown & Rykowski 1997; Goesele *et al.* 2003; Stumpfel *et al.* 2004], a way to reduce the gap is to use directly the light data measured from real world. Instead of introducing unnecessary conversions to construct empirical models. As illustrated in Figure 1.2, a more accurate digital model should be the measured-based one that is directly measured from real world, rather than the classical one constructed from empirical models.

For realistic rendering in Computer Graphics, speed and quality are two main requirements. Unbiased Monte Carlo techniques are successful methods to achieve high quality results [Szirmay-Kalos 2008], whereas parallel computing can deliver the speed requirement. Therefore, combining both of them leads to an ideal solution. However, speed and quality are orthogonal objectives, and thus tradeoffs are always needed. In order to satisfy flexibly any requirements of balancing, the solution should be scalable. A scalable solution can easily switch between speed and quality with the same algorithm. This property is even more helpful in the case of measured light sources, since measured data can be represented within different levels of accuracy, and the size of data has a big impact on performance.

In this thesis, we will focus on how to design scalable and unbiased solutions for realistic rendering with measured light sources, from real-time to offline. The success of this

work will lead to new applications, such as simulation, CAD, or virtual product design. Moreover, this is also a work that tries to reduce the gap between real world and virtual world.

1.1 Rendering using captured light sources from real world

Light sources are key components for realistic rendering since light is emanating from them. Without any light sources, we would not see anything. For realistic rendering, realistic light sources is thus essential. One way to obtain realistic light sources is to capture them from real world directly, and store them in a dedicated form. Images are very good containers and image-based lighting [Debevec 1998] has demonstrated its accuracy and efficiency for realistic rendering.

1.2 Two lighting effects: far-field and near-field

Lighting effects can be classified into two categories: far-field illumination and near-field illumination. The former one is generally caused by distant lighting model where the light sources are infinitely far away, and therefore they have only directional variations. Far field illumination is more likely to represent lighting effects for which spatial variations are not important. With this approximation, the light source model is simplified. The sacrifice of accuracy leads to speed improvement. This distant lighting model is already widely used in real-time applications and supported directly by graphics hardware. The latter one is more general. The spatial variations are taking into account, so its lighting effect includes many common and general lighting phenomena of real world. The distribution of directional variations is 2D. Storing and processing is quite cheap for today's computers. Spatial variations are dependent on the dimension of space (in this thesis, we only talk about 2D cases). To represent both directional and spatial information, 4D light sources are needed for far-field illumination. Due to the dimension, the main problem of using these light source models is the size of data which leads to memory cost and computation power consumption.

1.3 GPU computing

A Graphic Processing Unit (GPU) is a computing device designed for the purpose of graphics processing. Since 2006, the release of Gefore 8 series [NVIDIA 2006b] has leaded GPUs into a new period with generic stream processing architecture. The GPUs become more generalized computing devices with high parallel computing power. Moreover, the

new architecture gives more flexibilities to users with a programmable pipeline. Users can design more easily their parallel algorithms for the GPUs to reach the potential computation power of the devices. Since the implementation of our work is mainly done on GPUs, we will quickly describe its main properties.

Graphics Processing Unified Shader Model uses a consistent instruction set across all shader types (e.g., geometry, vertex and pixel shaders). Although the instruction set is not completely the same between different shader types, this model introduces a lot of programming flexibility. Currently, the most famous two graphics APIs supporting Unified Shader Model are Direct3D and OpenGL. The first one is specifically designed for some specific systems (e.g., Windows), whereas the second one is a generic solution for cross-platform applications. Although they are different APIs, they provide nearly the same level of functionalities [Microsoft 2013; Segal & Akeley 2013]. Although Rasterizer is still not programmable in the pipeline, it is very efficient for processing geometries which are represented by triangles. As a consequence, rasterization of 3D scenes still benefits from the pipeline and the rest of the processing steps can be dealt with the unified model flexibly.

General Computing Nowadays, GPUs are widely used for general computing. Instead of following the graphics pipeline and specific shaders, one can design highly parallel SIMD algorithms flexibly. On the one side, NVIDIA introduced the first GPU computing language, CUDA C [NVIDIA 2006a]. Currently, this language is specifically designed for NVIDIA CUDA GPUs. On the other side, Apple introduced a parallel computing language called OpenCL as a general and cross-platform solution [Apple 2007] which is managed by Khronos Group [khronos 2008] now. OpenCL is not only designed for GPU Computing, but also for other parallel computing devices. Nowadays, general computing is also embedded into the traditional Graphics APIs: Compute Shaders are introduced into both Direct3D 11 and OpenGL 4.3. Fast simulation can take benefit from these highly parallel architectures.

For the GPU implementation part of this thesis, we use both graphics processing and general computing, because, the rasterization pipeline is still useful for drawing 3D models. After this step, we use general computing to do per-pixel rendering. Finally, a post-processing on graphics processing pipeline is performed to generate images.

1.4 Problems and objectives

In this thesis, we mainly focus on how to use realistic light sources to render virtual objects with physically-based rendering techniques. According to the content above, the main objective is to develop a scalable solution for both quality and speed requirements. The accuracy of the rendering results can be verified by physics, but the performance remains a big problem even with parallel computing. Our first objective is real-time rendering with far-field lighting. Although rendering virtual objects with distant lighting in real time is already possible, real-time approach for dynamic environment maps that change over time is still an open question. The second objective is for near-field lighting. The use of 4D light field light sources for rendering cannot be done in real time. Therefore we should find a scalable solution to use such light source model for shading both in real time and offline efficiently. Another objective is balancing quality and speed. The solutions for both of the objectives should be unbiased to guarantee the accuracy, and they should also be real-time to satisfy the speed requirement. Monte Carlo integration is a good candidate to achieve these objectives, and the classical problem for Monte Carlo integration is the samples generation that needs to be as fast and efficient as possible. Therefore the last objective is that we have to generate and use samples efficiently.

For all of these objectives, there are detailed goals and related problems we have to face to: (sampling techniques and related concept will be discussed in the next caption)

Efficient sampling for far-field light sources. For far-field light sources, how to generate high quality samples and use them for shading both efficiently, in order to keep both quality and speed high enough?

Efficient sampling for near-field light sources. How to generate high quality samples from near-field light sources, and use them for shading efficiently? Since near-field light sources have 4D data, using a subset of the data is a reasonable way for efficiency as long as the subset retains the quality.

Combination with other sampling techniques. Since light-based sampling techniques cannot be efficient for arbitrary scenes, how can we combine these light-based sampling techniques with other sampling techniques such as BRDF sampling?

Corresponding unbiased estimators. How to design unbiased Monte Carlo estimators for the three objectives above?

In this thesis, we will give some solutions for all of these problems.

1.5 Contributions

In this thesis, we focus on the problem of efficient realistic rendering with both far-field and near-field illumination. The solutions of this problem leads to three different contributions.

Light source sampling for far-field illumination At the very beginning, we focus on the easiest illumination type: far-field illumination which has only 2D directional variations. The first work we made was sampling dynamic distant light sources stored in environment maps. It should be interesting, if we could capture realistic light sources from real world and use them to lit virtual scenes at the same time. For this motivation, we assume that cameras can capture environment lighting interactively. Therefore, to achieve the objective, the remaining problem is how to sample and use environment maps both in real time.

In order to solve this problem, we focus on generating high quality samples at low cost. We achieve this by sampling light sources by taking into account the light intensity and the cosine factor. An approximation of the cosine factor is introduced to maintain both sampling quality and performance speed. Combined with samples from BRDF, our effective sampling solution leads to high quality results with a low number of samples. With our GPU implementation, one can sample and use light sources from dynamic environment maps that change over-time for realistic rendering in real time.

Light source sampling for near-field illumination Near-field illumination can be captured by some dedicated devices. In this thesis, we focus on the one introduced by Goesele *et al.* [2003]. The main problem of such illumination is their 4D representation. At the beginning of the work, we face a problem statement that the measured data is large. It is very costly to use all of them to lit virtual objects. The latest published paper [Granier *et al.* 2003] about how to use this data for rendering is costly and inaccurate. This technique follows a traditional way that simulates light transport from the light source but with large approximations. Since accuracy and speed are both important, we decided to select and use a smaller set of data which has high contribution to the rendering.

In order to use such light source at interactive frame rate with high accuracy, we reduce its data size without introducing any approximations. Instead of doing global simulation with the whole light source, we focus on the light transport for each position in a 3D scene. Since we found that for a position, only a part of the light source contributes to the lighting, the data size can be conservatively reduced.

We introduce a method, that restricts the size of the lighting region for each pixel of the final rendering frame. Since the data outside of the region is omitted and contributes nothing to the lighting, the data size is conservatively reduced. With our new efficient

sampling technique, we can find important light sources which have high contributions to the lighting in the restricted regions. This technique leads to accurate rendering results with low number of samples. The computation cost of the technique is very low. With a GPU implementation, we achieve interactive frame rates on a personal computer.

Combination with other sampling techniques Different sampling techniques have different advantages [Pharr & Humphreys 2004]. Therefore combining different sampling techniques together is a reasonable way. Our third contribution is to combine light source sampling techniques with other sampling techniques by calculating their weights adaptively. The more important a sampling technique is, the larger its weight will be.

By default, sampling techniques are treated equally if the characteristics of a scene is unknown [Pharr & Humphreys 2004; Veach 1998], and thus their weights are the same. However this is not always efficient, since the properties of a sampling technique may not match the characteristics of the scene. We considered that enough prior knowledge of the scene may not always be available or too costly to get. Therefore, we introduced a "black-box" solution that uses a second order Taylor expansion to estimate the variance of the default combination. This estimation is very cheap and fast to compute. Then, we define weights dynamically for each sampling techniques according to the variance estimation. The chosen weights make the combination more efficient and therefore generate higher quality samples. Compared with state of art schemes, our technique reduces variance even further.

1.6 Organization of the rest document

The organization of the thesis is as follow. In Chapter 2, we give an overview on related topics, physically-based rendering techniques. Then different ways to represent light sources and their advantages and disadvantages are discussed. To conclude the chapter, we introduce captured light source and related techniques. In Chapter 3 and Chapter 4, we introduce our two contributions for sampling and rendering with captured light sources: far-field environment maps and near-field luminaires respectively. Then in Chapter 5, our work on distributing samples between lights source and BRDFs is introduced. Finally, in Chapter 6 we conclude and discuss all the subsequents researches that might be done in the near future based on our results.

1.7 Publications during the period

Lu, Heqi, Bao, Peng & Feng, Jieqing. (2012). OpenCL-based Real-time KD-Tree and Raytracing for Dynamic Scene. Journal of Computer-Aided Design & Computer Graphics. (Accepted to CAD/Graphics 2012 best student paper).

Lu, Heqi, Pacanowski, Romain & Granier, Xavier. (2013). Real-Time Importance Sampling of Dynamic Environment Maps. In Proceedings of Eurographics 2013. (Accepted to Eurographics 2013 short paper).

Lu, Heqi, Pacanowski, Romain & Granier, Xavier. (2013). A Second-Order Approximation for Variance Reduction in Multiple Importance Sampling. Computer Graphics Forum. (Accepted to Pacific Graphics 2013).

Lu, Heqi, Pacanowski, Romain & Granier, Xavier. (201X). Position-dependent Importance Sampling of Light Field Luminaires. Submitted to IEEE Transactions on Visualization and Computer Graphics. (Under revision).

Chapter 2

Previous Work

2.1 Physically-based rendering

As defined by Oxford Dictionaries ([Press 2013]), physics is the branch of science concerned with the nature and properties of matter and energy. It is reliable, at least rigorous in most of the aspects that human can observe. The study of light is one of the main subjects of physics. Thanks to light, the world is colorful rather than black. In Computer Graphics, based on a subset of physics laws, physically-based rendering techniques [Pharr & Humphreys 2004] are mainly dedicated to the simulation of light transport, to produce visually realistic rendering results. However, due to the limit of computing power, compromises are often required and lead to a very common problem in Computer Graphics: how to make the best tradeoff between efficiency and accuracy, between speed and quality.

Figure 2.1: *Left and right are two views of the San Miguel scene [Pharr & Humphreys 2004], lit by a sun which is represented by a uniform disk light source. The results are good from a visual point of view. However, in reality, the illumination from the sun is not a uniform disk*

Compromises we made for realistic rendering are approximations on different conceptual stages (as introduced by Arvo *et al.* [1994]). Each stage introduces a new error due to

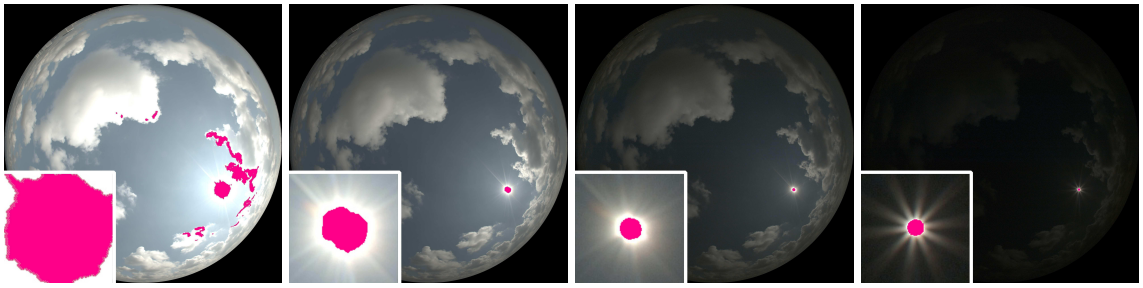


Figure 2.2: *Realistic sun: captured illumination from the sky with closeup of the sun using HDR camera with different exposure settings [Stumpfel* et al. *2004]. Pink regions indicate saturated pixels. Observe that, the illumination of the sun is not a uniform disk.*

a new additional approximation and reduces the accuracy once more. For efficiency reason, one common approximation made to realistic light sources is to use simple analytical light source models, such as area light sources which are still widely used in physically-based rendering techniques (cf. in Figure 2.1 where the sun is represented as an area light source). However, these models are only approximations to represent real light sources: the light source emits light uniformly from a specified area. Moreover, there are even simpler models such as point and directional light sources that are also commonly used for the same purpose. As illustrated in Figure 2.2, the illumination from the sun is neither a simple area light source nor a directional one. The incident lighting from the sun usually comes with atmospheric effects. Therefore, for more physical realism, we cannot resist using captured light sources to reduce the approximations.

As pointed by [Granier *et al.* 2003], the data from acquired light sources are more complex than classical models. Therefore, using such large data set generates new problems regarding the efficiency and the accuracy. Therefore, in order to cover this wide range of realism, scalable techniques become reasonable requirements and good choices. These techniques can easily balance between quality and speed instead of introducing large penalties to change the technique, so as to lead to a large variety of applications, from offline to online rendering and from design to entertainment. Recently, these rendering approaches show promising scalability for many researchers in the world both academia [Dachsbacher *et al.* 2013] and industry [Venkataraman 2012].

Before introducing how to use acquired light sources for realistic rendering, we will present the fundamentals of physically-based realistic rendering in the upcoming content. The concept behind realistic rendering is the simulation of light transport. Physically, light issued from light sources is scattered in the environment until it reaches an observer, which can be a human eye or a digital camera. The light transport problem has been placed in a unified context with the introduction of the rendering equation [Kajiya 1986].

The rendering equation describes the relationship between outgoing radiance and in-

coming radiance. In physics, radiance describes the quantity of radiation emitted or received within a unit solid angle (steradian) in a specified direction per surface area. It's unit is $W\dot{s}r^{-1}\dot{m}^{-2}$. Based on this formalism, the radiance leaving from a point $\boldsymbol{p}$ of a surface into a direction $\boldsymbol{o}$ is equal to the emitted radiance (if $\boldsymbol{p}$ is on a light source) plus the reflection of the incoming radiance from all incident directions:

$$\begin{aligned} L(\boldsymbol{p} \rightarrow \boldsymbol{o}) &= E(\boldsymbol{p} \rightarrow \boldsymbol{o}) + \int_{\Omega} \mathrm{s}(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega})\, \mathrm{d}\boldsymbol{\omega} \quad \text{with} \\ \mathrm{s}(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega}) &= \rho(\boldsymbol{o}, \boldsymbol{\omega})\ \langle \boldsymbol{p}, \boldsymbol{\omega} \rangle\ V(\boldsymbol{p}, \boldsymbol{\omega})\, L(\boldsymbol{\omega} \rightarrow \boldsymbol{p})\,. \end{aligned} \tag{2.1}$$

In this equation, Ω is the unit hemisphere, $\langle \boldsymbol{p}, \boldsymbol{\omega} \rangle$ expresses a positive clamped dot product operator (aka *cosine factor*), $V(\boldsymbol{p}, \boldsymbol{\omega})$ is the visibility function that describes the occultation between $\boldsymbol{p}$ and $\boldsymbol{\omega}$. $L(\boldsymbol{\omega} \rightarrow \boldsymbol{p})$ represents the incoming radiance from the direction $\boldsymbol{\omega}$ forward the point $\boldsymbol{p}$, $E(\boldsymbol{p} \rightarrow \boldsymbol{o})$ is the emitted radiance, and $\rho(\boldsymbol{o}, \boldsymbol{\omega})$ denotes the Bidirectional Reflection Distribution Function (BRDF). A BRDF is a function that defines how light is reflected on a surface [Nicodemus 1965]. Given an incoming light direction $\boldsymbol{\omega}$, the BRDF describes the ratio of reflected in radiance for an outgoing direction $\boldsymbol{o}$, For more general scattering (e.g., transmission and refraction), it can be replaced by a Bidirectional Scattering Distribution Function (BSDF) [Heckbert 1991]. However, due to the definition of a BSDF that it is sometimes over the full unit sphere, the Ω must be changed to the full unit sphere to keep energy conservation.

Many different methods to solve the rendering equation have been developed, such as finite element methods (e.g., radiosity-based techniques [Greenberg *et al.* 1986]), Monte Carlo methods (e.g., path tracing [Kajiya 1986], metropolis light transport [Veach & Guibas 1997]) and density estimation methods (e.g., photon mapping [Jensen 1996b]), which we will discuss in the next section.

2.1.1 Problem Statement

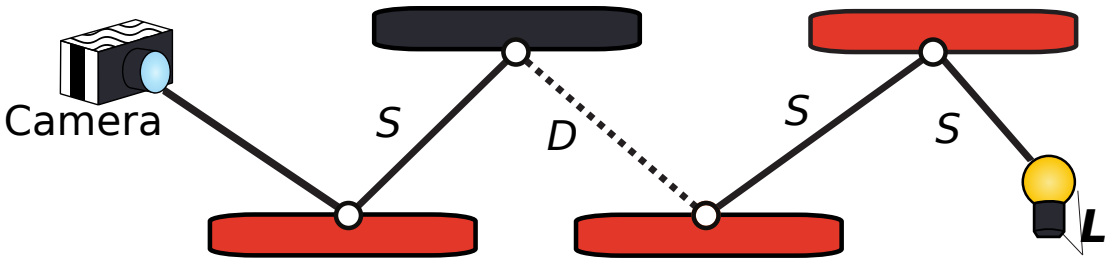


Figure 2.3: *High-dimensionality problem due to the recursion. A visible ray is issued from a light source, the path is reflected by different surfaces: specular-specular-diffuse-specular, before reaching the observer: camera.*

As illustrated in Figure 2.3, one of the principle problem encountered when solving the rendering equation (cf. Equation 2.1) is the possibility of its high-dimensionality. Since the incoming radiance is the result of the transfer of the outgoing radiance from another visible point , the computation is recursive. To clarify this point, we simply rewrite the integration as

$$T(\boldsymbol{L}) = \int_{\Omega} \mathrm{s}(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega})\, \mathrm{d}\boldsymbol{\omega}$$
$$\boldsymbol{E} = E(\boldsymbol{p} \rightarrow \boldsymbol{o})$$

where $T(\boldsymbol{L})$ is the transport operator [Kajiya 1986] applied on outgoing radiance $\boldsymbol{L}$ from another visible point and $\boldsymbol{E}$ is the emitted radiance. Then the rendering equation is successively derived as a Neumann series [Neumann 1877]:

$$\begin{aligned}
\boldsymbol{L} &= \boldsymbol{E} + T(\boldsymbol{L}) \\
\boldsymbol{L} &= \boldsymbol{E} + T(\boldsymbol{E} + T(\boldsymbol{L})) = \sum_{i=0}^{2} T^i(\boldsymbol{E}) + T^2(\boldsymbol{L}) \\
\boldsymbol{L} &= \sum_{i=0}^{3} T^i(\boldsymbol{E}) + T^3(\boldsymbol{L}) \\
\boldsymbol{L} &= \sum_{i=0}^{+\infty} T^i(\boldsymbol{E}) + T^{+\infty}(\boldsymbol{L})\,.
\end{aligned} \tag{2.2}$$

Each new line in Equation 2.2 corresponds to a new light bounce. Note that, the recursion only stops when the BRDF is equal to 0 in all directions. This is the case when incoming radiance in all directions only comes from pure light sources (at N bounces of recursion).

$$\boldsymbol{L} = \sum_{i=0}^{N} T^i(\boldsymbol{E})\,.$$

There is another major problem to compute Equation 2.1 even with only one bounce: generally, there is no analytic solution for this type of equation. Therefore, we have to use numerical methods to solve it.

2.2 Finite Element Methods

One category of numerical techniques for solving the rendering equation is finite element methods. These deterministic methods discretize the rendering equation to a linear system. Inspired by radiation heat transfer [Siege1 & Howell 1981], Goral *et al.* [1984] have introduced a method for object-to-object reflection, to solve the light transport problem in a diffuse radiative context: the radiosity equation.

Radiosity technique is an instance of finite element methods. It is deterministic and can render complex diffuse interreflections between Lambertian surfaces: the computation of directional radiance is not directly applicable in such case. In 1986 Greenberg *et al.* [1986] have extended the radiosity method for more general BRDFs. This method has created promising research directions during 1980s to early 1990s where many improvements have been made to the basic radiosity method [Sillion & Puech 1994]. However, the methods lead to difficult discretization and memory problem, since surface and space of direction need to be discretized. Such restriction hinder the research in advance. Since later 1990's, Monte Carlo methods have become more popular solutions. Nonetheless, radiosity is still commonly used where the scenes are mostly filled with diffuse objects.

Another good property of radiosity is that the illumination can be computed only once, and later rendered for multiple viewpoints. It is an advantage for interactive applications. With the increase of GPU computing power, radiosity was also transfered from CPU to GPU solutions. Carr *et al.* [2003] gave a GPU implementation of hierarchical radiosity with CPU for preprocessing. However, this technique is limited in terms of scene complexity (i.e., the number of triangles). In 2005, Pharr & Fernando [2005] introduced a new GPU implementation to reduce this limitation. By porting progressive radiosity [Cohen *et al.* 1988] on the GPU, it can render scenes with over one million elements. Radiosity is also integrated into video game engines with a real-time radiosity architecture [Tatarchuk 2010].

Figure 2.4: *Radiosity applications: (**Left**), A quasi-realtime radiosity called Enlighten (Geomerics) used in Frostbite™2 Game Engine (EA DICE™). (**Right**), "Radiosity Renderer and Visualizer (http://dudka.cz/rrv)", implemented with C++ and OpenGL.*

Recently, finite element methods have also been proved to be reliable for rendering of translucent materials. These materials such as skin, minerals and marble have sub-surface scattering events during light transport. In 2009, Arbree *et al.* [2011] introduced a method that use finite element method for heterogeneous subsurface scattering. Jakob *et al.* [2010] extended this method to anisotropic materials by providing a faster solution for the anisotropic diffusing equation. More recently, Sheng *et al.* [2013] have developed

a practical analytic model for the radiosity transport in translucent scenes. For this purpose, they have introduced a subsurface scattering matrix that operates in conjunction with a matrix that is used in traditional radiosity techniques. They have also introduced a fast iterative solution to radiosity. Thanks to these improvements, they can dynamically relight and change translucency of materials at nearly interactive frame rate.

However, there is a common limit for radiosity techniques. When surfaces have highly directional attributes, the requirement in dense discretization or clever adaptive meshing makes these techniques become impracticable for complex scenes.

2.3 Monte Carlo Integration

For high dimensional problem, Monte Carlo integration methods are generally preferred. Monte Carlo integration numerically computes an integral based on random samples instead of a defined discretization scheme. The main feature of this technique compared with other numerical integration ones is: the convergence rate is independent of the dimensionality of the integrand. It thus provides a suitable solution to the problem of estimating high-dimensional integrals such as the rendering equation.

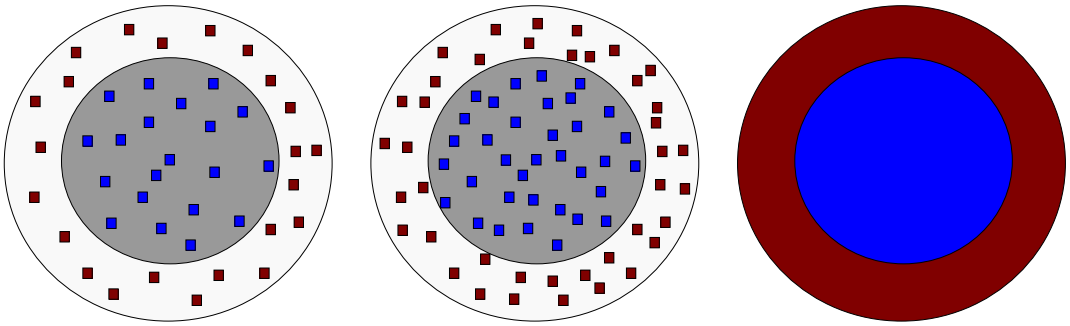


Figure 2.5: *Monte Carlo integration: estimating area using random samples. (From **Left** to **Middle**): the areas of the two concentric disks are proportional to the number of samples. More samples lead to more accurate estimation. (**Right**) Infinitive number of samples leads to the exact solution.*

The key idea of Monte Carlo integration is sampling. Rather than computing an exact solution, it distributes samples randomly to estimate the solution. For example in Figure 2.5, the areas can be estimated by distributing random samples. The more samples are drown, the more accurate of the estimation will be. If there is an infinite number of samples, the estimation will converge to the exact solution.

For simplicity, we will focus only on the first light bounce (direct lighting) from now on. Note that, this simplification is introduced without loss of generality because of the main feature of Monte Carlo Integration (i.e. Independent of the dimensionality of the integrand).

In this context, the reflected radiance is approximated using the following estimator:

$$L_N(\boldsymbol{p} \rightarrow \boldsymbol{o}) = \frac{1}{N_s} \sum_{i=1}^{N_s} \frac{\mathrm{s}(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega}_i)}{\mathrm{pdf}(\boldsymbol{\omega}_i)} \tag{2.3}$$

where $\mathrm{pdf}(\boldsymbol{\omega}_i)$ is the Probability Density Function (PDF) used to generate the sample $\boldsymbol{\omega}_i$, and $\mathrm{s}(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega}_i)$ is the shading function limited to the direct contribution of light sources. As we will show in the next section, the closer the pdf matches $\mathrm{s}(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega}_i)$, the lower number of samples N_s will be required to obtain a nearly converged solution. Moreover, if the PDF perfectly matches the numerator, only one sample is enough to compute the converge result. This is only possible if the analytical solution is known.

2.3.1 Quantitative Criteria

There are two criteria to estimate the quality of Monte Carlo estimation: bias and convergence. Bias is estimated with expected value, whereas convergence is related to variance.

The expected value $E[L_N(\boldsymbol{p} \rightarrow \boldsymbol{o})]$ of the estimator $L_N(\boldsymbol{p} \rightarrow \boldsymbol{o})$ is the average of all estimations with different sample distributions:

$$E[L_N(\boldsymbol{p} \rightarrow \boldsymbol{o})] = \lim_{K->\infty} \frac{1}{K} \sum_{i=1}^{K} L_{N_i}(\boldsymbol{p} \rightarrow \boldsymbol{o}) \tag{2.4}$$

where N_i expresses the i^{th} distribution of N random samples.

Note that, the expected value is also the converged solution that we get with an infinite number of samples. Bias is defined as the distance between this expected value and the integral we want to estimate:

$$\text{bias} = |E[L_N(\boldsymbol{p} \rightarrow \boldsymbol{o})] - L(\boldsymbol{p} \rightarrow \boldsymbol{o})| \ .$$

If the bias of the estimator is equal to zero, the estimate is said to be unbiased.

If there is not enough samples used for the estimation, the result will be different to the expected value. This difference is estimated with variance $V[L_N(\boldsymbol{p} \rightarrow \boldsymbol{o})]$, which is defined as the expected squared deviation of the function from its expected value:

$$\begin{aligned} V[L_N(\boldsymbol{p} \rightarrow \boldsymbol{o})] &= E[(L_N(\boldsymbol{p} \rightarrow \boldsymbol{o}) - E[L_N(\boldsymbol{p} \rightarrow \boldsymbol{o})])^2] \\ V[L_N(\boldsymbol{p} \rightarrow \boldsymbol{o})] &= E[(L_N(\boldsymbol{p} \rightarrow \boldsymbol{o}))^2] - E[L_N(\boldsymbol{p} \rightarrow \boldsymbol{o})]^2 \ . \end{aligned} \tag{2.5}$$

Standard deviation $\sigma[L_N(\boldsymbol{p} \rightarrow \boldsymbol{o})]$ is computed as the square root of variance:

$$\sigma[L_N(\boldsymbol{p} \rightarrow \boldsymbol{o})] = \sqrt{V[L_N(\boldsymbol{p} \rightarrow \boldsymbol{o})]} \ . \tag{2.6}$$

For an estimator, there is a direct relationship between its variance and the quality of its sampling technique. Actually, the variance of the estimation is proportional to the variance of its sampling technique [Veach 1998]:

$$V[L_N(\boldsymbol{p} \to \boldsymbol{o})] = \frac{1}{N} V[L_1(\boldsymbol{p} \to \boldsymbol{o})] \,. \tag{2.7}$$

This equation means that, the more constant the ratio $\frac{s(\boldsymbol{p},\boldsymbol{o},\boldsymbol{\omega}_i)}{\text{pdf}(\boldsymbol{\omega}_i)}$ is, the faster the convergence will be. If the ratio is constant, the variance is 0. This is achieved when $\text{pdf}(\boldsymbol{\omega}_i) = \frac{1}{L} s(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega}_i)$. This is the case only if both an analytical and invertible form of the integral of $s(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega}_i)$ exist. In this thesis, we want to find a sampling technique that is as close as possible to the integrand to reduce variance. The sampling variance $V[L_1(\boldsymbol{p} \to \boldsymbol{o})]$ can be directly computed using equation 2.5:

$$\begin{aligned} V[L_1(\boldsymbol{p} \to \boldsymbol{o})] &= E[(L_1(\boldsymbol{p} \to \boldsymbol{o}) - E[L_1(\boldsymbol{p} \to \boldsymbol{o})])^2] \\ V[L_1(\boldsymbol{p} \to \boldsymbol{o})] &= E[(L_1(\boldsymbol{p} \to \boldsymbol{o}))^2] - E[L_1(\boldsymbol{p} \to \boldsymbol{o})]^2 \,. \end{aligned} \tag{2.8}$$

We will use sampling variance in the rest of the document as a quantitative tool.

As previously introduced, expected value is related to bias whereas variance is related to convergence speed. Therefore, to improve the convergence speed of unbiased Monte Carlo techniques, we have to focus on how to reduce variance. A lot of work has been done to achieve this objective (e.g., [Burke *et al.* 2005; Pajot *et al.* 2011; Veach 1998]). With the same number of samples, lower variance is equivalent to better convergence rate.

Computation time is also a very important factor. In order to evaluate different estimators in more general context, timing cost $T[L_N(\boldsymbol{p} \to \boldsymbol{o})]$ is also introduced. The combined quantitative criterion is called the efficiency $\epsilon[L_N(\boldsymbol{p} \to \boldsymbol{o})]$, as introduced by [Veach 1998]:

$$\epsilon[L_N(\boldsymbol{p} \to \boldsymbol{o})] = \frac{1}{V[L_N(\boldsymbol{p} \to \boldsymbol{o})] T[L_N(\boldsymbol{p} \to \boldsymbol{o})]} \,. \tag{2.9}$$

According to this metric, lower variance or lower time cost will increase the estimator efficiency.

2.3.2 PDF and CDF

There are two important concepts for sampling, one is probability density function (PDF) and the other is cumulative distribution function (CDF). For sake of clarity, we start with a discrete case. Assuming that we have N light sources which intensities are I_i, the probability of selecting one of them is:

$$P_i = \frac{I_i}{\sum_{j=1}^{N} I_j} \,.$$

There are two properties for discrete probabilities:

$$P_i \in [0, 1]$$
$$\sum_{i=1}^{N} P_i = 1 .$$

The probability of a finite set of events should be non-negative and smaller than 1. The possibilities of all samples must sum to 1, otherwise bias is introduced.

A PDF extends this idea to continuous variable. The probability of a random variable belonging to an integral $[a, b]$ is the integral of this density function.

$$P\left[X \in [a, b]\right] = \int_a^b \text{pdf}(X) d(X) .$$

Furthermore, a PDF has the following two properties,

$$\text{pdf}(X) \geq 0$$
$$\int_{-\infty}^{\infty} \text{pdf}(X) dX = 1 .$$

Note that, the interval of an integral can be smaller than 1. Therefore a PDF can also be larger than 1. The only restriction is that the full integration must be equal to 1.

A CDF is the integral of a PDF from lower bound of the domain to a given value x. It represents the probability that the random variable X takes a value less than or equal to x. A PDF is the derivative of a CDF at a given value.

$$\text{c}df(x) = P[X \leq x] = \int_{-\infty}^{x} \text{pdf}(X) dX$$
$$\text{pdf}(x) = \frac{d\text{c}df(x)}{dx} .$$

The probability of a domain $[a, b]$ is:

$$P\left[x \in [a, b]\right] = \text{c}df(b) - \text{c}df(a) .$$

CDF has two interesting characteristics. Firstly, since a CDF is the integral of a PDF, it is smoother than the PDF. Secondly, as an integral of a positive function, CDF is monotonically increasing.

2.3.3 Random Number Generators

In general, any generated random sample is based on a random number ε. The distribution of this random number is uniformly defined on the interval $[0, 1]$. Random numbers can be

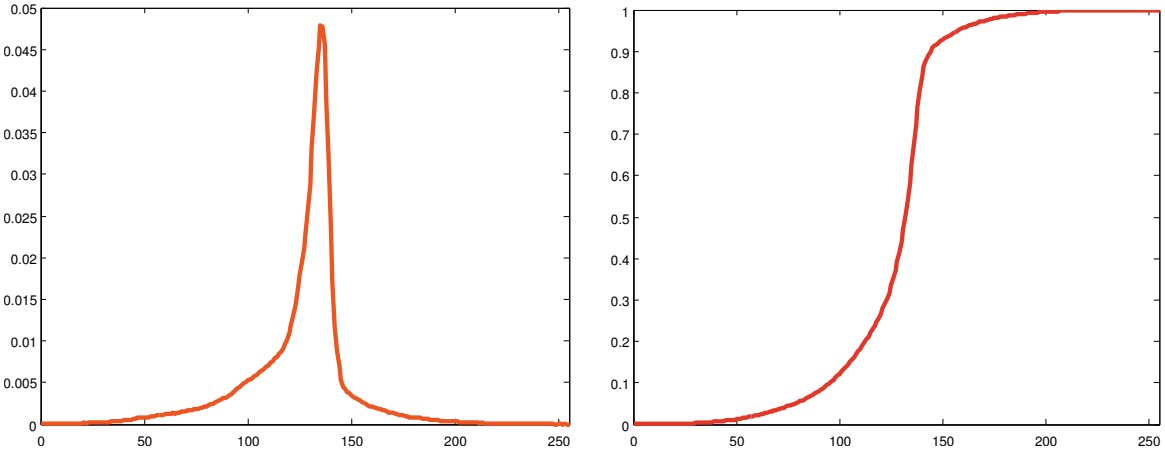


Figure 2.6: *PDF and CDF: the CDF (**Right**) is monotonically increasing and is smoother than (**Left**) its corresponding PDF.*

generated by a random number generator, which is commonly a random sequence.

Mersenne twister [Matsumoto & Nishimura 1998] | Halton [Halton 1964] | Hammersley [Hammersley 1960]

Figure 2.7: *Uniform sampling using different random number generators. Random sequences with low-discrepancy property such as Halton and Hammersley sequences produce better visual quality than classical random number generator (Mersenne twister).*

There are many random number generators. A straightforward method is to use the default random number generators from C programming language or more robust ones such as Mersenne Twister [Matsumoto & Nishimura 1998]. However, the resulting random sequences have low coherence between samples. This high-discrepancy property leads to visually noisy results. A better sampling technique should generate low-discrepancy samples [Pharr & Humphreys 2004], in order to have more coherency amongst themselves. This feature can reduce the noise and increase the visual quality (cf. Figure 2.7). The most commonly used low-discrepancy random sequences are Halton sequence and Hammersley sequence.

Random sequences are not real random numbers, they have limited property. Although, all random sequences keep the random distribution property when using all elements of

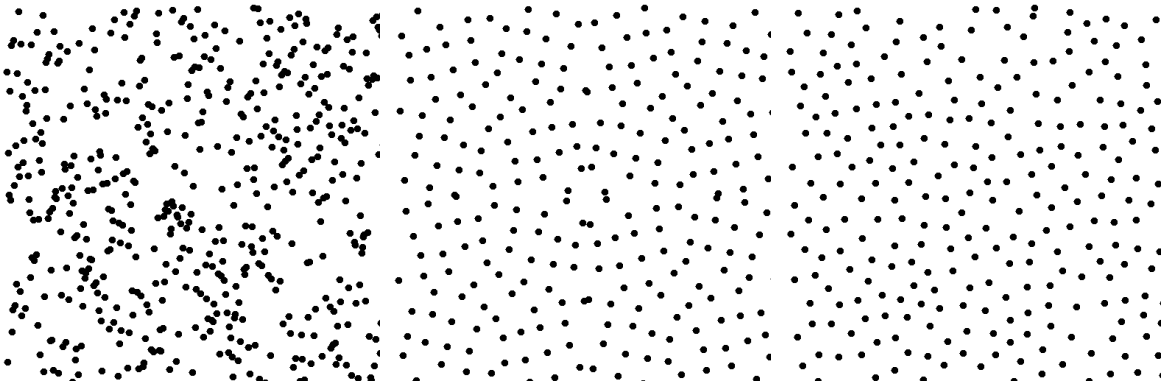
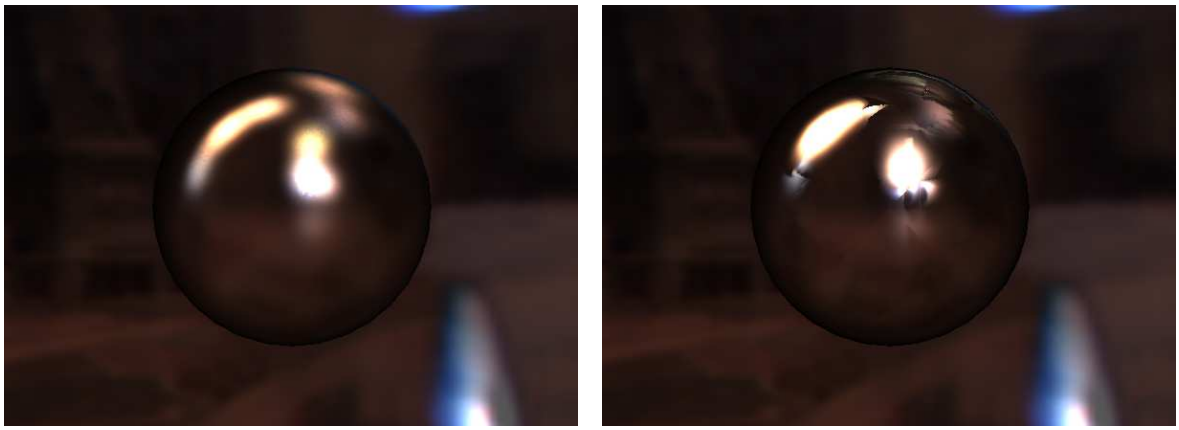


Figure 2.8: *Initial random number distribution, (**Left**) machine random number generated by invoking srand() function. (**Middle**) Low-discrepancy random number. (**Right**) ideal random sample distribution for rendering: Poisson disk distribution.*



Halton Sequence Hammersley sequence.

Figure 2.9: *Using a subset (256 samples) of regenerated random sequence (512 samples) for uniform sampling, Halton sequence still has a visually good sampling distribution whereas Hammersley sequence fails in this case.*

a sequence, for a subset of the sequence, the numbers may not be random at all (i.e., the distribution may not uniformly cover the domain $[0, 1]$). As shown in Figure 2.9, this has a direct impact on the rendering quality: when pregenerating a sequence with N numbers, but using only previous M numbers (M < N). The Halton subsequence is still a random sequence whereas the Hammersley one is not.

If a subset of the sequence maintains the property of a random distribution, we say that it has incremental ability [Pharr & Humphreys 2004]. This attribute is very important for dynamic random sample distribution, especially for interactive applications. Indeed, for efficiency purpose, rather than generating samples after knowing the required number of them, it is cheaper to precompute a large sequence and dynamically select a subset that is still covering the full distribution domain.

2.3.4 Uniform Sampling

Based on these initial random numbers, random samples for directional light source can be generated using different sampling strategies. The simplest one is the uniform sampling sphere. This sampling strategy is conservative since it involves all possible lighting directions. In local spherical coordinates, using two initial random numbers ξ_1 and ξ_2, which are defined in $[0, 1]$, the coordinate of generated sample is:

$$\begin{aligned} \theta &= \arccos(1 - 2\xi_1) \\ \phi &= 2\pi\xi_2 \end{aligned} \tag{2.10}$$

where θ is defined in $[0, \pi]$ and ϕ is defined in $[0, 2\pi]$.

If there are only reflective surfaces in the scene, a uniform sampling hemisphere is also conservative since reflection only happens in the hemisphere oriented along the surface normal. In local spherical coordinates, this sampling is formulated as:

$$\begin{aligned} \theta &= \arccos(\xi_1) \\ \phi &= 2\pi\xi_2 \end{aligned} \tag{2.11}$$

where θ is defined in $[0, \frac{\pi}{2}]$ and ϕ is defined in $[0, 2\pi]$.

Sampling over the sphere or the hemisphere is simple, reliable and unbiased. Therefore, they can be used as reference sampling strategies.

2.3.5 Importance Sampling

Uniform sampling over the hemisphere is effective only when the contribution from different directions is also uniform. However, this is rarely the case, especially for realistic lighting. BRDF, cosine factor and light sources are varying along the different lighting directions. Furthermore, their product may introduce more variations.

Importance sampling is specially designed to take into account these variations, the more contribution is coming from a direction, the more likely the sampling strategy will sample this direction. Important samples are supposed to contribute more to the estimator, reduce variance and increase the convergence rate. The main problem of such sampling technique, is to find what are the important factors and how to generate samples based on these factors.

We recall that the Monte Carlo estimator with N samples is:

$$L_N(\boldsymbol{p} \rightarrow \boldsymbol{o}) = \frac{1}{N} \sum_{i=1}^{N} \frac{\mathrm{s}(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega}_i)}{\mathrm{pdf}(\boldsymbol{\omega}_i)} \,. \tag{2.12}$$

In this equation, as we mentioned before, the more is the pdf($\boldsymbol{\omega}_i$) similar to s($\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega}$), the faster is the convergence rate. Therefore, the main idea of importance sampling is to find samples of which pdfs are similar to s($\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega}$).

Since s($\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega}$) is composed of a BRDF, a cosine factor and incoming lighting, these three quantities are important factors for sampling. A straightforward way to generate samples is to sample according to one of these quantities solely.

Sampling On Cosine The simplest important sampling is sampling weighted by the cosine factor. In local spherical coordinates, the sampling formula is:

$$\begin{aligned} \theta &= \arccos(\sqrt{\xi_1}) \\ \phi &= 2\pi\xi_2 \,. \end{aligned} \tag{2.13}$$

Cosine factor varies along lighting directions, so the PDFs of samples from cosine-based importance sampling are different in each direction:

$$\text{pdf}(\theta, \phi) = \frac{\cos(\theta)}{\pi} = \frac{\sqrt{\xi_1}}{\pi} \,. \tag{2.14}$$

Sampling on BRDF BRDF important sampling is more complex. Like cosine factor, BRDFs vary along lighting directions as well. Furthermore, BRDFs have different representations, sampling on a BRDF has to follow the property of its representation. Physically-based Phong BRDF [Lafortune & Willems 1994] is one of the simplest physically realistic analytical BRDF model. It is composed with a diffuse part and a specular lobe:

$$\begin{aligned} \rho(\boldsymbol{o}, \boldsymbol{\omega}) &= k_d \frac{1}{\pi} + k_s \frac{n+2}{2\pi} \cos^n \alpha \\ \text{with } & k_d + k_s = 1 \,. \end{aligned} \tag{2.15}$$

In this equation k_d is positive diffuse albedo, k_s is positive specular albedo, n is the exponential of Phong lobe, and α is the angle between the perfect specular reflective direction and the outgoing direction.

This BRDF can be sampled by sampling the diffuse part and the specular part respectively [Lafortune & Willems 1994]. It is done in two steps. The first step selects randomly one of these two parts. The second step generates samples on the selected part. For the diffuse part, it is the uniform sampling over hemisphere with a cosine weight (cf. Equa-

tion 2.13), for the specular part, it is:

$$\begin{aligned}\theta &= \arcsin\left(\sqrt{1-\xi_1^{\frac{1}{n+2}}}\right)\\ \phi &= 2\pi\xi_2.\end{aligned} \tag{2.16}$$

The specular-part of the PDF of Equation 2.16 is:

$$\text{pdf}\left(\alpha(\theta,\phi)\right) = \frac{n+1}{2\pi}\cos^n\alpha\,. \tag{2.17}$$

Note that, the PDF is only related to α. Once again, this corresponds to the use of the inverse of the CDF to compute the samples.

$$\begin{aligned}\alpha &= \text{c}df^{-1}(\xi_1)\\ \alpha &= \arccos\xi_1^{\frac{1}{n+1}}\,.\end{aligned}$$

Sampling on light source For light-based importance sampling, we may use their directional intensity. For example, if light sources are stored in a 2D image (cf. Section 2.7.1), importance sampling can be done by randomly selecting texels of the image. Furthermore, each texel records a constant value, the PDF of the light source represented by this texel is thus piecewise-constant.

If we use a piecewise constant 2D function $f(u,v)$ to represent the light source, we can store it in a 2D image. The full integral of $f(u,v)$ is

$$\boldsymbol{I}_f = \sum_i\sum_j f_{i,j}$$

where $f_{i,j}$ is the value of texel (i,j). The corresponding 2D PDF $\text{pdf}(u,v)$ is

$$\text{pdf}(u,v) = \frac{f(u,v)}{\boldsymbol{I}_f}\,.$$

A multidimensional PDF can be decomposed into a product of conditional PDFs: $\text{pdf}(u,v) = \text{pdf}(u)\text{pdf}(v|u)$ where $\text{pdf}(v|u)$ is the conditional PDF of v knowing u. In the context of a piecewise constant function, we have

$$\begin{aligned}\text{pdf}(u) &= \frac{\sum_i f(u,v_i)}{\boldsymbol{I}_f}\\ \text{pdf}(v|u) &= \frac{f(u,v)/\boldsymbol{I}_f}{\text{pdf}(u)}\,.\end{aligned}$$

Note that pdf($v|u$) can be computed without knowing $\boldsymbol{I}_f$. pdf($v|u$) is also a 2D piecewise constant function and can be stored in a 2D image.

$$\text{pdf}(v|u) = \frac{f(u,v)}{\sum_i f(u,v_i)}.$$

It would be very good, if an importance sampling technique could sample based on BRDF, light and cosine all together. However, although BRDF, light and cosine can be directly sampled separately, sampling the combination of them is not straightforward, since, they are independent: high intensity light might be blocked and thus does not contribute to the final result, low intensity light may contribute a lot when the BRDF is large. In general, there is no analytical solution of this combination. To solve this problem, combined sampling techniques have been introduced [Burke *et al.* 2005; Veach 1998; Wang & Akerlund 2009], we will discuss them in the next subsection.

Importance sampling is not always helpful to increase convergence rate. The variance of an estimator may increase if a poor sampling distribution is used. This is the case when the selected PDF has an opposite behavior compared to the integrand.

2.3.6 Combined Sampling Techniques

Combining different sampling strategies together leads to better approximation of the integrand of the rendering equation. Two kinds of techniques for importance sampling on multiple important factors have been developed.

The first category is bidirectional importance sampling that generate samples according to the product s($\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega}$). Based on Sampling Importance Resampling (SIR) technique [Burke *et al.* 2005; Talbot *et al.* 2005], initial samples are first generated using light-based or BRDF-based importance sampling. Their contribution is then evaluated using an approximation of the shading function. Part of these initial samples are selected later for final estimation. Intuitively, this method keeps only the best samples from the initial set.

Instead of generating initial samples, other researchers have proposed methods that focus on reconstructing a suitable representation for importance sampling [Clarberg *et al.* 2005; Cline *et al.* 2006]. As an example, based on illumination cut [Cheslack-Postava *et al.* 2008], Wang & Akerlund [2009] introduced a method that firstly clusters light sources by reconstructing them to lighter representations (illumination cut), which can be efficiently sampled by light-based importance sampling. Then, these clusters are updated by taking BRDF information into account. This updating is equivalent to a BRDF-based sampling. After these two steps, an approximated distribution of the product of light and BRDF is constructed. Finally, visibility samples are selected from this new distribution for the final shading estimation.

The second one is Multiple Importance Sampling (MIS) [Veach 1998]. It is a technique that follows the divide and conquer idea: instead of sampling the whole product, this technique combines different importance sampling techniques together with weights. It is much easier than bidirectional important sampling techniques since independent sampling from multiple distributions is quite straightforward. Another very interesting feature for MIS is that, if one sampling technique is unbiased, the final estimation is also unbiased. Combining different sampling techniques with different properties, biased or unbiased, provides a flexible solution, and thus match the purpose of scalability (between accuracy and efficiency).

If there are S sampling strategies combined with MIS, the new Monte Carlo estimator becomes:

$$L_{N_0 \ldots N_S}(\boldsymbol{p} \rightarrow \boldsymbol{o}) = \sum_{s=0}^{S} \frac{1}{N_s} \sum_{i=0}^{N_s} w_s(\omega_{s,i}) \frac{\mathrm{s}(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega}_{s,i})}{\mathrm{pdf}_s(\boldsymbol{\omega}_{s,i})}$$

where N_s is number of samples for the s^{th} sampling technique. $\boldsymbol{\omega}_{s,i}$ is a sample from sth sampling technique and $\mathrm{pdf}_s(\boldsymbol{\omega}_{s,i})$ is the corresponding PDF. The weights w_s are used to balance among different sampling techniques. They must satisfy the condition $\sum_{s=0}^{S} w_s(\omega_{s,i}) = 1$ to avoid bias.

For example, with two sampling strategies such as light-based importance sampling and BRDF-based importance sampling, the new Monte Carlo estimator with MIS is:

$$L_{N_L,N_B}(\boldsymbol{p} \rightarrow \boldsymbol{o}) = \frac{1}{N_L} \sum_{i} w_L(\omega_i) \frac{\mathrm{s}(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega}_i)}{\mathrm{pdf}_L(\boldsymbol{\omega}_i)} + \frac{1}{N_B} \sum_{j} w_B(\omega_j) \frac{\mathrm{s}(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega}_j)}{\mathrm{pdf}_B(\boldsymbol{\omega}_j)}$$

where N_L is the number of samples for light-based sampling, and N_B is the number of samples for BRDF-based sampling. $\boldsymbol{\omega}_i$ is a sample from light-based sampling, whereas $\boldsymbol{\omega}_j$ is a sample from BRDF-based one, $\mathrm{pdf}_L(\boldsymbol{\omega}_i)$ and $\mathrm{pdf}_B(\boldsymbol{\omega}_j)$ are their respective PDFs. A pair of weights w_L and w_B is used to balance between these sampling techniques. It must satisfy the condition $w_L + w_B = 1$. For the sake of simplicity, we will use this equation in the rest of this thesis.

Note that, generating samples with the two independent sampling strategies respectively is generally known, only the two weights w_L and w_B need to be investigated. There are two subproblems to be solved for MIS, the first problem is how many samples to generate from each sampling strategies, the second one is which weighting scheme should be used for them.

The former one is introduced to take into account combinations of the different properties of sampling techniques. If a prior knowledge is already known, such as the relationship between a sampling technique and the scene characteristics, it will be quite straightforward to predefine the number of samples ratio among sampling techniques [Pajot *et al.* 2011].

For example, if we know the surface is fully diffuse, BRDF does not influence the choice of directions. Therefore, only light sources have a major impact and using only the light-based sampling for generating all samples will be the best choice, which leads to $w_L = 1$ and $w_B = 0$. However, in most cases, the properties are unknown, especially when using captured light source illumination with complex materials and geometries. In general, a equal balancing is used that we assume different sampling techniques have the same quality. With this assumption, we would distribute the same number of samples for each sampling strategy.

The later problem is introduced by the factor that not all of the samples generated by a sampling technique have high quality. Some bad samples may increase the variance of the estimation. In order to increase the contribution of high quality samples, a weighting scheme should be defined. In the following, we present the most popular weighting schemes: balance heuristic, power heuristic, max heuristic [Veach 1998].

The balance heuristic is a fair scheme. The concept behind it is that the weight should be proportional to the PDF and the number of samples. For s^{th} sampling technique, the weighting factor of ith sample is thus:

$$w_s(x_i) = \frac{N_s \text{pdf}_s(x_i)}{\sum_j N_j \text{pdf}_j(x_i)} .$$

For example, if there are two sampling techniques such as light-based and BRDF-based, s would be one of these two techniques. With equal balancing, the same number of samples would be distribute between these sampling techniques, and thus $N_L = N_B$. If the PDF of a sample x is very small, then the corresponding weight is also very small, to decrease the impact of such bad samples.

The power heuristic sometimes reduces variance even further.

$$w_s(x) = \frac{(N_s \text{pdf}_s(x))^\beta}{\sum_i (N_i \text{pdf}_i(x))^\beta} .$$

In practice, the exponent β is by default set to 2. The intuition behind power heuristic is to increase even more the weight of high PDF samples. If the β is set to positive infinity, this weighting is equivalent to the max heuristic [Georgiev *et al.* 2012]:

$$w_s(x) = \begin{cases} 1 & \text{if } \forall i \in S : n_s p_s(x) \geq n_i p_i(x) \\ 0 & \text{otherwise} . \end{cases}$$

The concept of max heuristic is to use only samples with the highest PDF for each iteration of the estimation. This technique will waste a lot of samples that have already been computed, but it can prevent low PDF samples from being used to increase stability of the

results. This approach is especially efficient when different sampling techniques are almost orthogonal. The orthogonal sampling techniques can be designed by different study cases, the idea is to use orthogonal basis [Georgiev *et al.* 2012].

2.3.7 Conclusion

Monte Carlo methods are efficient solutions for the light transport problem, due to their appealing properties: flexibility, unbiased and dimension-independent. Furthermore, the accuracy of the estimation can be improved by increasing the number of samples.

However, before reaching the full convergence of an estimation, noise exists and reduces the visual quality. There are three quantitative criteria to evaluate convergence. Expected value is used to evaluate the bias of the estimation whereas variance is used to evaluate the convergence process. Furthermore the efficiency of an estimation can be measured using variance and processing time.

In order to reduce variance and increase the convergence rate, we need a efficient sampling technique to generate good samples whose PDF should match the integrand as close as possible. On the contrary, bad sampling technique reduces the convergence speed and may even make the Monte Carlo estimation never converge to an unbiased solution.

Therefore, the main problem for Monte Carlo methods is to find good sampling techniques. In this thesis, we mainly contribute in finding good sampling techniques for different type of light sources such as far-field (Section 3) and near-field (Section 4) as well as the combination of BRDF sampling (Section 5).

In the next section, we will introduce some rendering techniques based on Monte Carlo methods. Since our contributions are on sampling, all these rendering techniques can benefit from our work.

2.4 Monte Carlo Methods

During the last two decades, many physically-based rendering techniques based on Monte Carlo methods have been developed. Path tracing introduced by Kajiya [1986], is one of the most classical Monte Carlo technique to solve the light transport problem.

In this technique, random samples are rays. Issued from a camera (view point), these rays intersect geometries in the scene. For each intersection, a new direction is generated according to the BSDF. This method is accurate and unbiased, it is thus widely used as a reference solution. The resulting simulations accurately represent a large variety of lighting effects, such as soft shadows, depth of field, motion blur, and ambient occlusion. However, in most cases, this technique converges at the expense of a large computation cost. One

of its limitation is that it considers only rays from the view point, and thus gives a large priority to light paths from the camera. This limitation makes many indirect lighting effects such as caustics difficult to evaluate.

A decade later, bidirectional path tracing [Lafortune & Willems 1993; Veach & Guibas 1994] is introduced. By taking into account rays issued from the camera and the light sources, this technique is more robust than classical path tracing techniques, especially for indirect lighting cases. However, it is still costly and converges slowly due to the fact that the rays are only constructed with local choice of direction and thus their contribution may be small to the rendering equation.

In order to improve the efficiency with bidirectional path tracing, Metropolis Light Transport (MLT) [Veach & Guibas 1997] was introduced. After generating a set of initial paths with bidirectional path tracing, this technique mutates these paths by adding, deleting or replacing some vertices on them. Due to these perturbations, the paths are refined. The improved sample paths increase the convergence rate. Moreover, the mutations are designed to keep this technique unbiased.

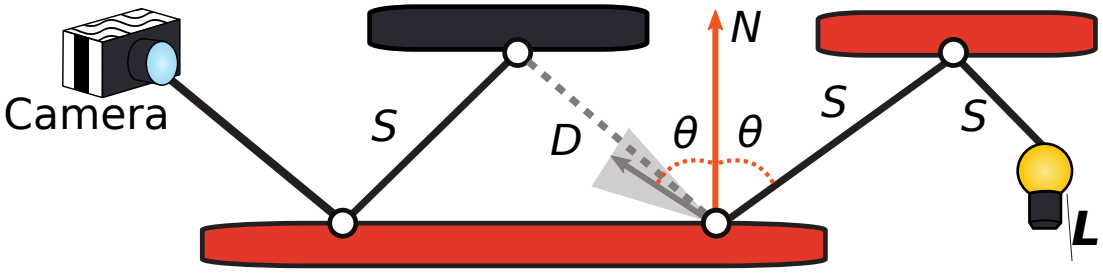


Figure 2.10: *A path that cannot be connected in an unbiased way. A S*DS path is constructed from the light source and the camera. Note that the gray paths: a dash line (sampled from diffuse surface) and a solid line (sampled from specular surface) cannot be joined together since the evaluation of the shading function is null. No further mutations can find a close path. Recently, Kaplanyan & Dachsbacher introduced a regularization (the gray triangle in the figure) to enlarge the possible connection area, so as to join the two paths together.*

However, as pointed by Veach & Guibas [1997], even if MLT improves the quality of generated images, there are still some special light effects that cannot be well handled by existing unbiased methods, such as reflected caustics, which is relies on specular-diffuse-specular paths (S*DS). In order to solve this problem, researchers have done a lot of work by explicitly searching specular paths [Jakob & Marschner 2012; Walter *et al.* 2009] or by implicitly handing the problem with consistent methods (e.g., with progressive photon mapping [Hachisuka & Jensen 2009; Hachisuka *et al.* 2008]). The latest one is the work of Kaplanyan & Dachsbacher [2013b] who introduced a regularization framework for refining such paths (cf. Figure 2.10). By introducing a small bias, it can capture reflected caustics effects. This work has been later extended to more complex path-space analysis [Schmidt

et al. 2013].

Figure 2.11: *Path-space gradients give a hint for sample distribution: (**Left**) Horizontal gradient image, (**Middle**) Vertical gradient image. The higher is the gradient, the brighter is the pixel. (**Right**) Sample distribution according to the gradient images, where red means high number of samples and blue means low number of samples.*

MLT needs many samples in smooth parts of the path space where radiance is high but does not vary much. For the human visual system, gradients are the regions of change and thus they are very important. Lehtinen *et al.* [2013] have introduced a method that approximates the gradients and use more samples in high gradient parts. Image gradients are approximated by computing a difference radiance between two neighboring pixels in path space. When estimating the radiance of pixel A, the radiance of its neighboring pixel B is approximated by using the modified light paths of pixel A. A shift function is introduced for MLT to perform this modification. Therefore each of samples corresponds to a pair of paths reaching neighboring pixels. As shown in Figure 2.11, path-space gradients give a hint for sample distribution. A coarse result is computed with distributed samples, then the final result is achieved by an image-space Poisson reconstruction.

One of the great advantage of ray-based physically-based rendering techniques is their simplicity. Real-time ray tracing is thus easily implemented. Purcell *et al.* [2002] presented a GPU solution for ray tracing. A lot of papers have followed to reach interactive ray-tracing with high quality acceleration structures. For example, with kd-tree acceleration structure, Shevtsov *et al.* [2007] have introduced an interactive ray-tracing for dynamic scenes on multi-core CPU by constructing KD-tree with bins efficiently. In 2008, Zhou *et al.* [2008] have presented a real-time kd-tree construction for ray-tracing for dynamic scenes on NVIDIA GPUs. For more general platforms, Lu *et al.* [2012] have introduced a real-time kd-tree based ray-tracing for dynamic scenes, on both AMD and NVIDIA GPUs by using OpenCL.

2.5 Multi-Pass Methods

The contributions of this thesis are related to the sampling process used in Monte Carlo methods. Therefore, multi-pass methods that include Monte Carlo methods as a subpart of the solution also benefit from our contribution. Monte Carlo methods have been combined

with finite element methods or density estimation methods, to provide multi-pass solutions in order to solve light transport problems. These combinations are often more efficient than path tracing techniques that use only Monte Carlo methods.

One of the well-known method using density estimation comes from [Heckbert 1990]. They have introduced an algorithm that treats diffuse and specular paths separately. In their paper, radiosity textures are constructed via shotting particles (photons) from light sources. For each particle, the impact on the geometries is stored with its carried energy and accumulated. This method is unbiased, however, it is slow and memory costly.

Shirley *et al.* [1995] have formalized the use of density estimation with a basic three-step framework for global illumination. Firstly, a step is performed in a similar way to [Heckbert 1990]: power carrying particles are emitted from each light source. A list of "hit points" is generated and stored. Secondly, an approximate irradiance function is constructed by using these "hit points". Finally, a meshing phase is performed at last for rendering. However, as shown in their later work [Walter *et al.* 1997b], this method was a proof-of-concept implementation and too limited for general use. So they presented several improvements to make this method more general and useful. This new partial tracing technique uses local linear density estimation, with improving color accuracy, BRDFs efficiency, and visual quality. The method is also unbiased, but still limited by the mesh size.

More recently, to balance both accuracy and efficiency, one of the most successful techniques is photon mapping [Jensen 1996b]. Based on two passes, the core idea is to sample independently different light paths: light paths for direct lighting, specular paths from light sources that create caustics, and more generic paths that contribute to the global indirect lighting are stored in different maps using an acceleration structure: kd-tree. Then, paths from the view point are generated and a local reconstruction from these maps is performed at their impacts. This reconstruction introduces some bias depending on the local reconstruction size: the bigger is the size, the larger is the bias.

To control bias, progressive photon mapping was introduced [Hachisuka *et al.* 2008]. By inversing the ray tracing step and shooting photons step, they introduced a progressive approach to reduce bias: for each iteration, this technique generates new samples and reduces the reconstruction size. Unfortunately, the convergence is slow, especially when rendering distributed ray tracing effects such as depth of field, motion blur and glossy reflections. Therefore, Hachisuka & Jensen [2009] improved their approach to support these effects by sharing photon statistics within regions to compute average radiance. Later on, Weiss & Grosch [2012] extended this approach for dynamic scenes. More recently, adaptive progressive photon mapping [Kaplanyan & Dachsbacher 2013a] was introduced to balance noise and bias in rendered images with an optimal bandwidth for each new added sample.

The optimal bandwidth is computed by minimizing an asymptotic mean square error, which consists of the variance of the Monte-Carlo estimation, and the bias as well as variance of the kernel estimator, which is used for estimating new added samples.

Other improvements have focused on speed by using GPU computing power. The process of photon mapping can be real-time by using dynamic kd-tree acceleration structure on graphics hardware [Zhou *et al.* 2008]. Recently, a more practical GPU real-time photon mapping technique [Mara *et al.* 2013] was introduced. This technique evaluates a hierarchy of designs for photon mapping techniques, and decomposes it into eight specific algorithms with different dimension of polygon bounds and acceleration structures.

Since radiosity and Monte Carlo approaches have their own advantages, combing them could theoretically deal with more general scenes. Some of these methods take the form of multi-pass methods, and use radiosity in the first step and ray tracing techniques to add specular paths, such as [Chen *et al.* 1991; Wallace *et al.* 1987]. It is also possible to combined them progressively, Granier *et al.* [2000] integrate particle tracing into the gather step of hierarchical radiosity, to produce rapidly smooth and noise free results with both diffuse and specular effects.

Monte Carlo methods can be used as an independent solution or a part of an combined solution for the light transport problem. Since our work contributes to the generation of rays for Monte Carlo methods, these solutions also benefit from our work. In the upcoming section, we will explain image-based light sources, which are 2D and 4D data sets, since our work focuses on how to sample them and to use them for shading with Monte Carlo methods.

2.6 Representations of Light Sources

Light sources are very important since they are one of the main factor in the rendering equation. If light equals to zero, the final color is black, no matter what kind of geometry or material are present in the scene. Light sources are expressed by some digital models. One type is user-defined analytical models such as directional, point and area light models. However, these light sources cannot directly represent realistic light sources and also hardly represent other complex lighting environment of virtual worlds (games or movies).

Generally, realistic light sources and complex lighting environment are too complicated to be expressed by simple analytical models. A direct capture is a better solution. However, captured light source are not directly usable, since the corresponding data is often much larger and thus increases significantly the computation cost. In order to use them, we can either project the captured data into representation models which are cheaper to compute, or select only part of them which have high contribution to the lighting. More precisely,

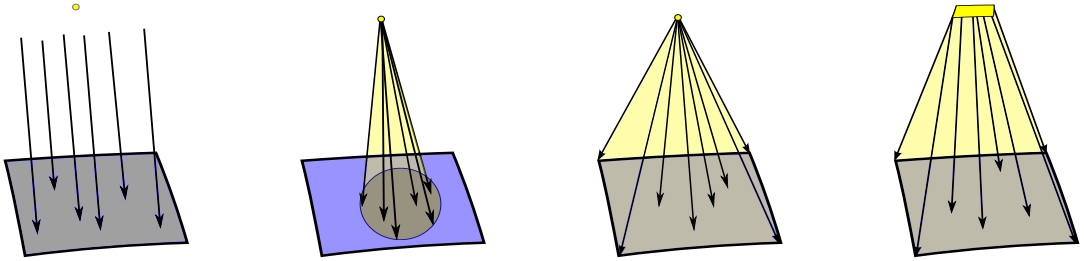


Figure 2.12: *From **Left** to **Right**, classical light sources: directional light source, spot light source, point light source and area light source. Rays (black line) are emitted by a light source (yellow) to a planar (blue). Lighting areas are highlighted with yellow.*

there are two categories of methods: one is composed of representations using classical basis (e.g., B-spine, Haar wavelet) with their respective coefficients. The other models decompose lighting environment into smaller and simpler lights (e.g., virtual point light sources).

2.6.1 Classical models

Some shapes have simple representations, such as point, triangle, disk and sphere. They are used as classical light models with the assumption that the radiance is uniform over their domains. The main advantage of classical light sources is their simplicity. First, because of their analytical representations, the lighting is generally defined without any approximation. Second, analytical solutions often lead to cheaper computation compared to numerical estimations.

Another advantage is their simplicity for scene designers who can use these light sources to configure 3D scenes in an intuitive way. The easy understanding of how they cast shadows, highlights, and other properties helps designers to implement their ideas. Therefore, these light sources are commonly included in designer tools (e.g., Autodesk® 3ds Max®), game editors (e.g., CRYENGINE® Free SDK), and game mods tools (e.g., Skyrim Creation Kit™).

The third advantage is that these light sources may be modified and further controlled intuitively to get the required shading effects, such as BendyLights [Kerr *et al.* 2010]. By curving light paths of a spot light model, this technique allows artists to control independently light direction and shadow position at different points in the scene.

The simplest classical lights are point and directional lights. As pointed by Verbeck & Greenberg [1984], the use of light sources must consider also the spatial variations: the distribution of luminous intensity varying with direction and geometry. Point light sources may thus be extended with goniometric diagrams for Computer Graphics. Nishita *et al.* [1985] introduced shading models for point and linear light sources that can produce umbrae for point lights and penumbrae for linear light sources with shadow volume. However, only the

diffuse component in this paper is analytical. Poulin & Amanatides [1991] reduced this limitation by also introducing analytical directions in specular component, but, they are still limited in terms of spatial variations of emitters.

Area light sources are more realistic than directional and point light sources since they produce soft shadows, which are quite common in our daily life. Picott [1992] introduced a realistic rendering method based on radiosity with linear and area light sources. In this paper, area light sources are treated as an infinite series of point light sources. During the last two decades, this logic has been widely used for Monte Carlo rendering dealing with area light sources: a set of point light samples are selected for estimating the lighting of the area [Pharr & Humphreys 2004]. Further approximations have also been made for interactive applications. For some shadow mapping techniques [Annen *et al.* 2008; Li *et al.* 2011; Ritschel *et al.* 2008], area light sources are treated as a finite set of point light sources from which shadow maps are built. Filtering is needed to reduce the artifacts caused by the approximation.

There are also other models such as spherical light sources [Amanatides 1984] and polygonal ones [Cohen & Greenberg 1985]. These models have less flexibility and are introduced for specific rendering techniques.

The disadvantage of these classical models is their fixed form, they are thus limited for representing more complex light sources. Especially in real world, almost none of the light sources perfectly matches these ideal models. One example is the sun. Even if the sun can be represented as a uniform sphere, its lighting arriving at the ground can be changed by many phenomena, such as occlusion of clouds and atmospheric effects. The resulting models are consequently more complex.

2.6.2 Basis Functions for Light Representations

It is very common to approximate complex measured data with coefficients in a given basis [Papoulis 1978]. Complex light sources are one of these instances. For decades, directional light sources have been commonly used. These light sources have only 2D directional variations and are distributed over the sphere. In order to represent them, using spherical basis is quite straightforward.

As shown by Ramamoorthi & Hanrahan [2001], environment maps can be projected to an analytic expression with spherical harmonics. These bases have been also proven to well approximate BRDFs [Kautz *et al.* 2002]. Spherical harmonics have four advantages. The first advantage is their orthonormality: these orthogonal basis reduce the rendering equation into a simple dot product, which leads to cheap computation cost. The second advantage is applicability: it is equivalent to Fourier basis, which means it fits for a large variety of complex data. The last one is rotation friendly: there is an analytical rotation solution for

spherical harmonics that allows cheap computation cost and flexible rotation at the same time. The appealing properties of these bases successfully lead to real-time applications in the game industry, such as simple real-time global illumination [Kaplanyan & Dachsbacher 2010], which uses low-order spherical harmonics to represent directional distribution of light.

However, spherical harmonics are only good at representing low-frequency signals. If the signals have high frequency or discontinuities, coefficients should be also high to fit, it may introduce instability and high computation cost. Therefore, this technique suits well low-frequency environment maps, not high frequency [Mahajan *et al.* 2008].

Another disadvantage is their global support: all functions are non-zero over whole sphere that may also cause instability when the number of coefficients is high. Furthermore, as Fourier basis, they hardly represent discontinuous local signals causing ringing artifacts (cf. Figure 2.14). Spherical harmonics can represent large area sources efficiently, but need a lot of coefficients to represent small sources which correspond to high frequency phenomena.

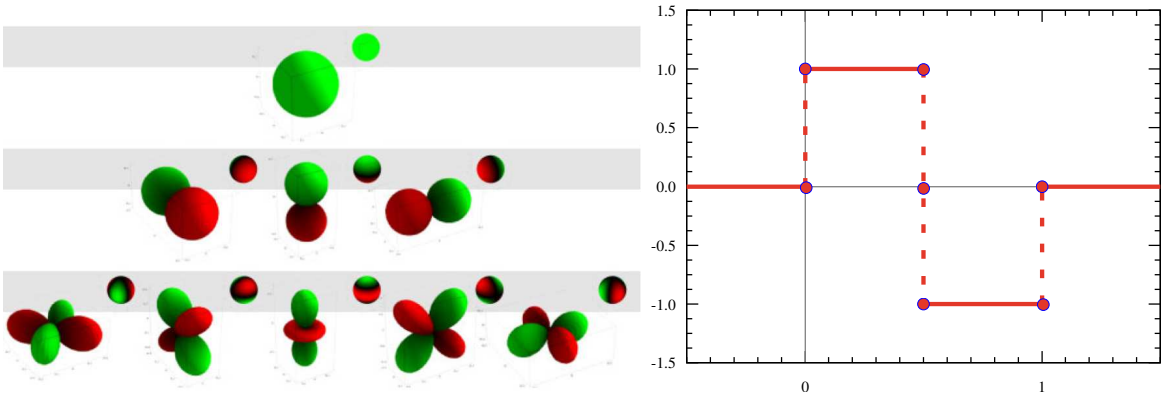


Figure 2.13: *Spherical basis and planar basis: (**Left**) spherical harmonics. Red portions represent regions where the function is positive, and green portions represent regions where the function is negative. (**Right**) Haar basis. Red solid lines represent the function.*

Some effects like shadows require even more coefficients to be represented. For example, for diffuse shading, irradiance is insensitive to high-frequency, the shadow still suffered from low-frequency representation. In order to solve this problem, Ng *et al.* [2003] chose Haar wavelets. The reason why Haar wavelets is reliable in expressing higher frequency is that, wavelets provide locality in both spatial and frequency space.

The theoretical advantage of Haar basis is that, it can represent high-frequency details for captured environment maps with only few coefficients. Furthermore, their compact supports and the sparsity of the coefficients increase the rendering performance. Although the rotation of Haar basis is more costly (it needs precomputed matrices [Wang *et al.* 2006a] and huge memory cost), it still retains other advantages: orthogonality and all-frequency

signals friendly representation.

However, the main problem of Haar wavelets is their planar form. As pointed by Lessig & Fiume [2008], when using planar wavelets to represent spherical signals, parametrization artifacts are inevitable when the sphere is mapped onto a planar domain. Spherical wavelets [Schröder & Sweldens 1995] is a better choice for this problem.

Spherical wavelets are defined on the unit sphere. Like Haar wavelets, they also have appealing properties: frequency, space and localization. However, they are limited in their efficiency when approximating and processing all-frequency signals [Lessig & Fiume 2008; Tsai & Shih 2006].

Lessig & Fiume [2008] introduced spherical Haar wavelet basis and then extended to the ball [Chow 2010] for the purpose of combining all appealing properties from wavelets and spherical basis. This basis is orthogonal, symmetric and compact. Compared to Haar wavelets, their basis is defined on the unit sphere, it is thus free of distortion. However, as noted by the Chow [2010], a required basis without constraints may not exist.

Tsai & Shih [2006] introduced Spherical Radial Basis Functions (SRBF). This basis is a more general and more efficient spherical basis function compared to spherical wavelets (cf. Figure 2.14). It has nearly the same properties as spherical wavelets that it localized in both frequency and space and thus represent all-frequency signals efficiently. However, as pointed by Lessig & Fiume [2008], projecting a signal to SRBF is expensive and processing a signal is difficult.

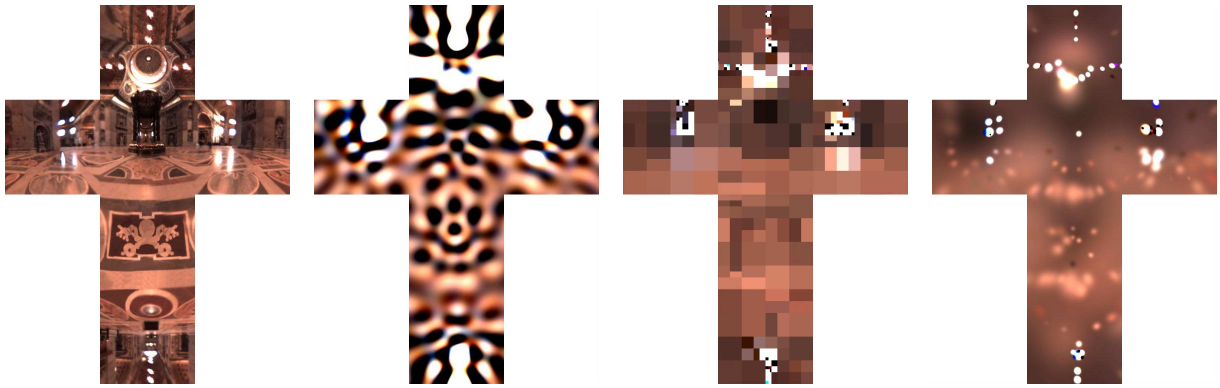


Figure 2.14: *Comparison of the projection of the St. Peter's Basilica HDR lighting environment using (from **Left** to **Right**), spherical harmonics, area-weighted wavelet and scattered spherical radial basis functions with 972 coefficients or parameters (from the paper [Tsai & Shih 2006]).*

These well-known bases are orthogonal bases (SH and Haar) or exponential bases (SRBF), which leads to compute the product in the rendering equation for light transport efficiently. During the last decade, precomputed radiance transfer (PRT) has been commonly used (e.g., [Hašan *et al.* 2006; Liu *et al.* 2004; Sloan *et al.* 2002a; Tsai & Shih 2006]). These

techniques precompute for each rendering point a transfer matrix with arbitrary granularity (such as including direct transport, indirect transport, BRDF and geometries) and store this information into approximated forms such as spherical harmonics or wavelets. The same idea is used in precomputed light transport (PLT) that decomposed BRDF from the transfer matrix as well [Green *et al.* 2006; Ng *et al.* 2004; Sun & Ramamoorthi 2009]. The light sources and transfer matrices (or BRDF and transport matrix) are always using the same basis function. At run-time rendering step (mainly relighting), radiance can be computed by instancing the representative view directions, which are computed by a simple dot product between light source and transfer matrix (or BRDF and transport matrix). Obviously, the biggest limitation is the heavy constraints for dynamic scenes that even only supports static scenes, since the transfer matrix is stored for each vertex. By targeting only coarse-scale and distant indirect lighting, Modular Radiance Transfer (MRT) [Loos *et al.* 2011] evaluates indirect lighting for dynamic scenes efficiently. However, since they perform all computations in low-dimensional subspaces, errors are inevitable. A user interaction is needed to represent the rendering scene with cubes which radiance transfer is precomputed between. The objects in the cubes will be ignored, therefore shadows and reflections with clutters are lacking. Loos *et al.* [2012] then reduced this problem by introducing three additional operators to capture these related propagations. However, the accuracy is still limited by the cubes, and the performance depends on the complexity of the representation of the clutters. Similar idea is applied to render materials with repeated structures [Zhao *et al.* 2013]. The repeated structures is similar to the cubes of MRT. Since other smaller structures inside are invisible, a limitation of MRT is avoided. Furthermore, since these repeated structures are many and simple, modular combination of these "cubes" is effective and efficient.

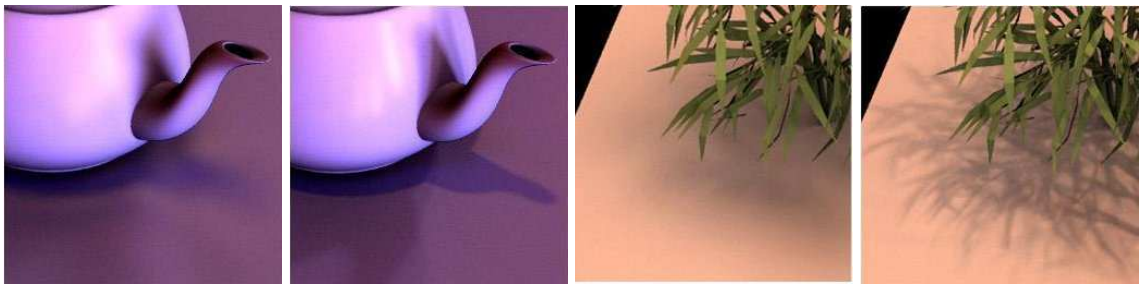


Figure 2.15: *precomputed light transport with different basis: spherical harmonics vs wavelets with 200 coefficients.*

As pointed by Ng *et al.* [2004], when using Haar wavelet basis with classical PRT, the view is fixed and this only suits for Lambertian surfaces and not for glossy materials. In order to solve these problems, they introduced a triple product wavelet method, by building analytical models for BRDF and lighting as well as visibility, it can render rich light effects with changing view. Liu *et al.* [2004] introduced a new method to remove the fixed view limitation, by combining BRDF factorization and Haar wavelets together, the method

achieves interactive rendering with shadows and all-frequency lighting for glossy surfaces. Tsai & Shih [2006] speed up the rendering by using SRBF and Clustered Tensor Approximation instead of Haar basis.

However, for all the representations, the coefficients need to be precomputed and the precomputation is expensive. Although, the light source is encoded with bases and allows a bit manipulation, it still cannot represent arbitrary dynamic light sources that are changing over time. Thus precomputation and preprocessing for each frame are required, and making the method too time consuming and memory expensive for practical use [Havran *et al.* 2005].

2.6.3 Many-Light Models

Incoming radiance with complex light sources can be approximated by a simple accumulation of the incoming radiance from many simpler light sources, such as virtual point light sources (VPL) [Keller 1997], virtual directional light sources (VDL) [Walter *et al.* 1997a] or virtual spherical light sources (VSL) [Hašan *et al.* 2009]. These techniques are known as Many-Light models [Dachsbacher *et al.* 2013]. Similar to basis functions, these simple light sources follow the same idea to use a simpler model for approximating complex data with some decompositions such as coefficients or trees. In this context, VPL, VDL and VSL are also a type of "basis" with differences in discretization and instantiation.

Instant radiosity [Keller 1997] approximates the global illumination problem with a number of VPLs. Rather than deriving a complete analytical model, a many-light model approach is discrete and can be used directly as long as the model built. The direct lighting is computed from the set of VPLs. For indirect lighting, new VPLs are generated for each lighting bounce and used similarly. This method has some advantages. First, it provides a unified mathematical framework for global illumination that includes direct lighting and indirect lighting. Second, it makes the approach scalable, the quality and the speed can be easily balanced with the same algorithm.

For image-based lighting, light sources are stored in images: a many-light model can be easily driven by selecting some texels on the images. For direct lighting, many-light model can be unbiased, the building of such model is the same as generating samples from the image-based light sources. For indirect lighting, virtual light sources are always created under approximations. The bias introduced by the approximation accumulates while indirect bounce increases. However, by using sufficiently virtual light sources and a highly scalable evaluation algorithm, the bias can be reduced to negligible level. Since the focus of this thesis is on sampling light sources, indirect lighting or global illumination content will not be discussed here, the reader can find more information in the paper [Dachsbacher *et al.* 2013].

Biased Many-Light methods can also produce images with good visual quality. By using a few virtual sources as a coarse approximation, these methods can produce nice results in a few seconds [Ritschel *et al.* 2011].

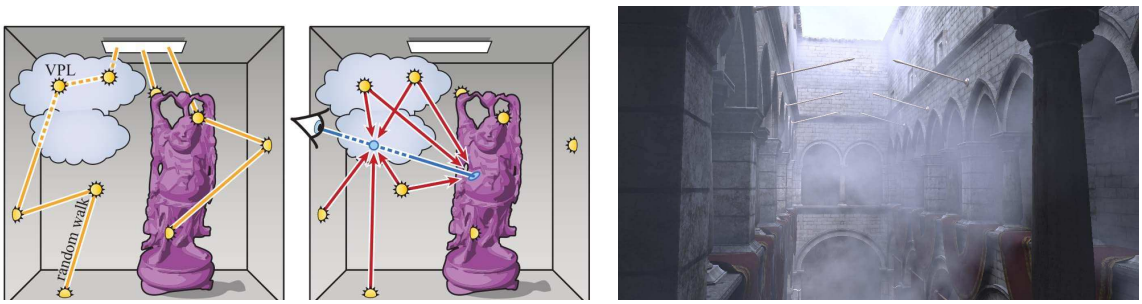


Figure 2.16: ***(Left)*** *Two passes of many-light algorithms: first distribute VPLs for each bounce and then use them to illuminate.* ***(Right)*** *The image took 52 minutes to render and demonstrates many-light methods with participating media [Engelhardt* et al. *2009].*

In general, each VPL does not contribute equally, some of them have low importance because they contribute very little to a region of interest. Scalable algorithm tries to exploit important VPLs and unimportant ones to reduce computation cost. In order to do this, some methods have tried to cluster unimportant VPLs together (e.g., Lightcuts [Walter *et al.* 2005a]). After generating a set of VPLs, a light tree is built: VPLs are stored in leaves, other nodes are represented as hierarchical clusters. A cut is defined as a cut of the light tree: a set of nodes that partitions the lights into equivalent VPLs. All nodes and leaves on this cut are used for rendering, the others will be omitted. The main issue is how to find a cut. In the original paper [Walter *et al.* 2006], a cut is generated to keep the expected error under a threshold. One year later, multidimensional lightcuts [Walter *et al.* 2006] extend the domain of clusters and cuts to include receiving points as well as lights to handle motion blur and participating media with scalable performance. In 2012, bidirectional lightcuts [Walter *et al.* 2012] were introduced as well as related weighting strategies. By combining bidirectional path tracing with VPL, this method can reduce the bias and extend the generality of lightcuts to more complex materials and lighting effects.

The many light problem can be also interpreted as a matrix problem. Matrix row-column sampling [Hašan *et al.* 2007] samples sparsely rows and columns of the lighting matrix to reconstruct the image (rows represents shading points and columns represents lights). This method is also extended by the paper [Hašan *et al.* 2009] and [Davidovič *et al.* 2010] to better support glossy materials. In 2011, Ou & Pellacini [2011] improved the idea by discovering and exploring the global as well as local matrix structure since they found that the matrices of local surfaces always have low ranks. This improvement leads to a speedup to the previous algorithms.

Large collection of VPLs makes big computation penalty. Although the clustering or

gathering process is accurate for a static frame, they cannot be directly applied to animated applications since it is too costly to repeat the same process for each frame, flickering may also occur. To solve this problem, previous work tried to find coherence between frames and keep as many as possible VPLs instead of regenerating VPLs for each frame (e.g., [Laine *et al.* 2007]). Hašan *et al.* [2008]introduced a tensor formulation of the animated many-light problem. For each frame and for each pixel, each element of a tensor expresses the contribution of a light. Similar to the matrix row-column sampling [Hašan *et al.* 2007], they also extended it to handle temporal coherence explicitly, so as to reduce flickering.

VPLs have two inherent limitations. When connecting the camera to a VPL, the BRDF is much larger than the probability density of generating the path. This limitation leads to spikes [Hašan *et al.* 2009; Kollig & Keller 2004]. The closer a VPL is, the more the artifact will be. Clamping the VPLs is a straightforward way but it causes energy lose, which means bias. Kollig & Keller [2004] introduced an unbiased path tracing approach to compensate for the loss of energy. However, large computation penalty was also introduced. The other limitation is the difficult path (cf. Section 2.4). When glossy materials exist in the scene, S*DS paths are inevitable since the paths are connected from a surface sample to a VPL. The main reason for these difficult paths is that the probability of generating a proper path is too small (cf. Section 2.4). In order to solve this problem, intuitively, we can increase the size of the path [Kaplanyan & Dachsbacher 2013b] or increase the size of the light [Hašan *et al.* 2009]. Of course, bias is also introduced.

Many-Light models yield good results at balancing the quality and the speed. However, the previous methods that select VPLs and cluster them for each frame is still too costly. The methods of finding coherence between frames may not prevent generating too many new VPLs when the distribution of the light sources is changing significantly. For example, light sources are switched from off to on, or even repeatedly (e.g., a dance hall). For another example, light sources are blocked over and over (e.g., the sun blocks by the clouds or leafs). Therefore, these methods are limited to dynamic light sources that are changing over time such as an environment map stream.

2.6.4 Conclusion

Classical analytic light models are simple and friendly to use for artists. However, they are limited to representing complex light sources, such as captured light sources.

Complex light sources can be approximated by many simpler entities, which could be basis functions or classical light models. Basis functions are good at representing all-frequency light sources, but too expensive for dynamic light sources which are changing over time. Precomputed radiance transfer and precomputed light transport are very efficient for relighting step due to the basis representations they use. However, they have big con-

strains for dynamic scenes that makes these techniques away from interactive applications that include dynamic objects and lighting. Most of basis-based light source representations are limited to expressing only 2D directional variant, not spatial variant. Therefore precomputed radiance transfer and precomputed light transport are also limited to distant lighting only.

Many-Light models can provide unbiased methods or biased methods with artifact free images. Moreover, the unified framework of many light methods leads scalable solutions, from online best performance to offline best quality. However, the generation and clustering of virtual light sources are quite expensive for per-frame dynamic environment light sources.

In this thesis, we focus on improving many light methods for representing dynamic environment light sources (in Chapter 3), we also focus on improving many light methods for representing 4D light sources with basis functions (in Chapter 4).

2.7 Acquired Light Sources

2.7.1 Image Based Lighting

As we discussed before, for realistic rendering, it is natural to use captured and thus realistic light sources. Compared with classical light sources (cf. Section 2.6.1), which are mainly designed with analytical models, realistic light sources are measured with some dedicated devices (e.g., [Ashdown & Rykowski 1997; Goesele *et al.* 2003; Mas *et al.* 2008; Verbeck & Greenberg 1984]). These acquired light sources are generally stored in images. In order to better understand how to use such light sources for shading, we first introduce image-based lighting techniques.

Direction / Space	0 Dimension	1 Dimension	2 Dimension
0 Dimension	Point Light	Linear Light	Environment Map
1 Dimension	Linear Light	Linear Light + 1D GD	Linear light + 2D GD
2 Dimension	Area Light	Area Light + 1D GD	Light Field light
GD: Goniometric Diagram			

Table 2.1: *Light source models with different dimension of variations.*

By using measured light sources, level of realism of the scene and the visual interest are increased: the shadows, reflections, and shading all exhibit complexities and subtleties that are realistic and consistent. Figure 2.17 illustrates a comparison between using classical area light sources and measured image-based light sources for the same scene.

Figure 2.17: *Lighting with images: A scene illuminated with (**Right**) three traditional area light sources. The same scene illuminated with (**Left**) a HDR image-based lighting environments inside a cathedral. Note that, even with the same geometry and BRDF, the illumination results are very different due to the light sources. Images-based lighting yields a more versatile result.*

In general, a light source contains 4D information: both 2D directional variations and 2D spatial variations. For simplicity we first focus on light sources with only 2D directional variations . More general cases will be discussed in Section 2.7.3.

Low Dynamic Range (LDR) is commonly used for direct display. However it is limited to represent the visible radiance, which is much larger than measurement or display devices generally.

In order to solve this problem, High Dynamic Range (HDR) [Reinhard *et al.* 2010] has been introduced. Instead of recording what can be directly displayed, HDR images record a much wider range of values closer to real radiance. By using HDR space, the representation is more precise but a final tone mapping process is needed to map a part of recored radiance to a specific range of luminance that can be directly display by LDR devices. Therefore, with different tone mapping, a HDR image can express many LDR images in different range of luminance.

However, because a radiance value (in HDR) will be mapped to different values (in LDR), HDR encoding needs larger precision to preserve the quality.

2.7.2 Environment Map Light Sources

Distant light sources are well represented with only 2D directional information in the local spherical domain and can therefore be stored on 2D images directly.

2D directional distant lighting can be easily captured using different devices (cf. Figure 2.18). The simplest way is to use a camera to take a picture from a mirror sphere (cf. left image in Figure 2.18). Although, an ideal mirror sphere is hard to obtain, the device should be as specular as possible in order to reflect radiance from all possible directions

Figure 2.18: *Capture devices: (**Left**) Environment map can be captured using a specular light probe, the smoother is the sphere, the better is the quality, or (**Middle**) use 360 degree 5 lens camera called lady bug camera. (**Right**) Environment maps can be captured using fisheye cameras as well [Stumpfel* et al. *2004]*

.

Figure 2.19: *Three different views of a captured HDR environment map. The HDR image is combined with a series images which are captured by a digital camera in one-stop exposure increments from 0.25 second to 0.0001 second. The environment is displayed at three exposure levels (-0, -3.5, and -7.0 stops): relative luminance (100%, 2.14%, 0.046%) (images from [Debevec 1998]).*

into the camera. Furthermore, the selected shape has to be as convex as possible to avoid self-reflections. With such a proper device, radiance from full set of incoming directions is captured into an image. Since the captured probe represents a environment and will be used as is, the probe is generally called environment map.

Range of reflected radiance is often larger than what a camera can capture. In order to capture HDR environment map, one can compose a HDR picture by capturing multiple pictures with different exposure and combine them [Debevec 1998]. As shown in Figure 2.19, different exposure is able to capture full dynamic range lighting environment from real world. The radiance of a pixel on the probe corresponds to the incoming radiance from the reflected direction.

Since we use the captured sky data from [Stumpfel *et al.* 2004] for our work, we give

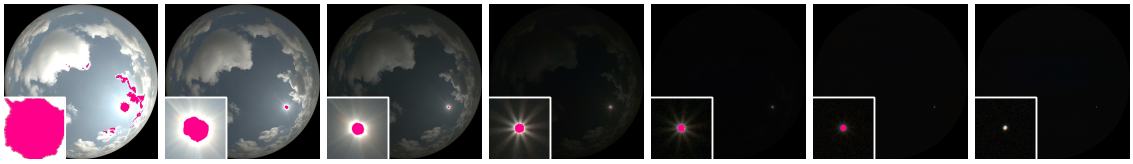


Figure 2.20: *HDR sequence and camera settings. Captured sky using HDR camera with different exposure settings. A detailed view of the sun is shown in the bottom left of each image; pink regions indicate saturated pixels. The darkest image is the only image that does not saturate for the sun (images from [Stumpfel* et al. *2004]).*

an overview of this method. Sky as a hemisphere can be captured using a fisheye lens. However, there are two remaining problems. First, the sky is a extreme dynamic range spectrum: the sun can be over five orders of magnitude brighter than the sky and clouds. Second, the absolute intensity of the sun is much brighter than cameras are designed to capture. In their paper, Stumpfel *et al.* [2004] selected carefully exposure times, aperture and neutral density filters. By using a calibration procedure with optimized settings for these three mechanisms, they are able to capture the full dynamic range of the sky using several exposures (cf. Figure 2.20). They captured the fully dynamic HDR images of the sky from sunrise to sunset. This data set is a HDR sequence of the sky with sun in real world, we use this data set for two of our projects.

Figure 2.21: *A rendering application in the paper [Stumpfel* et al. *2004]. Frames at: 7:04am, 10:35am, 4:11pm and 5:37pm on the same day.*

Cube maps are widely supported by different graphic hardwares and compatible with texture arrays. For the performance requirement, the captured environment probes are usually converted to cube maps before being used on GPUs. A cube map can be constructed from a probe with predefined directions: six faces of the cube map represents the lighting distribution in six directions (+X, -X, +Y, -Y, -Z, +Z) of the Cartesian coordinate system. Depending on the parametrization of the predefined directions, a light probe leads to different cube maps (a light probe and one of its corresponding cube map are shown in Figure 2.22).

Although cube maps are 2D textures with uniform texels, the direction distribution stored in each texel is often not uniform in the spherical domain. The pixels closer to the boundaries have larger directional variations, which leads to more density at the boundaries. As shown in Figure 2.23, the change of parametrization from cubic to spherical causes distortion, if the solid angle of each texel are not calculated properly.

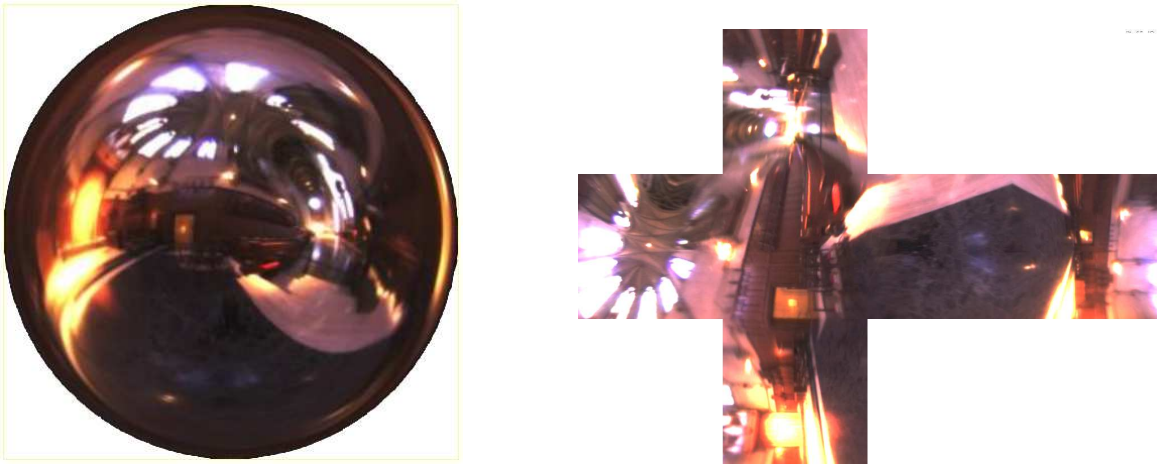


Figure 2.22: *Environment map: Captured environment map with light probe (**Left**). The corresponding cube map (**Right**). From* `http://www.pauldebevec.com/Probes/`

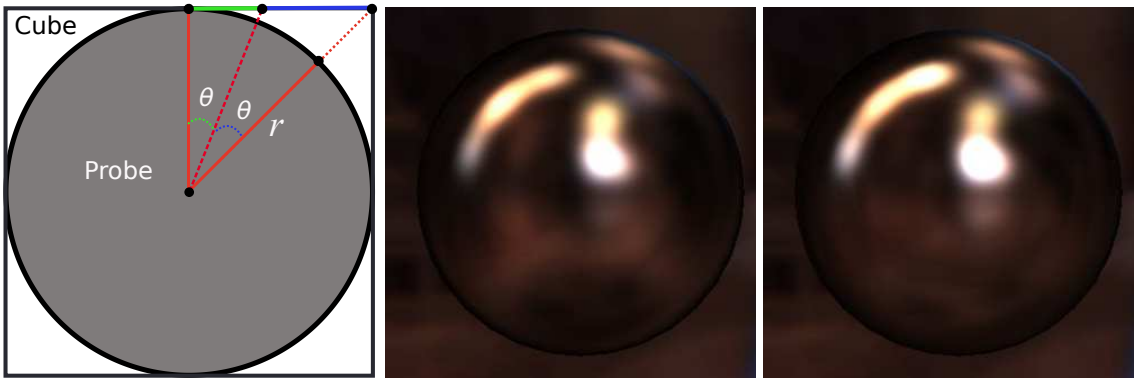


Figure 2.23: *Non-uniform directional information: Mapping directional distribution from a cube map to a sphere(**Left**) Using environment as uniform 2D textures (**Middle**) distorts the lighting directions. The correct usage with solid angle (**Right**).*

As illustrated in the left image of Figure 2.23, for the same solid angle θ, the size of the two surfaces areas (dashed green and blue) on the probe is the same, but the size of the corresponding areas (solid green and blue) on the cube is different (green line is shorter than blue line). Therefore, if we compute solid angles uniformly, the blue parts will have higher density than green parts in directional distribution. In order to make them properly, a solution of adjusting intensity is needed to change the distribution according to the solid angle. In Section 3.3.3, we discuss this in details.

2.7.3 Light Field Light Source

Light field light sources [Gortler *et al.* 1996] reduce a 4D lighting problem to an array of 2D image problems. Inspired by flies' eyes, light fields are designed as an array of 2D images that captured at constant offsets or rotations of lens (cf. Figure 2.24). Once light fields are acquired, emitted radiance is recorded in 2D images for predefined view points

and projection directions. Radiance between these view directions can be reconstructed by interpolation among neighboring images (cf. Figure 2.25). Since the radiance is cached in texels and depth information is discretized on neighboring images. the reconstruction quality depends on the resolution of the images as well as the number of images. Moreover, rather than a computation power hungry problem, rendering using light fields are more likely a bandwidth hungry problem that happens on data transfer between different computation units, different levels of cache as well as memory, and different PCBs. This is one of the main problem that we face to in this thesis.

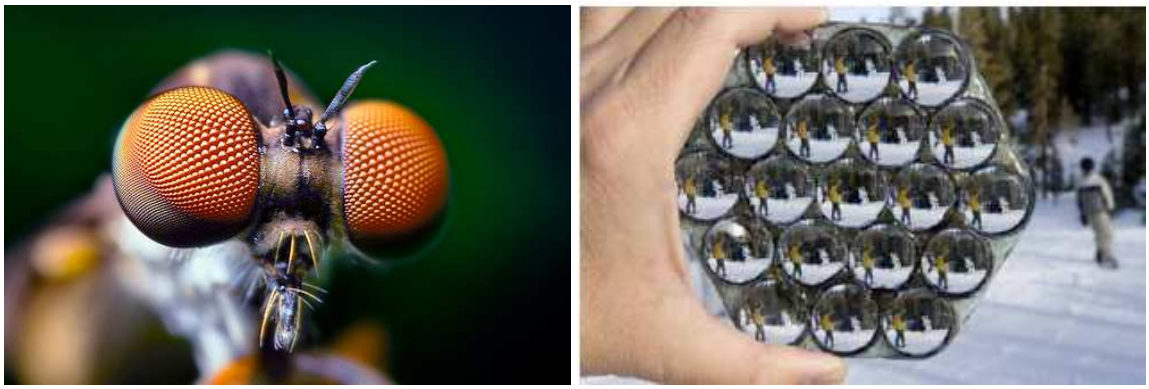


Figure 2.24: *Fly eyes (by Thomas Shahan) and LightField Camera (Adobe Magic Lens).*

One advantage of light field based techniques is its container: images. Like other image-based techniques, the versatility of the real world can be captured without knowing physical rules behind. Since the recorded radiance is already "computed" by the real world, it can be directly displayed like simple photographies. This property leads to realistic rendering applications for very complex scenes (cf. [Kim *et al.* 2013]).

Goesele *et al.* [2003] introduced an different plane parametrization to the light field, this plane is composed with uniform 2D bases where each basis has a corresponding image on the light field. A set of directions is reconstructed by a pair of planes. Since one of our work (cf. Chapter 4) is based on this configuration, we will introduce it in the upcoming content.

Measuring the light field of luminaires Light sources that have directional variations can produce far-field lighting, whereas both directional and spatial variations lead to near-field effects. Near-field photometry [Ashdown 1993, 1995] has been introduced to capture both near-field and far-field illumination. A digital camera is placed on a robot arm to capture the lighting from a light source by moving the camera on a surrounding sphere or hemisphere, then an imaging sensor (e.g., a CCD chip) is used to record the radiance. Similar setups [Jenkins & Mönch 2000; Rykowski & Wooley 1997] and a modification of the setup by replacing the camera lens with a pinhole [Siegel & Stock 1996] have been

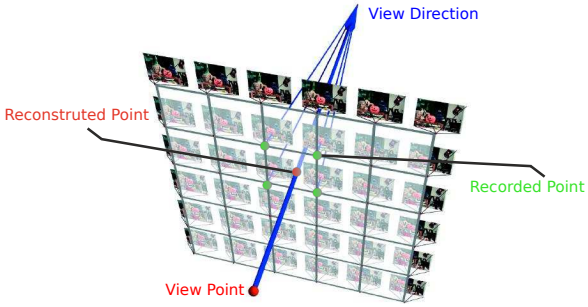


Figure 2.25: *Light field rendering from [McMillan & Gortler 1999].*

The nodes of transparent grids are 2D images. Red point is a view point whereas blue arrow is a view direction. Radiance is recored in the texels of these 2D images during acquisition of the light field. Non-recored radiance of a new ray is reconstructed by its corresponding texels of nearest images. Therefore, it is possible to reconstruct a new image by the interpolating its neighbor 2D images.

used to measure the light field of luminaires. As noted by [Halle 1994], the light source may produce arbitrarily high spatial frequencies, and thus introduce aliasing on imaging. As a consequence, a low-pass filter is needed to reduce the artifacts. This problem was also noted by [Levoy & Hanrahan 1996]. They tried to solve this issue by using the finite aperture of the camera lens as a low-pass filter.

However, as pointed by Goesele *et al.* [2003], since the aperture size must be equal to the size of a sample on the camera plane, this is a side effect of the lens system. In order to improve this, they built an acquisition system where the 4D near-field is projected on a predefined basis, leading to a priori control of the model accuracy. Since one of our main contribution is based on this model, we will introduce it in details. The basis plane can do the filtering to reduce aliasing and it can be specifically designed for a particular sampling scheme. In the paper, they introduced two possible setups to measure the light field of luminaires. As illustrated by Figure 2.26, One can capture the data by either making the camera static and moving the light source, or making the light source static and moving the camera.

Parametrization of the two plane setup A representation light source model is built after the measurement. This model is equivalent to a basis function representation model, where filters record bases and images record coefficients. Compared to classical representation models (cf. Section 2.6.1), the main difference is that the coefficients of this model are not constant. These coefficients are actually another representation model that are composed with box basis functions and constant coefficients. To clarify this parametrization, we rename the sampling plane as the filter plane and the measurement plane as the image plane (We will keep these two names in the rest of the thesis). Then the two-plane setup can be illustrated as Figure 2.27.

Similarly to any other light field models, it approximates the radiance $L(\boldsymbol{u} \rightarrow \boldsymbol{s})$ emitted

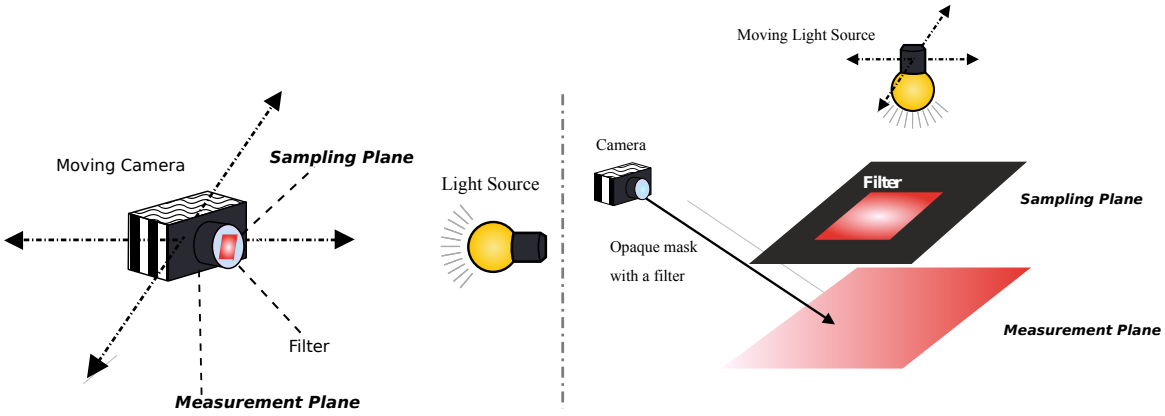


Figure 2.26: *Two-plane setup introduced by [Goesele* et al. *2003], (**Left**): The camera lens is replaced by a filter and the light field is directly projected onto the imaging sensor inside the camera. In this setup, the measurement plane is the imaging sensor. (**Right**): The filter projects an image on a diffuse reflector which is captured by a camera. In this setup, the measurement plane is the diffuse reflector plane (large red one). Note that, for both setups, the sampling plane is where the filter is placed.*

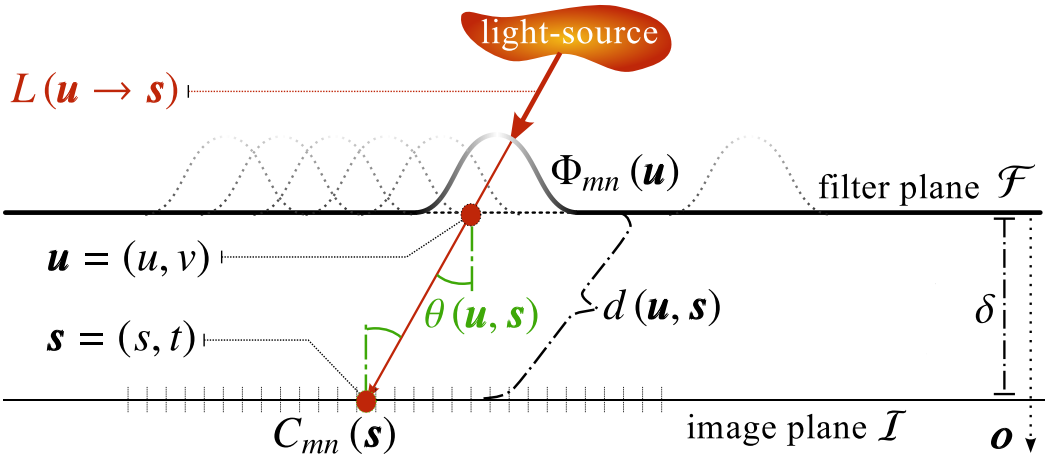


Figure 2.27: *Model parameterization. The 4D space of rays emitted from the light source is parameterized by a position $\boldsymbol{u}$ on a plane $\mathcal{F}$, called filter plane and supporting the reconstruction basis functions $\{\Phi_{mn}, m = 1..M, n = 1..N\}$, and a position $\boldsymbol{s}$ on a plane $\mathcal{I}$, called image plane supporting the images $\{C_{mn}(\boldsymbol{s}), m = 1..M, n = 1..N\}$. $d(\boldsymbol{u}, \boldsymbol{s})$ is the distance between these two points and $\theta(\boldsymbol{u}, \boldsymbol{s})$ the angle between the ray $\boldsymbol{u} \rightarrow \boldsymbol{s}$ and $\boldsymbol{o}$, the setup axis. δ is the inter-plane distance.*

by a luminaire by a weighted sum of images:

$$\widetilde{L}(\boldsymbol{u} \rightarrow \boldsymbol{s}) = \frac{d^2(\boldsymbol{u}, \boldsymbol{s})}{\cos^2 \theta(\boldsymbol{u}, \boldsymbol{s})} \sum_{m,n} C_{mn}(\boldsymbol{s})\ \Phi_{mn}(\boldsymbol{u}) \tag{2.18}$$

Φ is the reconstruction basis on the filter plane and C is the image on the image plane. A ray traces from the filter plane to image plane and intersects with the two planes at $\boldsymbol{u}$ and $\boldsymbol{s}$ respectively. $d(\boldsymbol{u}, \boldsymbol{s})$ is the distance between these two points and $\theta(\boldsymbol{u}, \boldsymbol{s})$ the angle between

the ray $\boldsymbol{u} \rightarrow \boldsymbol{s}$ and $\boldsymbol{o}$.

For rendering, the original method introduced in [Goesele *et al.* 2003] is based on a simulation on CPU. Although, the authors have shown that their model can be also used in an interactive manner using GPU [Granier *et al.* 2003], their dedicated approach was still limited to relatively small data set while introducing large approximations.

Data The available data we get from [Goesele *et al.* 2003] uses quadratic basis on the filter plane and $N \times M$ images with 300×300 resolution on the image plane.

The main advantage of this model is that images $C_{mn}(\boldsymbol{s})$ are directly acquired by using optical filters $\{\Phi^{\star}_{mn}\}$ designed to be dual functions of the reconstruction basis $\{\Phi_{mn}\}$. The images on the image plane measure the irradiance for each $\boldsymbol{s}$. It is defined as the integral of the lighting passing through the filter Φ on the filter planes,

$$E^{\star}_{mn}(\boldsymbol{s}) = \iint E(\boldsymbol{u} \rightarrow \boldsymbol{s})\, \Phi^{\star}_{mn}(\boldsymbol{u})\, \mathrm{d}\boldsymbol{u} = \langle E, \Phi^{\star}_{mn} \rangle \tag{2.19}$$

where $\langle , \rangle$ is the classical dot product in function space and $E(\boldsymbol{u} \rightarrow \boldsymbol{s})$ is the differential irradiance

$$E(\boldsymbol{u} \rightarrow \boldsymbol{s}) = L(\boldsymbol{u} \rightarrow \boldsymbol{s}) \frac{\cos^2 \theta(\boldsymbol{u}, \boldsymbol{s})}{d^2(\boldsymbol{u}, \boldsymbol{s})}. \tag{2.20}$$

If $L = \widetilde{L}$, Goesele et al. have shown that $E^{\star}_{mn} = C_{mn}$.

2.7.4 Conclusion

Rendering techniques have significantly improved qualitatively, to obtain realistic and plausible solutions and quantitatively to simulate physical phenomena. One of the reasons for this improvement is the introduction of using realistic light sources.

Realistic light sources are hardly approximated by uniformly emitting surfaces. They are usually measured directly by dedicated devices. In general, cameras are used for the measurement and thus images have become the main container for storing light source data. Furthermore, in order to capture the wide range of radiance in real world, HDR images are always needed.

Environment maps are appropriate for storing distant environment lighting, which cause far-field effects. They can be captured by HDR cameras taking pictures from a mirror sphere or by directly capturing radiance coming from the environment with dedicated devices. Unbiased Monte Carlo estimations need to sample the environment maps before lighting computation. Currently, generating good samples for the estimation is still too costly. As a consequence, sampling and lighting cannot be both done in real-time. Therefore, applications that capturing realistic lighting and use it to lit virtual objects are still not

applicable.

Near-field lighting can be measured by two-plane light field devices. Since it is a 4D representation, using them for rendering is very costly and too slow to interactive applications.

Chapter 3

FAR-FIELD ILLUMINATION: ENVIRONMENT MAP LIGHTS

In this chapter, we introduce our work on improving sampling technique for far-field illumination. This is a simple and effective technique for light-based importance sampling of dynamic environment maps based on the formalism of Multiple Importance Sampling (MIS). The core idea is to balance per pixel the number of samples selected on each cube map face according to a quick and conservative evaluation of the lighting contribution: this increases the number of effective samples. In order to be suitable for dynamically generated or captured HDR environment maps, everything is computed on the fly for each frame without any global preprocessing. Our results illustrate that the low number of required samples combined with a full-GPU implementation lead to real time performance with improved visual quality. Finally, we illustrate that our MIS formalism can be easily extended with other strategies such as BRDF importance sampling. This work has been published in Eurographics 2013 as a short paper [Lu *et al.* 2013b].

3.1 Motivation

Since the seminal work Debevec and Malik [1997], a lot of researches in rendering have been devoted to use acquired HDR environment maps efficiently. Environment maps participate, as natural light sources, to the realism of synthetic images.

As we discussed in the previous work section, Unbiased technique and realistic light sources are ideal provisions for realistic rendering. The logic of our motivation is shown in Figure 3.1. On the left side of the figure, it is a captured sky light [Stumpfel *et al.* 2004]. We want to use this kind of light sources to lit objects of the virtual world with unbiased techniques in real time (cf. right side of Figure 3.1). Furthermore, realistic lighting in real world is often dynamic. In general, the lighting distribution of the sky changes over time (e.g., when clouds appear). We also want to make these changes impact the lighting of

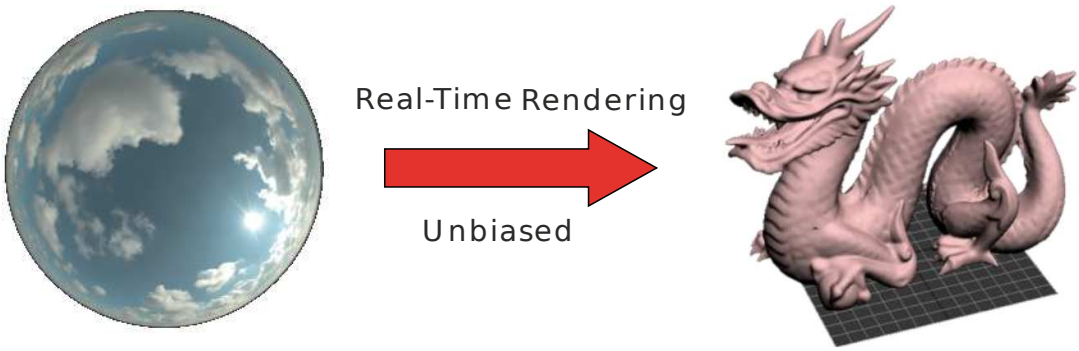


Figure 3.1: *The goal is to achieve real-time shading with realistic distant lighting during its capturing.*

the virtual world. In this logic, environment maps are dynamic like a real time stream (cf. Figure 3.2). A possible application would be to let people capture the light source and use it directly for shading at the same time.

For this purpose, the two steps, sampling and rendering should both be done in real time. The main problem is how to balance between the speed and the quality of the sampling. The objective is that the speed of sampling should be fast and the quality of the sampling should be good enough to be able to achieve high quality rendering results with a low number of samples.

Figure 3.2: *Environment maps which we want to use are a real-time stream (The example is from [Stumpfel* et al. *2004]). Our motivation is to capture and use such environment maps both in real time.*

3.2 Context and Motivation

For an environment map, each texel is considered as an individual light source. The direct illumination at position p with normal $\boldsymbol{n}$ is:

$$L(\boldsymbol{p} \rightarrow \boldsymbol{o}) = \int_{\Omega} \mathrm{s}(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega})\, \mathrm{d}\boldsymbol{\omega} \quad \text{with}$$

$$\mathrm{s}(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega}) = \rho(\boldsymbol{o}, \boldsymbol{\omega})\ \langle \boldsymbol{n}, \boldsymbol{\omega} \rangle\ V(\boldsymbol{p}, \boldsymbol{\omega})\, L(\boldsymbol{\omega} \rightarrow \boldsymbol{p})$$

where Ω is the full unit sphere, $\rho(\boldsymbol{o}, \boldsymbol{\omega})$ is the reflectance function (BRDF), $<,>$ the positive clamped dot product operator (aka *cosine factor*), $V(\boldsymbol{p}, \boldsymbol{\omega})$ is the visibility function and $L(\boldsymbol{\omega} \rightarrow \boldsymbol{p})$ represents the radiance in the direction $\boldsymbol{\omega}$ from the environment map. The size of the texture (e.g., $512 \times 512 \times 6$ floating texels) representing the environment map prohibits achieving real time results with a brute force approach. This becomes even more obvious if $L(\boldsymbol{\omega} \rightarrow \boldsymbol{p})$ is dynamic (i.e., on-line generated or acquired [Havran *et al.* 2005]).

Precomputed Radiance Transfer (PRT) techniques (e.g., [Sloan *et al.* 2002b; Wang *et al.* 2006b]) improve rendering performances and integrate complex BRDFs and inter-reflections, by projecting the lighting after one or multiple bounces into bases (cf. Section 2.6.2). However, the required memory to store the precomputed data can quickly become a bottleneck (e.g., with high frequency BRDF [Mahajan *et al.* 2008]) and the costly precomputation time prevents using PRT with dynamic environment map [Havran *et al.* 2005].

Another trend is to use the general Monte-Carlo framework. As we mentioned in Section 2.3, the reflected radiance $L(\boldsymbol{p} \rightarrow \boldsymbol{o})$ is approximated by N_s samples:

$$L(\boldsymbol{p} \rightarrow \boldsymbol{o}) \approx \frac{1}{N_s} \sum_{i=1}^{N_s} \frac{\mathrm{s}(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega}_i)}{\mathrm{pdf}(\boldsymbol{\omega}_i)} \tag{3.1}$$

where $\mathrm{pdf}(\boldsymbol{\omega}_i)$ is the Probability Density Function (PDF). The closer it matches the numerator of Equation 3.1 the lower the number of samples N_s will be required to converge to the solution. Therefore, the performance of such an approach depends on the ability of the PDF.

In this chapter, we focus on light importance sampling of dynamic environment maps (i.e., $\mathrm{pdf}(\boldsymbol{\omega})$ is proportional to $L(\boldsymbol{\omega} \rightarrow \boldsymbol{p})$). More precisely, we introduce a real-time GPU importance sampling technique that recomputes for each frame a tabulated version of the Cumulative Distribution Function (CDF) of an environment map. To generate the light samples $\{\boldsymbol{\omega}_i\}$, the CDF is inverted through a binary search before the shading pass. Our work can be applied to any kind of environment map (spherical, hemispherical, longitude-latitude) but is particularly interesting for cube maps that are widely used for real-time applications. Furthermore, to improve rendering performance, we also introduce an unbiased Monte-Carlo estimator that:

- limits the number of useless light samples that would have been null due to the cosine factor
- reduces popping artifacts that occur when using a low number of samples (this property is particularly important when using time-dependent light sources)
- integrates easily with Multiple Importance Sampling (MIS).

In the context of light importance sampling, many techniques (e.g., [Ostromoukhov *et al.* 2004]) for static environment maps have been introduced. However, their high computational cost make them unsuitable for dynamic lighting case. One notable exception is the work from Havran et al. [Havran *et al.* 2005] where they generate samples on the CPU. However, contrary to their work, we do not precompute a set of visible light sources for each shaded point p. Our approach computes dynamically a set of light samples according to the light distribution of the environment map and selects some of them according to the pixel normal: it is therefore more suited for dynamic environment maps.

3.3 Per-pixel Dynamic Sampling

3.3.1 Overview

Our technique generates high quality samples at very low cost that allows both sampling and rendering performing in real-time. The distribution of dynamic light samples for the highlighted pixel p (with a cyan dot) on a dragon when using dynamic environment maps are shown in Figure 3.3. In the cube map images (cf. right side of the Figure 3.3), the cyan dot corresponds to the normal direction $\boldsymbol{n}$. The green and yellow dots are the samples selected by our technique among the set of pre-generated samples (red dots, $N_s = 64$). The green dots correspond to effective samples (i.e., not clamped by the cosine factor) whereas the yellow ones correspond to samples clamped. We invite the reader to zoom in the images to have a better look at the dots.

3.3.2 Fast and Continuous Sampling

When using cube maps, one needs to distribute the M samples between the six faces of the cube. Let μ_f be the proportion of samples used for the face f (i.e., $M_f = \mu_f M$), and $\mathrm{pdf}_\mathrm{L}(\boldsymbol{\omega}|f)$ the probability density function to select direction $\boldsymbol{\omega}$ on the face f. Then, Equation 3.1 can be rewritten in a MIS formalism [Veach 1998]:

$$L_N(\boldsymbol{p} \rightarrow \boldsymbol{o}) = \sum_{f=1}^{6} \frac{1}{M_f} \sum_{i=1}^{M_f} \frac{\mathrm{s}(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega}_i)\, w_f}{\mathrm{pdf}_\mathrm{L}(\boldsymbol{\omega}_i|f)}$$

w_f is the weight for the sampling technique on f. In this formalism, we use the balance heuristic to compute each weight:

$$w_f = \frac{M_f \mathrm{pdf}_\mathrm{L}(\boldsymbol{\omega}_i|f)}{\sum_{j=1}^{6} M_j\, \mathrm{pdf}_\mathrm{L}(\boldsymbol{\omega}_i|f)}.$$

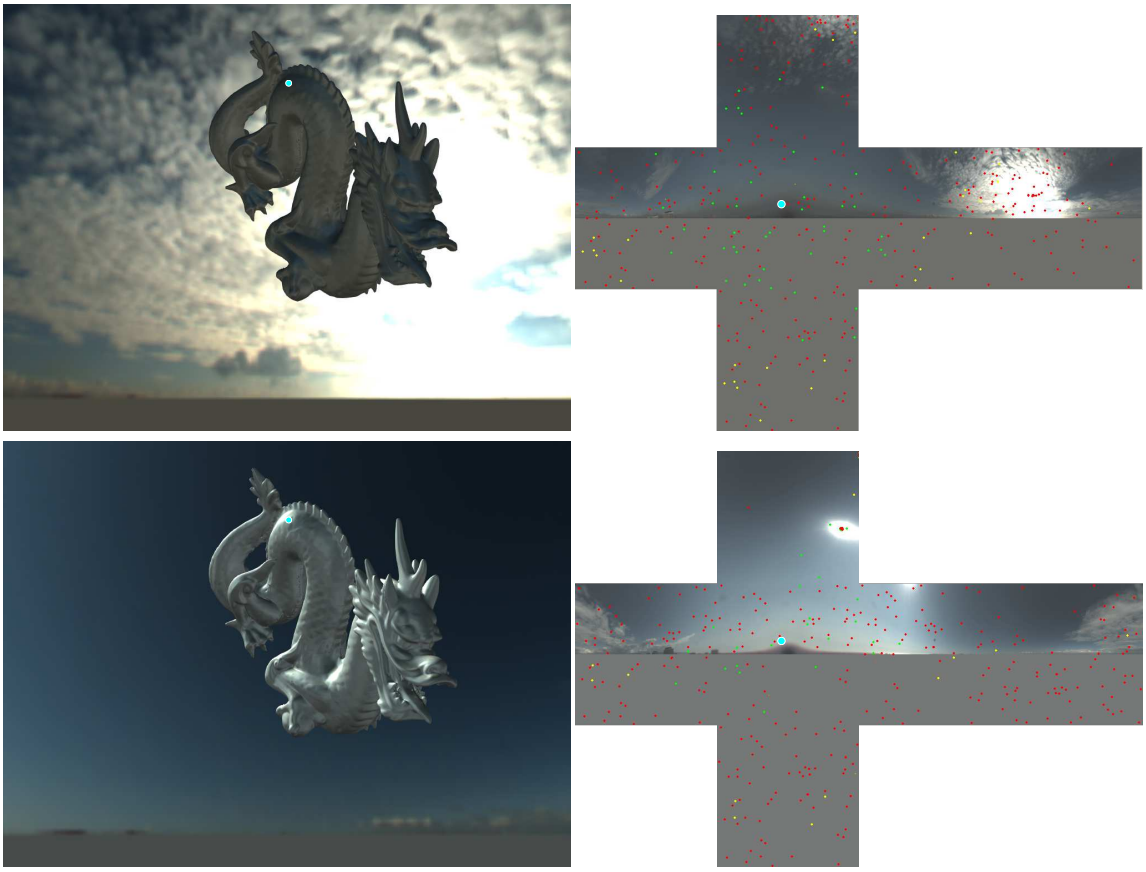


Figure 3.3: ***(Up)*** *At sunrise our technique prevents to generate samples on the sun,* ***(Down)*** *as soon as the sun starts to be visible for the highlighted pixel, our technique generates samples on the sun.*

Then the equation can be reduced to:

$$L_N(\boldsymbol{p} \to \boldsymbol{o}) = \frac{1}{M} \sum_{f=1}^{6} \sum_{i=1}^{M_f} \frac{\mathrm{s}(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega}_i)}{\sum_{j=1}^{6} \mu_j \, \mathrm{pdf}_{\mathrm{L}}(\boldsymbol{\omega}_i | f)} \,. \tag{3.2}$$

In this chapter, we target GPU importance sampling for dynamic environment maps. For this purpose, the cumulative distribution function of each cube face need to be computed at each frame. We thus use tabulated CDFs computed using optimized versions for GPU of the prefix sum [Harris *et al.* 2007] on each face.

Probability Density Function $\mathbf{pdf}_{\mathrm{L}}(\omega_i|f)$ The piecewise PDF associated with the tabulated CDF is computed as follow:

$$\mathrm{pdf}_{\mathrm{L}}(\boldsymbol{\omega}_i | f) = \frac{I_{ij}}{I_f} \frac{K}{4} (1 + u_i^2 + v_i^2)^{3/2} \tag{3.3}$$

where (u_i, v_i) are the face coordinates corresponding to direction $\boldsymbol{\omega}_i$ (values between $[-1, 1]$), I_{ij} is the luminance of the corresponding pixel and, I_f is the sum of all pixels luminance belonging to face f. K is the total number of pixels of the environment map face.

Intuitively, $4/K$ is the texel size on the unit cube and $4(1 + u_i^2 + v_i^2)^{-3/2}/K$ is an approximation of its solid angle (cf. Section 3.3.3). This factor thus projects a unit cube onto a unit sphere.

The corresponding CDF is the expected piecewise linear function. This is demonstrated as follow. For pixel (k, l) on face f, the CDF is expressed as

$$\mathrm{cd}f_{\mathrm{L}}^{kl}(\boldsymbol{\omega}|f) = \sum_{i \le k} \sum_{j \le l} \int \int \mathrm{pdf}_{\mathrm{L}}(\boldsymbol{\omega}|f)\, \phi_{ij}(u, v)\, \mathrm{d}\boldsymbol{\omega}$$

where ϕ_{ij} is equal to 1 on pixel (i, j) and zero elsewhere. Introducing Equation 3.3 leads to

$$\mathrm{cd}f_{\mathrm{L}}^{kl}(\boldsymbol{\omega}|f) = \frac{1}{I_f} \frac{K}{4} \sum_{i \le k} \sum_{j \le k} I_{ij} \int \int (1 + u^2 + v^2)^{3/2}\, \phi_{ij}(u, v)\, \mathrm{d}\boldsymbol{\omega}\,.$$

Since $\mathrm{d}\boldsymbol{\omega} \approx 1/(1 + u^2 + v^2)^{3/2} \mathrm{d}u\mathrm{d}v$, this simplifies to

$$\mathrm{cd}f_{\mathrm{L}}^{kl}(\boldsymbol{\omega}|f) = \frac{1}{I_f} \frac{K}{4} \sum_{i \le k} \sum_{j \le k} I_{ij} \int \int \phi_{ij}(u, v)\, \mathrm{d}u\mathrm{d}v\,.$$

Since ϕ_{ij} is constant, the resulting function is therefore piecewise linear. Since $\int \int \phi_{ij}(u, v)\, \mathrm{d}u\mathrm{d}v$ is the area of a pixel that is, $4/K$, and finally since I_f is the sum of all I_{ij}, we obtain on pixel (k, l)

$$\mathrm{cd}f_{\mathrm{L}}^{kl} = \frac{\sum_{i \le k} \sum_{l \le k} I_{ij}}{\sum_i \sum_j I_{ij}}\,,$$

which is exactly computed using prefix sum. We discuss the function $(1 + u^2 + v^2)^{3/2}$ more precisely in the next section.

Dynamic Number of samples for each face An obvious choice is to distribute uniformly the samples across the faces (i.e., $\mu_f = 1/6$). Unfortunately, depending on the orientation of the normal $\boldsymbol{n}$, some samples may be behind the point and will be canceled by the cosine factor. A drastic example is when a very bright source (such as the sun) is behind the point p: many useless samples will still be generated on the face the sun belongs to. To prevent this behavior, one solution is to integrate the cosine factor $\langle \boldsymbol{n}, \boldsymbol{\omega} \rangle$ into the PDF at the price of computing new CDFs that depends also on the normal $\boldsymbol{n}$. However, this becomes too costly in terms of memory and computation. Instead, we introduce a conservative, simple

and dynamic strategy where μ_f depends on the normal $\boldsymbol{n}$ and is computed as follow:

$$\mu_f(\boldsymbol{n}) = \frac{F_f(\boldsymbol{n})I_f}{\sum_f F_f(\boldsymbol{n})I_f} \text{ with } F_f(\boldsymbol{n}) = \sum_{c_j \in f} \left\langle \boldsymbol{n}, c_j \right\rangle \tag{3.4}$$

where c_j are normalized vertexes of the cube map face f. $F_f(\boldsymbol{n})$ can be seen as a conservative pseudo form factor (point to face). Equation (3.4) balances dynamically for p the number of samples on each face according to its importance.

To prevent popping artefacts when the number of samples is low and changing between neighboring normals, we use a floating point M_f and introduce a new weight β_i for each sample. By combining Equation (3.2) and Equation (3.4) our estimator becomes:

$$L(\boldsymbol{p} \to \boldsymbol{o}) \approx \frac{1}{N_s} \sum_{f=1}^{6} \sum_{i=1}^{i<M_f(\boldsymbol{n})+1} \beta_i \frac{\mathrm{s}(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega}_i)}{\mu_f(\boldsymbol{n}) \, \mathrm{pdf}_{\mathrm{L}}(\boldsymbol{\omega}_i | f)} \tag{3.5}$$

$$\text{with } \beta_i = \begin{cases} 1 & \text{if } i \leq M_f(\boldsymbol{n}) \\ \mathrm{frac}(M_f(\boldsymbol{n})) & \text{else} \end{cases} \tag{3.6}$$

where frac returns the fractional portion. Since $\mathrm{pdf}_{\mathrm{L}}(\boldsymbol{\omega}|f) = 0$ if $\boldsymbol{\omega} \notin f$, the sum in the denominator in Equation 3.2 disappears. Thanks to the use of the weight β_i, the last sample is introduced progressively.

As demonstrated in the results section, compared to the classical approach, our strategy improves the frame rate because it limits the generation of useless light samples. Based on MIS formalism, we can easily integrate other strategies. We illustrate this by using BRDF importance sampling $\mathrm{pdf}_{\mathrm{B}}$. For the balance heuristic, half samples are used for light importance sampling, half for BRDF. Therefore, we use $N_f(\boldsymbol{n}) = \mu_f(\boldsymbol{n})N_s/2$ for each face and $N_b = N_s/2$ for the BRDF. This leads to the following estimator:

$$\begin{aligned} L(\boldsymbol{p} \to \boldsymbol{o}) &\approx \frac{1}{N_s} \left(\sum_{f=1}^{6} \sum_{i=1}^{i<M_f(\boldsymbol{n})+1} \beta_i \, g_{f,i} + \sum_{i=1}^{i<N_b+1} \beta_i \, h_i \right) \\ \text{with } g_{f,i} &= \frac{\mathrm{s}(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega}_i)}{(1/2)\,\mu_f(\boldsymbol{n}) \, \mathrm{pdf}_{\mathrm{L}}(\boldsymbol{\omega}_i|f) + (1/2)\,\mathrm{pdf}_{\mathrm{B}}(\boldsymbol{\omega}_i)} \\ \text{and } h_i &= \frac{\mathrm{s}(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega}_i)}{(1/2)\,\sum_{j=1}^{6} \mu_j(\boldsymbol{n}) \, \mathrm{pdf}_{\mathrm{L}}(\boldsymbol{\omega}_i|j) + (1/2)\,\mathrm{pdf}_{\mathrm{B}}(\boldsymbol{\omega}_i)} \end{aligned} \tag{3.7}$$

$g_{f,i}$ is the sampling technique that generates samples from face f of the environment map and h_i is the one that generates samples from the BRDF, which face that a sample belongs to is unknown in advance. After generating a sample, the PDF for the light source can be computed according to the direction of the sample.

3.3.3 More Accurate Solid Angle

As we discussed in the previous work (cf. Section 2.7.2), when mapping environment lighting from parametrization of cube to sphere, solid angle should be taking into account to adjust the lighting intensity. There are two ways to introduce solid angle computation. Both of them lead to a easy way to compute CDFs. We remind reader that the solid angle of a micro surface on a texel is approximated by:

$$\mathrm{d}\omega \approx (1 + u^2 + v^2)^{-3/2}\mathrm{d}u\mathrm{d}v\,.$$

This is valid when the cube of an environment map is a cube whose fastest two corners are positioned at $(-1, -1, -1)$ and $(1, 1, 1)$ respectively.

We assume the color stored in each texel is radiant flux I_{ij} (unit W). Therefore, the radiant flux density l_{ij} (unit Wm^{-2}) of a texel at position ij is

$$l_{ij} = I_{ij}/\frac{N}{4}$$

where 4 is the area of a face, $N/4$ is the size of a texel and N is the number of texels on the face.

Brute force method In order to construct tabulated CDF for importance sampling, we have to compute the corresponding PDF where the solid angle is introduced. For each texel on a face of the environment map, the radiance I'_{ij} (unit $Wsr^{-1}m^{-2}$) of the texel is calculated as:

$$I'_{ij} = l_{ij}/\Delta\omega \tag{3.8}$$

where the solid angle $\Delta\omega$ (unit sr) of the texel is

$$\Delta\omega = \int\int (1 + u^2 + v^2)^{-3/2}\mathrm{d}u\mathrm{d}v\,.$$

Therefore the radiance is

$$I'_{ij} = \frac{l_{ij}}{\int\int (1 + u^2 + v^2)^{-3/2}\mathrm{d}u\mathrm{d}v}\,. \tag{3.9}$$

The radiant emittance $L(u, v)$ (unit Wm^{-2}) of the environment light is then defined as:

$$L(u, v) = \sum_{ij} I'_{ij}\phi_{ij}(u, v)(1 + u^2 + v^2)^{-3/2} \tag{3.10}$$

where $\phi_{ij}(u, v)$ is a piecewise constant function, if (u, v) is in the texel (i, j), the function is

equal to 1. Otherwise, it is equal to 0. Finally, the PDF of the light source becomes:

$$\text{pdf}(u, v) = \frac{L(u, v)}{L_{total}} . \tag{3.11}$$

Therefore, the corresponding CDF is

$$cdf(u, v) = \int \text{pdf}(u, v) . \tag{3.12}$$

Note that, since the PDF is not piecewise constant, the CDF is not piecewise linear. If we access the tabular CDF with linear interpolation, errors would be introduced (For efficiency purpose, linear interpolation is preferred on GPU).

Practical method The main overhead of the brute force method is the computation of $\Delta\omega$, since there is a integral which needs numerical methods to solve. This overhead is too heavy for real-time applications. Therefore, in practice, we prefer to use an approximated one.

To compute $\Delta\omega$ efficiently and keep the accuracy as much as possible, we select only one point (u_0, v_0) in the middle of the texel area. Then the $\Delta\omega$ is proportional to:

$$\Delta\omega = (1 + {u_0}^2 + {v_0}^2)^{-3/2} . \tag{3.13}$$

Therefore, the radiance in a texel is constant.

$$I'_{ij} = \frac{l_{ij}}{(1 + {u_0}^2 + {v_0}^2)^{-3/2}} . \tag{3.14}$$

According to the Equation 3.10, the radiant emittance $L(u, v)$ (unit Wm^{-2}) of the environment light becomes:

$$L(u, v) = \sum_{ij} l_{ij}\phi_{ij}(u, v) . \tag{3.15}$$

Then the corresponding PDF and CDF are

$$\text{pdf}(u, v) = \frac{L(u, v)}{L_{total}}$$
$$cdf(u, v) = \int \text{pdf}(u, v) .$$

Since $L(u, v)$ is constant in a texel, the PDF is piecewise constant and the CDF is piecewise linear. Linear interpolation introduces no error when accessing the CDF.

Although the practical method introduces a little approximation, this approximation is quite small, since the difference between these two methods is dependent on the pixel size.

Conservatively, the maximum directional error of using the practical method compared with the brute one is 0.003068 in radian (0.1% error) when each cube map face has resolution of 256×256 pixels (0.05% error with 512×512 pixels and 0.025% error with 1024×1024 pixels). We will use the practical method for all of our results in this chapter.

3.3.4 Combining with BRDF Sampling

Our light source sampling technique is able to be combined with BRDF-based techniques that have close-formed or non-close-formed. We have combined our light source sampling with a close-from BRDF [Lafortune & Willems 1994] (cf. Section 2.3.5). When sampling from an environment map, we use our method introduced in Section 3.3.1. When sampling from the BRDF, we follow the idea from the original paper that randomly select diffuse or specular part.

Close-from BRDF for BRDF representing is similar to classical analytical light sources for light sources representing. It is quite limited in representing measured BRDF which has arbitrary frequencies. A non-close-form representation model is required to fit this measured data (e.g., rational BRDF [Pacanowski *et al.* 2012]).

In order to use rational BRDF, we need to fit not only the BRDF itself, but also related cdf^{-1}s which are used for PDF computing. For BRDF-based importance sampling we need to fit these two cdf^{-1}s functions representing the BRDF $\times$ Cosine factor:

$$\begin{aligned} cdf^{-1}(\varepsilon_1, \theta_v) &= \theta_l \\ cdf^{-1}(\varepsilon_2, \theta_v, \theta_l) &= \phi_l \end{aligned} \tag{3.16}$$

where ε_1 and ε_2 are two random number defined in $[0, 1]$. As shown in Figure 3.4. v is the view direction, l is a light direction. θ and ϕ are angles to the normal n.

Then the PDF can be computed as:

$$\text{pdf}(\theta_v, \theta_l, \phi_l) = \frac{\text{d}}{\text{d}\varepsilon_1}\left(cdf^{-1}(\varepsilon_1, \theta_v)\right) \times \frac{\text{d}}{\text{d}\varepsilon_2}\left(cdf^{-1}(\varepsilon_2, \theta_v, \theta_l)\right) . \tag{3.17}$$

For light-based importance sampling, we use our method introduced in Section 3.3.1, since the samples and their PDFs calculation is independent from the BRDF. However, if we want to use MIS to combined BRDF-based and light-based importance sampling, we need to fit more functions:

$$\begin{aligned} cdf(\theta_v, \theta_l) &= \varepsilon_1 \\ cdf(\theta_v, \theta_l, \phi_l) &= \varepsilon_2 . \end{aligned} \tag{3.18}$$

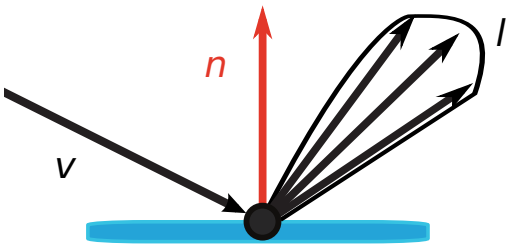


Figure 3.4: *Light transport and the parametrization: v is the view direction, l is a light direction and n is the normal.*

These two CDFs are required since the samples generated from light-based importance sampling need PDFs both from the BRDF and the light source for MIS. Although the samples generated from BRDF-based importance sampling also need two PDFs, they can be easily computed since the light intensity of a sample and the cosine can be obtained straightforwardly, and Equation 3.16 is provided.

Then the PDF can be computed as:

$$\text{pdf}(\theta_v, \theta_l, \phi_l) = \frac{\text{d}}{\text{d}\varepsilon_1}\left(\text{c}df(\theta_v, \theta_l)\right) \times \frac{\text{d}}{\text{d}\varepsilon_2}\left(\text{c}df(\theta_v, \theta_l, \phi_l)\right) . \tag{3.19}$$

Blue-metallic (64fps) Beige (32fps) Chrome(10fps)

Figure 3.5: *Our light-based importance sampling for rational BRDFs: (**Left** and **Middle**) are "Blue" and "Beige" materials. The light-based importance sampling yields good results since these two materials are not very specular. On the contrary, (**Right**) "Chrome" material is too specular to be suited our light-based importance sampling. A BRDF-based importance sampling or MIS will lead to better results (all results are rendered with 512 light samples per pixel at 1024x768 resolution, the environment map is the same one as Figure 3.6).*

We only tested light-based importance sampling for rational BRDF (cf. Figure 3.5).

3.4 GPU Implementation

Algorithm 1 presents the rendering process for each frame. The whole process starts by an early GBuffer pass (line 4) and ends by the final tone mapping (lines 15 and 16). In-between, we compute the tabulated CDF (lines 5 to 11), then we generate the light samples (line 10) and compute the shading (line 13).

We compute the 2D CDF by using the inversion method [Pharr & Humphreys 2004]) and store it on GPU. More precisely, a 1D CDF (CDF(u)) and a 2D CDF (CDF($v|u$)) are computed using parallel prefix sum [Harris *et al.* 2007] and stored as floating point buffer for each cube map face (u and v are the pixel coordinates on a face corresponding to a given light sample). The CDF computations are implemented using two GPU Computing kernels (lines 8 and 9) that are called successively because CDF($v|u$) depends on CDF(u). Based on these CDFs, we conservatively generate $N_s/2$ light samples per face before computing the shading. This ensures the generation of a sufficient amount of samples on each face for the dynamic balancing. For degenerated cases, all the $N_s/2$ samples dedicated to light sampling will be on a unique face. These samples are generated using a classical binary search.

High performance in CUDA requires thread coherency, memory coalescing and branch consistency. We have therefore designed the Algorithm 2 for the shading step (cf. line 13) to reach the best performance of our different implementations. After shading, the computation of the average luminance is done using a reduction operation [Roger *et al.* 2007] for the tone mapping step.

Finally, since our per pixel Monte Carlo estimator is unbiased, our approach does not introduce any bias for a given pixel. However, for efficiency reason, we use the same precomputed random sequence for all the pixels. This introduces a spatial bias between neighbor pixels but it has the advantage to limit disturbing noise in the final image. A possible extension to reduce the spatial bias would be to use interleaved sampling [Keller & Heidrich 2001].

3.5 Results and Discussion

All results presented in this chapter were computed on a 2.67 GHz PC with 6 GB of RAM and a NVIDIA GTX 680 graphics card. We implemented our system using DirectX, CUDA and the Thrust Library. The static environment map used in Figure 3.6 has a resolution of $256 \times 256 \times 6$ pixels. For the dynamic environment map used in Figure 3.3, we use the available 67 frames from the capture made during a full day by Stumpfel et al. [Stumpfel *et al.* 2004]. Before using it, the only preprocess we apply on the captured images is a

Algorithm 1 Steps to render one frame. We use GC when using GPU computing and GS when using GPU shader. We indicate the for-loop that are parallelized using CUDA threads with the keyword **"do in parallel"**.

```
1:  procedure RenderFrame
2:      GEO_2D                                   ▷ Buffer for vertex positions and normals
3:      LS_1D[f]                                 ▷ Buffers of light samples for each face f
4:      initialize(GEO_2D)                                                          ▷ GS
5:      for each face f of the cube environment map do
6:          compute_luminance_per_pixel                                             ▷ GC
7:          compute_prefix_sum_by_inclusive_scan                                    ▷ GC
8:          compute_CDF_1D_u                                                        ▷ GC
9:          compute_CDF_2D_v_knowing_u                                              ▷ GC
10:         LS_1D[f]=generate_light_samples(N_s/2)                                  ▷ GC
11:     end for
12:     for each pixel (p, n) ∈ GEO_2D  do in parallel
13:         shade(o,p,n,N_s)                                                        ▷ GC - Algo. 2
14:     end for
15:     compute_average_intensity(frame)                                            ▷ GC
16:     Tone_mapping(frame)                                                         ▷ GS
17: end procedure
```

re-parametrization (from hemisphere to cube) to obtain $256 \times 256 \times 6$ pixels images. We also fill with an average color the pixels belonging to the roof where the acquisition device was placed. For all results, we achieve real-time framerates without visibility computation. The results with the dragon model (369K polygons) as well as the model (169K polygons) used in Figure 3.6.

Figure 3.3 shows a set of images with a time-varying environment map captured from daylight to night. The central images show where the light samples are located for a given pixel. Notice how our technique avoids generating useless samples on the sun for the highlighted pixel. Moreover, the companion video demonstrates our real-time performance as well as the temporal coherence of our Monte Carlo estimator when changing the view direction or the lighting or even the BRDF parameters.

Figure 3.6 presents a comparison between a classical light-based importance sampling and our approach. As shown in the picture, both images have almost the same quality when compared to a reference solution, but our technique uses three times less samples and is consequently faster. Finally, Figure 3.7 demonstrates the convergence speed and the quality obtained with our MIS technique. As shown by the Lab difference between our results and a reference solution, even with 60 samples per pixel our method achieves visual plausible results in real-time (198 fps). Figure 3.9 shows a comparison between our floating number of samples (β_i) and classical integer number of samples. The floating number of samples allows introducing each new sample progressively among neighboring pixels, and

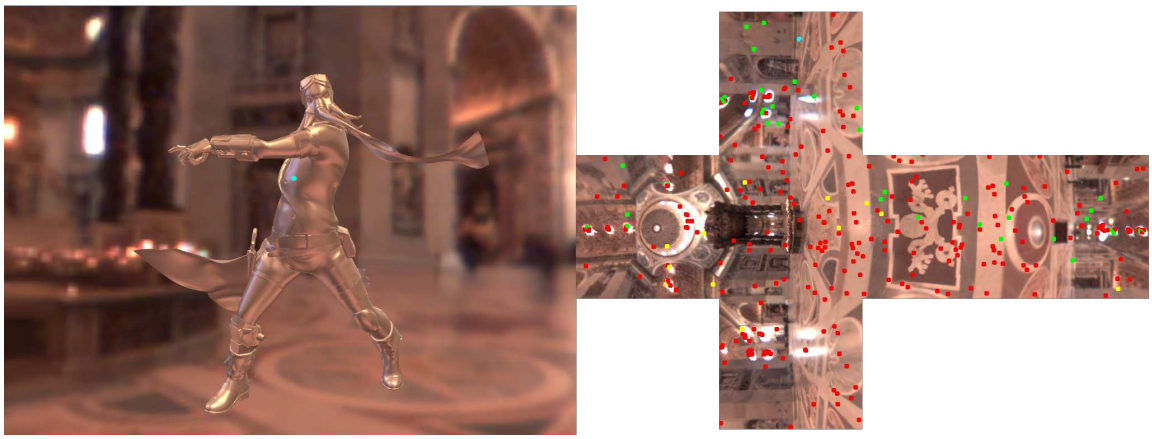
Algorithm 2 Shading procedure with our dynamic balancing technique. To take advantage of the CUDA architecture, we have integrated the BRDF sampling pass as a 7th pass and used a fixed maximal number of iterations (cf Line 6).

```
1:  procedure SHADE(o,p,n,N_s)                          ▷ in parallel for each pixel p
2:      N_f[1..6] = Samples_Per_Face(N_s/2)             ▷ Equation 3.4
3:      N_f[7] = N_s/2                                  ▷ number of BRDF samples
4:      for each step f=1..7 do
5:          for each i = [0, N_s/2] do
6:              break when i ≥ N_f[f]
7:              compute β_i                             ▷ Equation 3.6
8:              if f < 7 then
9:                  sample = LS_1D[f][i]                ▷ see Algo. 1
10:             else
11:                 sample = BRDF_SAMPLING
12:             end if
13:             L(p → o)+= β_i g_{f,i}                  ▷ Equation 3.7
14:         end for
15:     end for
16:     L(p → o) = L(p → o)/N_s
17: end procedure
```

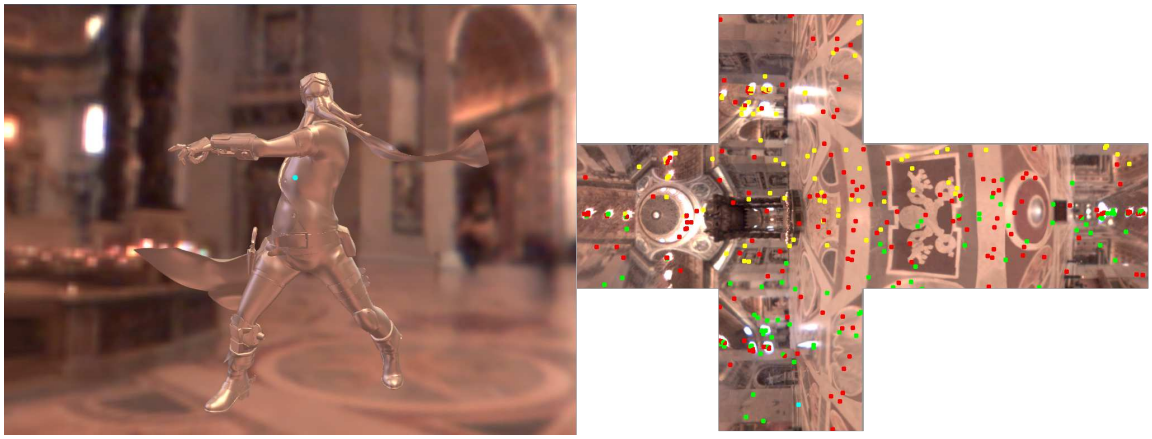
thus reduces discontinuities. Figure 3.8 shows a sequence of screen shots from our real-time application, as well as their corresponding distribution of samples.

3.6 Conclusion

In this chapter we have introduced an improved Monte Carlo estimator for light-based importance sampling of dynamic environment maps. Our pixel-based technique increases the number of effective samples and is faster for the same quality compared to a uniform distribution of samples on each face. Furthermore, our technique handles efficiently dynamic and time-varying environment maps. Based on Multiple Importance Sampling formalism, it can be easily combined with other sampling strategies. For future work, we would like to incorporate a more robust balancing scheme to distribute light samples and also to introduce visibility and indirect lighting effects.

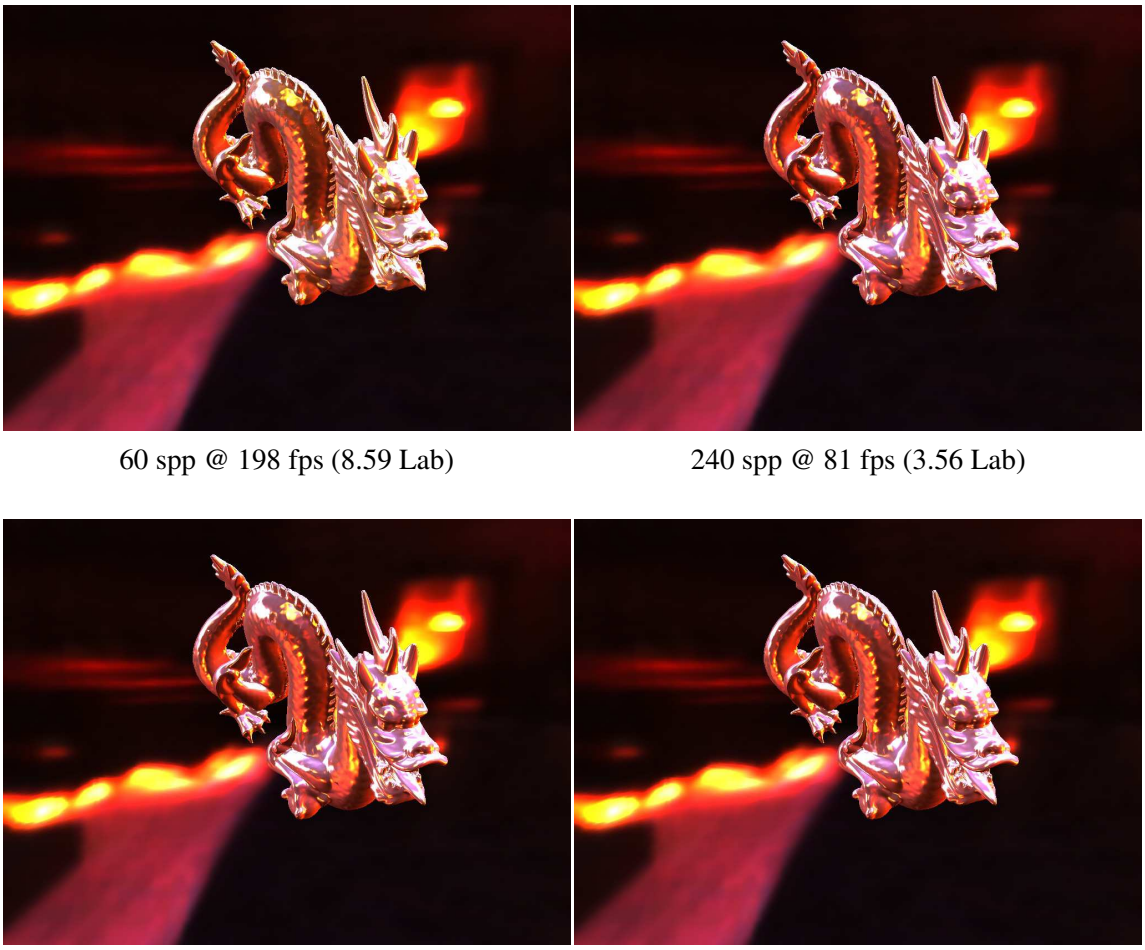


Our Dynamic Balancing: Mean Lab Error: 6.44 Number of samples: 60 Valid samples: 47



Uniform Balancing: Mean Lab Error: 6.22 Number of samples: 180 Valid samples: 116

Figure 3.6: *Comparison of (**Up**) our dynamic sampling technique for the highlighted pixel (with a cyan dot) with (**Down**) uniform balancing of the samples per face. Among the pre-generated samples (red+yellow+green), our technique selects* 60 *of them for the current pixel (yellow and green dots) from which 47 are effective samples (green dots). However, to achieve the same quality with uniform balancing, three times more samples are required (180 vs 60) resulting in 116 effective samples. The Lab errors are compared with a reference solution computed with* $256 \times 256 \times 6$ *samples generated uniformly on the environment map.*



60 spp @ 198 fps (8.59 Lab)

240 spp @ 81 fps (3.56 Lab)

480 spp @ 45 fps (1.87 Lab)

655 000 spp - Reference Solution

Figure 3.7: *Convergence speed of our technique when increasing the number of samples per pixel (spp). We compute the mean Lab error between our results and a reference solution (right image) computed using 655 000 spp generated from a cosine-based hemisphere sampling scheme. As shown by the decreasing Lab errors our MIS technique converges toward the correct solution when increasing the number of samples. Even with 240 spp, we achieve real-time frame rate (81 fps) with a very low Lab error (3.56).*

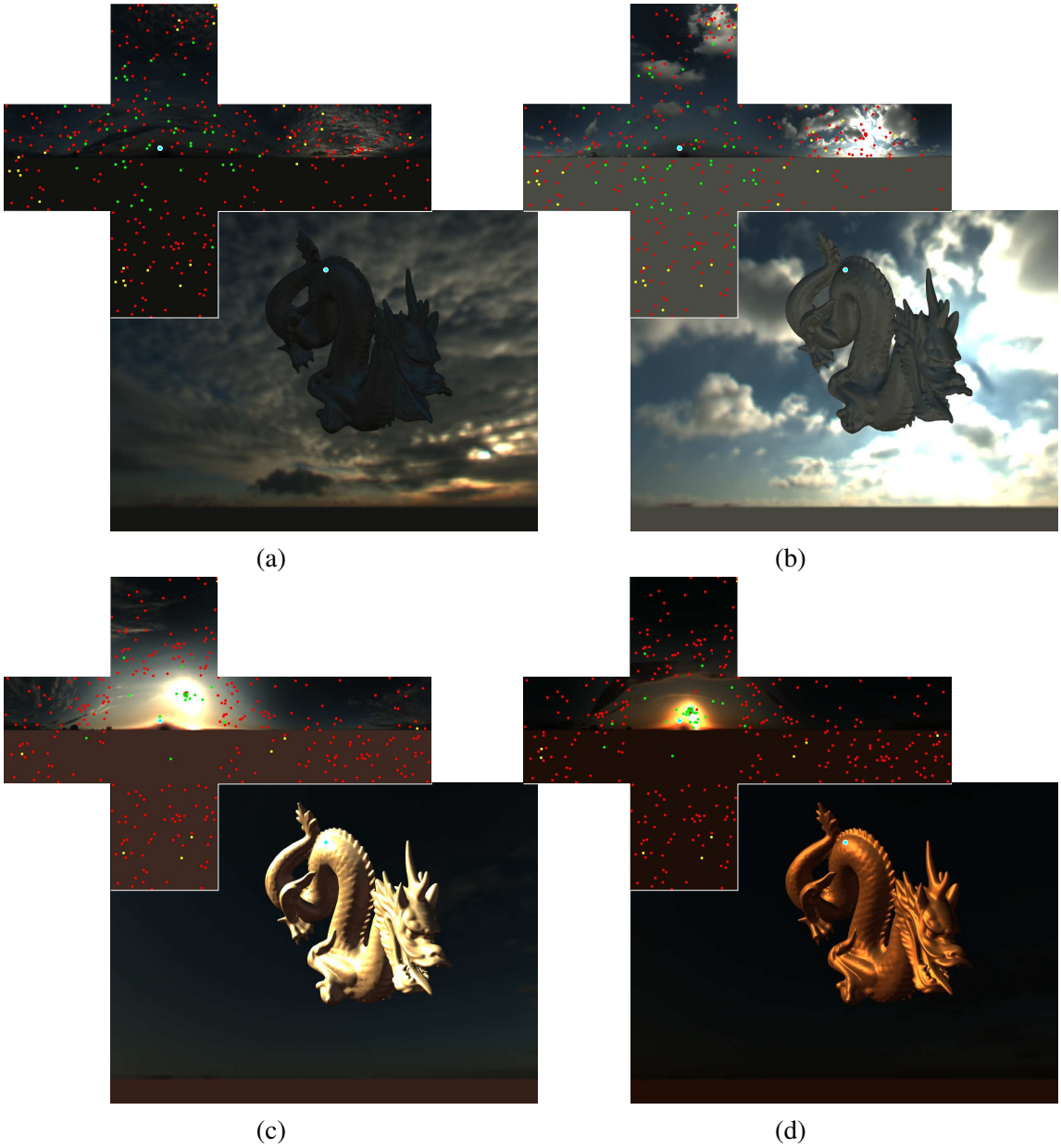


Figure 3.8: *Rendered images at different time of the day with Stumpfel's[2004] data. For each image, we show where the light samples are located in the cross-map for the pixel located by the cyan dot on the dragon. Please refer to Figure 3.3 for the dots color code meaning. Observe how in images (a) to (b) our technique prevents to generate samples on the sun because it is behind the shaded point. On the other side, in images (c) and (d) when the sun faces the shaded point the many samples are generated to faithfully represent its importance. The number of targeted samples (N_s) is 60 and the framerate is around 215 fps.*

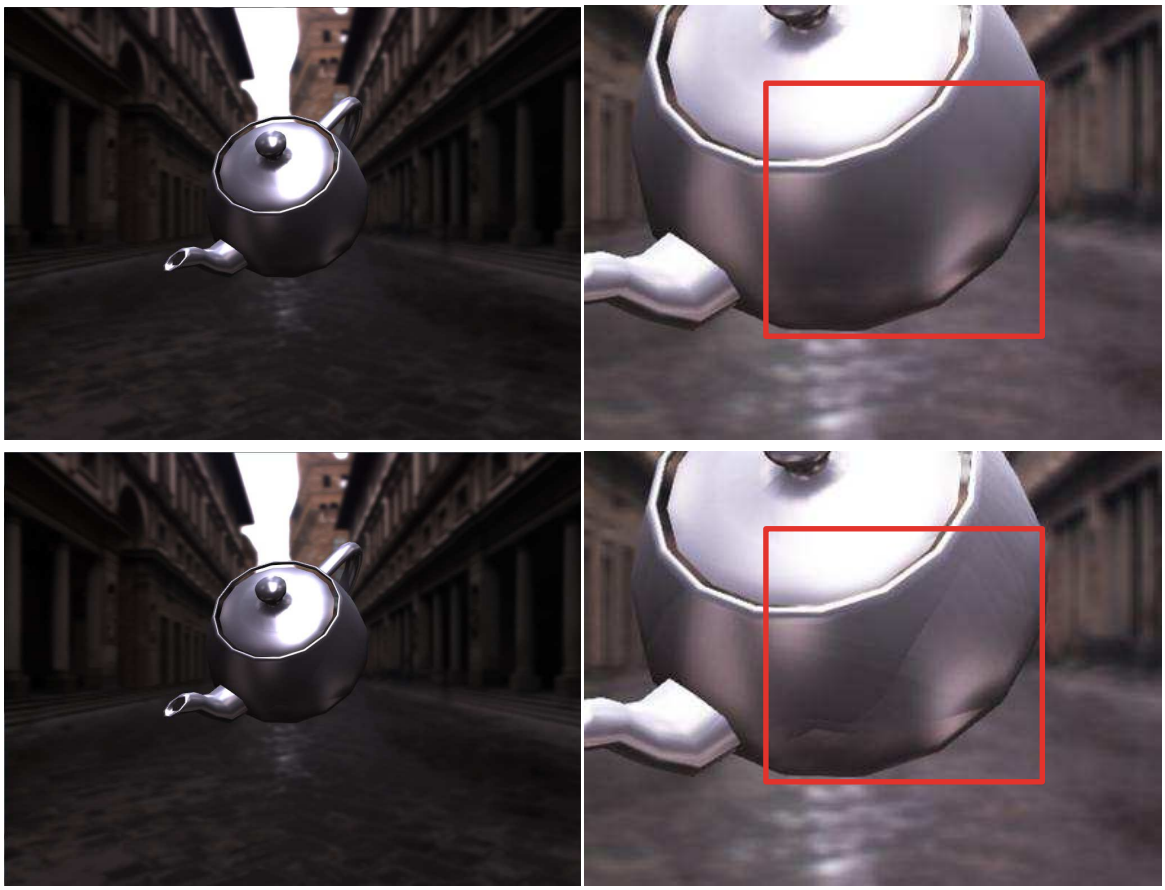


Figure 3.9: *Comparisons of rendered images with (**Up**) the weight β_i and without (**Down**) where discontinuities are introduced. The cube map size is* $512 \times 512 \times 6$.

Chapter 4

Near-field illumination: Light field Luminaires

In the previous chapter, our method was targeted for realistic light sources that contain only directional variations. In this chapter, we will focus on a more general case: realistic light sources that have both directional and spatial variations. Among existing models, the ones based on light fields (cf. Section 2.7.3) are attractive due to their ability to represent faithfully the near-field and due to their possibility to be directly acquired.

In this chapter, we introduce a dynamic sampling strategy for complex light field luminaires with the corresponding unbiased estimator. The sampling strategy is adapted, for each 3D scene position and each frame, by restricting the sampling domain dynamically and by balancing the number of samples between the different components of the representation. This is efficiently achieved by simple position-dependent affine transformations and restrictions of CDF that ensure that every generated sample conveys energy and contributes to the final results. Therefore, our approach only requires a low number of samples to achieve almost converged results. We demonstrate the efficiency on modern hardware of our approach by introducing a GPU-based implementation. Combined with a fast shadow algorithm, our solution exhibits interactive frame rates for direct lighting and for large measured luminaires. This work has been submitted to IEEE Transactions on Visualization and Computer Graphics.

4.1 Motivation

Thanks to the realistic light sources, rendering techniques have significantly improved qualitatively, to obtain realistic and plausible solutions, and quantitatively to simulate physical phenomena (cf. Chapter 2).

As we mentioned in Section 2.6.1, standard light sources are represented by point, directional or uniform area models. Even though point light sources may be extended with

goniometric diagrams for computer graphics [Verbeck & Greenberg 1984] or for professionals [IESNA Committee 2001], all of these models are still limited in term of spatial variations of emitters. The point assumption is only valid for regions of a 3D scene where the distance to the luminaire is large compared to its size. Furthermore, real luminaires are hardly approximated by uniformly emitting surfaces. For example, complex emittance function of indoor luminaires or headlights cannot be represented accurately by uniform distributions.

Using captured distant light sources such as environment maps leads to better realism (cf. Chapter 3). However, as a many directional light sources model, these light sources are lacking spatial information inherently.

A classic way to improve the physical accuracy is to capture the so-called 4D *near-field* emissivity by sampling the light space around the emitter using either a *ray set* [Ashdown & Rykowski 1997; Mas *et al.* 2008], or more densely with a set of images [Ashdown 1995]. Goesele et al. [Goesele *et al.* 2003] have built an acquisition system where the 4D near-field is projected on a predefined basis, leading to a priori control of the model accuracy. Although the authors have shown that their model can be used in an interactive manner using graphics hardware [Granier *et al.* 2003], their dedicated approach was still limited to relatively small data set while introducing large approximations. Despite their accuracy, their realism and the relatively simple acquisition systems required by light field luminaires, the lack of efficient and accurate rendering approaches for them is probably the reason why they are still not widely used compared the limited classical light models. In this chapter, we demonstrate that a simple importance sampling approach is sufficient to obtain real-time and accurate solution when combined with a GPU implementation. Furthermore, we hope this efficient rendering technique would promote the use of such near-field light sources.

4.2 Related Work

Importance sampling (cf. Section 2.3.5) is a large research area in Computer Graphics. In this chapter, we focus only on importance sampling to compute direct lighting from 4D real world luminaires. Despite the recent progress in global illumination, direct lighting is still a very important step for any computation since it is always the first one and since, in most cases, it greatly contributes to the final quality. This is even more true for interactive global illumination techniques as detailed in the state-of-the-art of Ritschel et al. [Ritschel *et al.* 2012]. Since we are focusing our work only on light source importance sampling, we do not review techniques that apply to BRDF or visibility, or the product of both. Our approach is complementary to these solutions and we discuss this point in Section 4.6.

One possible solution to integrate complex real world luminaires is to use photon map-

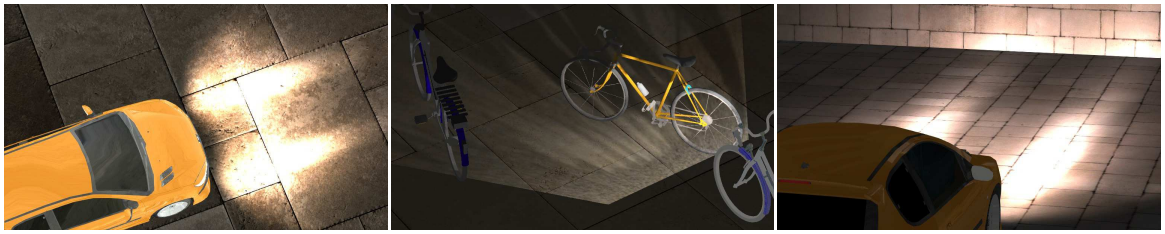
ping [Jensen 2001] as demonstrated by Goesele et al. [Goesele *et al.* 2003]. Despite recent improvements in interactive photon mapping [Yao *et al.* 2010] a final gathering [Wang *et al.* 2009] pass is still required to capture accurately all the details of direct illumination. Recently, progressive photon mapping [Knaus & Zwicker 2011] has greatly improved the quality control by reducing progressively and adaptively the search neighborhood in order to balance between noise and bias reduction. However, to reach high-quality images, direct lighting requires a large number of passes and photons. Since we focus on direct lighting, it seems more efficient to sample directly the incident field for each scene position.

For this purpose, one possible solution is to approximate complex light sources by a set of fixed point lights (e.g., the technique of Agarwal et al. [Agarwal *et al.* 2003] for environment maps). A position-dependent selection of light sources can be done by using importance resampling [Talbot *et al.* 2005] at the additional cost to evaluate a function for each precomputed light sample.

One way to select quickly a large set of light sources, according to the 3D scene position $\boldsymbol{p}$, is to organize them hierarchically. With *lightcuts* [Walter *et al.* 2005b], the direct and indirect light sources are organized in a binary hierarchy. At rendering time, a cut is performed in the hierarchy and the corresponding nodes are used for computation. Lightcuts have been used for light-based importance sampling [Wang & Akerlund 2009] but with the original limitation to constant or cosine-distribution luminaires. This technique has been extended to spherical light-sources [Hašan *et al.* 2009], but they may not be directly used for light field luminaires. Our proposed approach does not require any conversion of the original data set into a set of lightcuts-compatible light sources.

Structured data such as light fields can be organized hierarchically by projecting them onto a wavelet basis. The technique introduced by Clarberg et al [Clarberg *et al.* 2005] and later improved [Clarberg & Akenine-Möller 2008] uses Haar basis for both the lighting space (2D environment maps) and the BRDF space. The product is evaluated on the fly and then used to guide the sample distribution. The memory and computation costs limit their approach to low resolution approximation of BRDF and light sources. This limitation has been later reduced by Cline et al. [Cline *et al.* 2006] thanks to a hierarchical splitting of the environment map guided by BRDF peaks. However, all of these techniques have been developed for far-field 2D lighting where incoming lighting is independent of the 3D scene position. Therefore, they are not applicable directly to 4D light field luminaires because near-field effects lead to a different incoming lighting for each 3D scene position.

To our knowledge, only two techniques deal with complex light field luminaires. The first one is the work of Granier et al. [Granier *et al.* 2003], but it only achieved low speed on small models with quite large approximations. In the second one [Mas *et al.* 2008], importance sampling is done according to the direct map (i.e., a set of particle emitted



9fps - 200spp 12fps - 200spp 7fps - 200spp

Figure 4.1: *Our new light importance sampling technique estimates direct lighting interactively with only 200 samples per pixel (spp) that are distributed among the different images of the light field luminaire. (**Left and Right**) The car headlights are represented by the same light field composed of* 11×9 *images (*256×256 *pixels). (**Center**) The bike headlight light field contains* 9×7 *images (*300×300 *pixels). For center image, the visibility is computed with 80 shadow maps (*256×256 *pixels).*

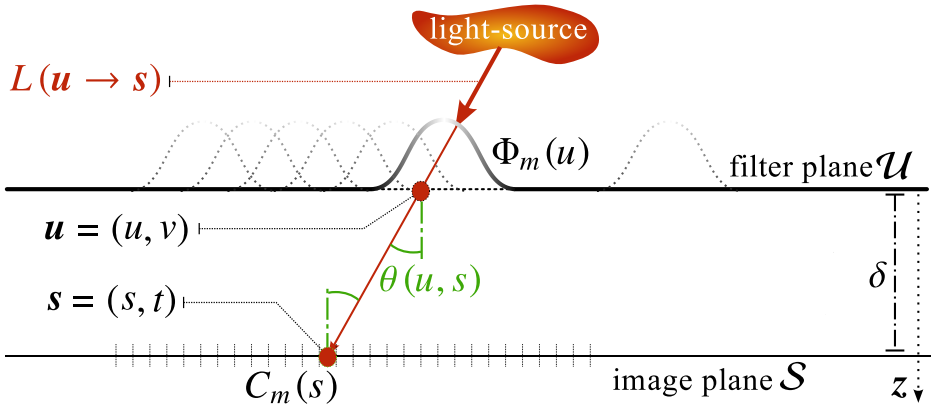


Figure 4.2: *Original rendering model for the two-plane setup (cf. Section 2.7.3).*

from the luminaire). Bias may be introduced when a low number of particles is used if the importance sampling function is not conservatively reconstructed. A too conservative approach may generate useless samples that correspond to rays with low or null energy. As pointed by Cline et al. [Cline *et al.* 2006], the coarser the approximation, the greater the risk to generate useless samples. In our approach, we stay as close as possible to the original data, without introducing any approximation: this ensures that we render almost all details that were originally measured by the acquisition process. Furthermore, our importance sampling closely mimics the behavior of the luminaire without introducing any bias: it quickly converges to the desired results with a low number of samples.

4.3 Position-dependent Importance Sampling

In order to facilitate the understanding of our contributions, we start by reminding that the near-field emission of a light source can be represented by a light field parametrized by two parallel planes (cf. Section 2.7.3). As illustrated in the Figure 4.2, the radiance $L(\boldsymbol{u} \rightarrow \boldsymbol{s})$ is

computed as:

$$L(\boldsymbol{u} \rightarrow \boldsymbol{s}) = \frac{|\boldsymbol{u} - \boldsymbol{s}|^2}{\cos^2 \theta\,(\boldsymbol{u}, \boldsymbol{s})} \sum_m C_m(\boldsymbol{s})\, \Phi_m(\boldsymbol{u})$$

with $L(\boldsymbol{u} \rightarrow \boldsymbol{s})$ representing the radiance transfered from $\boldsymbol{u}$ to $\boldsymbol{s}$. $\Phi_m(\boldsymbol{u})$ expresses m^{th} basis function on filter plane and $C_m(\boldsymbol{s})$ represents m^{th} image parametrized on the image plane (all notations in this chapter are defined in Table 4.1). To simplify the notation, the reader can note that the geometric configuration leads to

$$\delta = |\boldsymbol{u} - \boldsymbol{s}| \cos \theta\,(\boldsymbol{u}, \boldsymbol{s}) \ .$$

By combining these two equations together we obtain the equation used in this section:

$$L(\boldsymbol{u} \rightarrow \boldsymbol{s}) = \frac{|\boldsymbol{s} - \boldsymbol{u}|^4}{\delta^2} \sum_m C_m(\boldsymbol{s})\, \Phi_m(\boldsymbol{u})\,. \tag{4.1}$$

For simplicity, we denote $\Psi_m(\boldsymbol{u} \rightarrow \boldsymbol{s}) = \Phi_m(\boldsymbol{u})\, \delta^2 \cos^{-4} \theta$, then Equation 4.1 simplifies to:

$$L(\boldsymbol{u} \rightarrow \boldsymbol{s}) = \sum_m C_m(\boldsymbol{s})\, \Psi_m(\boldsymbol{u} \rightarrow \boldsymbol{s}). \tag{4.2}$$

This notation generalizes the lumigraph [Gortler *et al.* 1996], the lumigraph-inspired canned light sources [Heidrich *et al.* 1998] and the luminaire models of Goesele et al. [Goesele *et al.* 2003]. More details about the differences for Ψ_m and C_m are given in Section 4.3.1.

For a light source model based on light field, the irradiance $I(\boldsymbol{p})$ that potentially reaches $\boldsymbol{p}$ is defined by

$$I(\boldsymbol{p}) = \int_{\mathcal{I}} L(\boldsymbol{u} \rightarrow \boldsymbol{p}) \frac{\Delta(\boldsymbol{p})}{|\boldsymbol{s} - \boldsymbol{p}|^3}\, d\boldsymbol{s}$$

or, in a more compact form, by

$$I(\boldsymbol{p}) = \Delta(\boldsymbol{p}) \int_{\mathcal{I}} L(\boldsymbol{u} \rightarrow \boldsymbol{s})\, \frac{1}{|\boldsymbol{s} - \boldsymbol{p}|^3}\, d\boldsymbol{s} \tag{4.3}$$

where $\Delta(\boldsymbol{p})$ is the distance between $\boldsymbol{p}$ and the plane $\mathcal{I}$. A **key** observation with any light field model, is that $L(\boldsymbol{s} \rightarrow \boldsymbol{p})$ is equal to $L(\boldsymbol{u} \rightarrow \boldsymbol{s})$ when assuming that no visibility events nor participating media are present. Therefore, combining Equation 4.2 and Equation 4.3 results in a new formulation for the irradiance:

$$I(\boldsymbol{p}) = \sum_m I_m(\boldsymbol{p}) \tag{4.4}$$

$$I_m(\boldsymbol{p}) = \Delta(\boldsymbol{p}) \int_{\mathcal{I}} C_m(\boldsymbol{s})\, \Psi_m(\boldsymbol{u} \rightarrow \boldsymbol{s})\, \frac{1}{|\boldsymbol{s} - \boldsymbol{p}|^3}\, d\boldsymbol{s}\,. \tag{4.5}$$

The reader should keep in mind that $\boldsymbol{u}$ depends on $\boldsymbol{p}$ and $\boldsymbol{s}$. In fact, $\boldsymbol{u}$ is the intersection between the line $\overline{\boldsymbol{ps}}$ and plane $\mathcal{F}$. As detailed in the next section, methods that are both efficient and accurate have never been introduced to sample $I(\boldsymbol{p})$ for light field sources on a view-dependent and adaptive manner. To our knowledge, our new technique is also the *first* to achieve interactive frame rates while providing high-quality and accurate results.

In this chapter, we focus on sampling $I(\boldsymbol{p})$ efficiently and without introducing any bias by dynamically constructing restricted Cumulative Distribution Functions (CDFs) for each scene position $\boldsymbol{p}$. More specifically, we introduce the following contributions:

- **Position-dependent Restriction of CDF**. We demonstrate that we can *dynamically* apply a *position-dependent* affine transformation on a CDF to *restrict* the sampling domain and consequently reduce the number of light samples without decreasing the result quality.

- **Simple Balancing** Strategy. Additionally, we introduce an efficient balancing strategy that prevents generating light samples that convey only a small amount of energy. In other words, for each 3D scene position, our sampling strategy distributes light samples dynamically among the different light field images according to their intensity.

- **GPU implementation**. We demonstrate a GPU implementation of our CDF restriction and balancing strategy that reaches interactive frame rates (cf. Figure 4.1). Furthermore, we combine direct lighting with shadow effects by introducing a new shadow map-based algorithm that approximates the visibility.

4.3.1 Preprocess of CDF Construction

The key idea of our approach is to define a sampling strategy that depends on the scene position $\boldsymbol{p}$. This is achieved by using a position-dependent Probability Density Function (PDF) denoted by $\mathrm{pdf}_m(\boldsymbol{s}|\boldsymbol{p})$. With such PDF, irradiance from the light source due to the image $I_m(\boldsymbol{p})$ (cf. Equation 4.5) is estimated by generating K_m random samples $\boldsymbol{s}_k$:

$$I_m(\boldsymbol{p}) \simeq \Delta(\boldsymbol{p}) \frac{1}{K_m} \sum_{k=1}^{K_m} \frac{C_m(\boldsymbol{s}_k)\, \Psi_m(\boldsymbol{u}_k \rightarrow \boldsymbol{s}_k)}{|\boldsymbol{s}_k - \boldsymbol{p}|^3} \frac{1}{\mathrm{pdf}_m(\boldsymbol{s}_k|\boldsymbol{p})} \tag{4.6}$$

where $\boldsymbol{u}_k$ is the intersection of the line $\overline{\boldsymbol{s}_k\boldsymbol{p}}$ with the plane $\mathcal{F}$ and $\Psi_m(\boldsymbol{u} \rightarrow \boldsymbol{s})$ is a reconstruction function (cf. Figure 4.3). In the original canned light source model [Heidrich *et al.* 1998], $\Psi_m(\boldsymbol{u} \rightarrow \boldsymbol{s}) = \Phi_m(\boldsymbol{u})$ where Φ_m is a piecewise bilinear interpolation function. In this chapter we use the Goesele et al. [Goesele *et al.* 2003] model $\Psi_m(\boldsymbol{u} \rightarrow \boldsymbol{s}) = \Phi_m(\boldsymbol{u})\, \delta^2 \cos^{-4}\theta$ where Φ_m a piecewise biquadratic function.

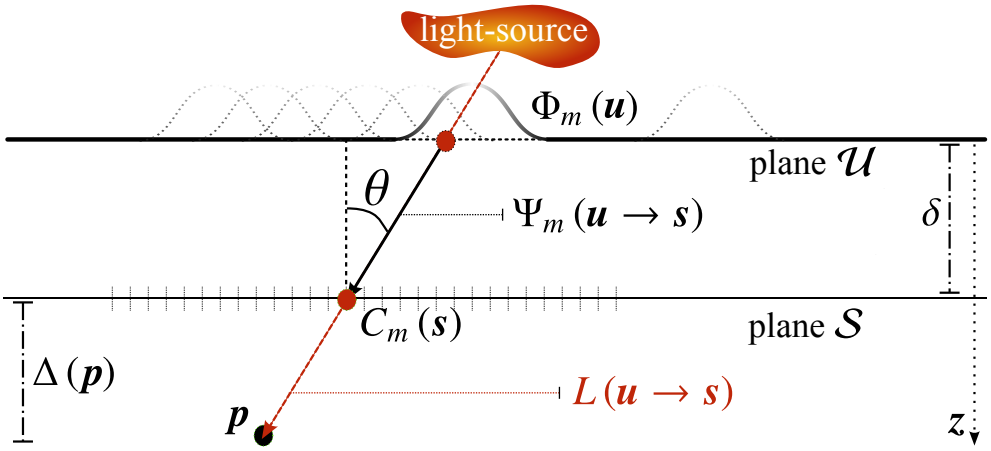


Figure 4.3: *Our new rendering model: Reconstruction function $\Psi_m(\boldsymbol{u} \to \boldsymbol{s})$ for the different luminaire models of Equation 4.2. In the canned light source [Heidrich* et al. *1998], $\Psi_m(\boldsymbol{u} \to \boldsymbol{s}) = \Phi_m(\boldsymbol{u})$ where Φ_m is a piecewise bilinear interpolation function. In the model of Goesele et al. [Goesele* et al. *2003], $\Psi_m(\boldsymbol{u} \to \boldsymbol{s}) = \delta^2 \cos^{-4}\theta\, \Phi_m(\boldsymbol{u})$ where Φ_m is a piecewise biquadratic function.*

Geometric Configuration	
$\boldsymbol{p}$	Position of a shaded point in the scene
δ	Absolute distance between planes $\mathcal{U}$ and $\mathcal{S}$
$\Delta(\boldsymbol{p})$	Absolute distance between $\boldsymbol{p}$ and plane $\mathcal{S}$
$\boldsymbol{u} = (u, v, 0)$	Position on plane $\mathcal{U}$
$\boldsymbol{s} = (s, t, \delta)$	Position on plane $\mathcal{S}$
$\boldsymbol{u} \to \boldsymbol{s}$	Ray passing through $\boldsymbol{u}$ in the direction of $\boldsymbol{s}$
θ	Angle between $\boldsymbol{u} \to \boldsymbol{s}$ and the normal of $\mathcal{S}$
Light Field Models	
$L(\boldsymbol{u} \to \boldsymbol{s})$	Radiance along the ray $\boldsymbol{u} \to \boldsymbol{s}$
$C_m(\boldsymbol{s})$	m^{th} image parameterized on plane $\mathcal{S}$
$\Phi_m(\boldsymbol{u})$	m^{th} basis function on plane $\mathcal{U}$
$[\boldsymbol{u}_m^{min}, \boldsymbol{u}_m^{max}]$	Axis-aligned bounding box support of Φ_m
$[\boldsymbol{s}_m^{min}(\boldsymbol{p}), \boldsymbol{s}_m^{max}(\boldsymbol{p})]$	Axis-aligned bounding box on plane $\mathcal{S}$ position-dependent projection of $[\boldsymbol{u}_m^{min}, \boldsymbol{u}_m^{max}]$
$\Psi_m(\boldsymbol{u} \to \boldsymbol{s})$	m^{th} reconstruction function (based on Φ_m)
Sampling	
$\text{pdf}_m^\star(\boldsymbol{s}\|\boldsymbol{p})$	Optimal position-dependent PDF
$\text{pdf}_m(\boldsymbol{s}\|\boldsymbol{p})$	Position-dependent PDF
$\text{cdf}_m(s\|\boldsymbol{p})$ $\text{cdf}_m(t\|(s, \boldsymbol{p}))$	Corresponding position-dependent CDFs
$\boldsymbol{s}_k, \boldsymbol{u}_k$	k^{th} sample on $\mathcal{S}$ and $\mathcal{U}$

Table 4.1: *Notation table for this chapter*

The optimal PDF $\text{pdf}_m^\star$ is proportional to

$$\text{pdf}_m^\star(\boldsymbol{s}_k|\boldsymbol{p}) \propto C_m(\boldsymbol{s}_k)\, \Psi_m(\boldsymbol{u}_k \to \boldsymbol{s}_k) |\boldsymbol{s}_k - \boldsymbol{p}|^{-3}$$

since it leads to a null variance of the estimator when evaluating $I_m(\boldsymbol{p})$. Since a generic analytical and invertible form of the integral $\text{pdf}_m^\star(\boldsymbol{s}_k|\boldsymbol{p})$ does not exist, a direct use for importance sampling is impossible.

Combining Equation 4.1 with Equation 4.3 leads to the following value of light source emittance $I(\boldsymbol{p})$ that reaches $\boldsymbol{p}$:

$$\begin{aligned} I(\boldsymbol{p}) &= \sum_m I_m(\boldsymbol{p}) \\ I_m(\boldsymbol{p}) &= \int_{s\in\mathcal{J}} \frac{|\boldsymbol{s}-\boldsymbol{u}|^4}{\delta^2} \frac{\Delta(\boldsymbol{p})}{|\boldsymbol{s}-\boldsymbol{p}|^3} C_m(\boldsymbol{s})\,\Phi_m(\boldsymbol{u})\,d\boldsymbol{s}\,. \end{aligned} \tag{4.7}$$

We replace $\Delta(\boldsymbol{p})$ by $|\boldsymbol{s}-\boldsymbol{p}|\cos\theta(\boldsymbol{u},\boldsymbol{s})$ and δ by $|\boldsymbol{u}-\boldsymbol{s}|\cos\theta(\boldsymbol{u},\boldsymbol{s})$ to obtain

$$\begin{aligned} I_m(\boldsymbol{p}) &= \int_{s\in\mathcal{J}} \frac{|\boldsymbol{s}-\boldsymbol{u}|^4\,|\boldsymbol{s}-\boldsymbol{p}|\cos\theta(\boldsymbol{u},\boldsymbol{s})}{|\boldsymbol{u}-\boldsymbol{s}|^2\cos^2\theta(\boldsymbol{u},\boldsymbol{s})\,|\boldsymbol{s}-\boldsymbol{p}|^3} C_m(\boldsymbol{s})\,\Phi_m(\boldsymbol{u})\,d\boldsymbol{s} \\ \Leftrightarrow I_m(\boldsymbol{p}) &= \int_{s\in\mathcal{J}} \frac{|\boldsymbol{s}-\boldsymbol{u}|^2\,|\boldsymbol{s}-\boldsymbol{p}|\cos\theta(\boldsymbol{u},\boldsymbol{s})}{\cos^2\theta(\boldsymbol{u},\boldsymbol{s})\,|\boldsymbol{s}-\boldsymbol{p}|^3} C_m(\boldsymbol{s})\,\Phi_m(\boldsymbol{u})\,d\boldsymbol{s} \\ \Leftrightarrow I_m(\boldsymbol{p}) &= \int_{s\in\mathcal{J}} \frac{|\boldsymbol{s}-\boldsymbol{u}|^2\,|\boldsymbol{s}-\boldsymbol{p}|\cos^2\theta(\boldsymbol{u},\boldsymbol{s})}{\cos^3\theta(\boldsymbol{u},\boldsymbol{s})\,|\boldsymbol{s}-\boldsymbol{p}|^3} C_m(\boldsymbol{s})\,\Phi_m(\boldsymbol{u})\,d\boldsymbol{s} \\ \Leftrightarrow I_m(\boldsymbol{p}) &= \int_{s\in\mathcal{J}} |\boldsymbol{s}-\boldsymbol{p}| \frac{|\boldsymbol{s}-\boldsymbol{u}|^2\cos^2\theta(\boldsymbol{u},\boldsymbol{s})}{\cos^3\theta(\boldsymbol{u},\boldsymbol{s})\,|\boldsymbol{s}-\boldsymbol{p}|^3} C_m(\boldsymbol{s})\,\Phi_m(\boldsymbol{u})\,d\boldsymbol{s} \\ \Leftrightarrow I_m(\boldsymbol{p}) &= \int_{s\in\mathcal{J}} |\boldsymbol{s}-\boldsymbol{p}| \frac{\delta^2}{\Delta^3(\boldsymbol{p})} C_m(\boldsymbol{s})\,\Phi_m(\boldsymbol{u})\,d\boldsymbol{s} \end{aligned}$$

Since $\Delta(\boldsymbol{p})$ and δ do not depend on $\boldsymbol{s}$, we finally obtain the Equation 4.5:

$$I_m(\boldsymbol{p}) = \frac{\delta^2}{\Delta^3(\boldsymbol{p})} \int_{s\in\mathcal{J}} |\boldsymbol{s}-\boldsymbol{p}|\; C_m(\boldsymbol{s})\;\Phi_m(\boldsymbol{u})\;d\boldsymbol{s} \tag{4.8}$$

$$I_m(\boldsymbol{p}) = \Delta(\boldsymbol{p}) \int_{\mathcal{J}} C_m(\boldsymbol{s})\,\Psi_m(\boldsymbol{u}\rightarrow\boldsymbol{s}) \frac{1}{|\boldsymbol{s}-\boldsymbol{p}|^3}\,d\boldsymbol{s}\,. \tag{4.9}$$

Consequently, we have to find a pdf_m that mimics closely $\text{pdf}_m^\star$ while achieving a low variance. For this purpose, we need the following **two properties**: (i) the generated samples do not introduce any bias in the estimator that is, pdf_m has to ensure that it generates random samples at any position where $\text{pdf}_m^\star(\boldsymbol{s}_k|\boldsymbol{p}) \neq 0$, and (ii) each sample must convey some energy that is, $|\boldsymbol{s}_k-\boldsymbol{p}|^{-3}\,C_m(\boldsymbol{s}_k)\,\Psi_m(\boldsymbol{u}_k\rightarrow\boldsymbol{s}_k) \neq 0$. (i) and (ii) are achieved by guaranteeing that:

$$\text{pdf}_m(\boldsymbol{s}_k|\boldsymbol{p}) \neq 0 \Leftrightarrow \text{pdf}_m^\star(\boldsymbol{s}_k|\boldsymbol{p}) \neq 0\,.$$

Since $|\boldsymbol{s}_k-\boldsymbol{p}|^{-3} > 0$, and since for existing luminaire models [Heidrich *et al.* 1998][Goesele *et al.* 2003], with the definitions introduced in Figure 4.3, $\Psi_m(\boldsymbol{u}_k\rightarrow\boldsymbol{s}_k) \neq 0 \Leftrightarrow \Phi_m(\boldsymbol{u}_k) \neq 0$,

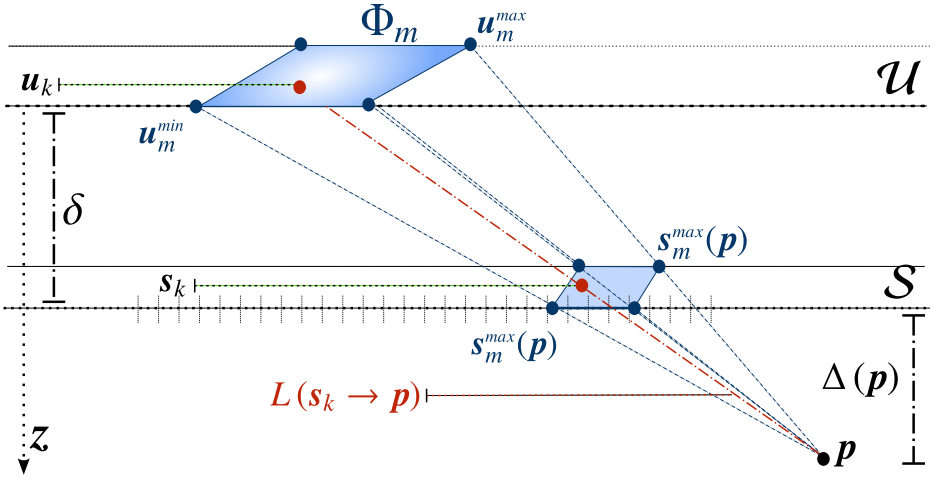


Figure 4.4: *Projection support of Φ_m on $\mathcal{I}$ for a given position $\boldsymbol{p}$. Φ_m is strictly positive over a 2D axis-aligned box bounded by $\boldsymbol{u}_m^{min}$ and $\boldsymbol{u}_m^{max}$. The projection of this axis-aligned bounding box on $\mathcal{I}$ is still an axis-aligned box bounded by $\boldsymbol{s}_m^{min}(\boldsymbol{p})$ (resp. $\boldsymbol{s}_m^{max}(\boldsymbol{p})$), which is the intersection of the plane $\mathcal{I}$ with the line $\overline{\boldsymbol{p}\,\boldsymbol{u}_m^{min}}$ (resp. $\overline{\boldsymbol{p}\,\boldsymbol{u}_m^{max}}$).*

it implies

$$\mathrm{pdf}_m(\boldsymbol{s}_k|\boldsymbol{p}) \neq 0 \Leftrightarrow C_m(\boldsymbol{s}_k) \neq 0 \;\text{ and }\; \Phi_m(\boldsymbol{u}_k) \neq 0\,.$$

The special case when $\boldsymbol{p}$ is on $\mathcal{I}$ is explained in Section 4.3.4.

4.3.2 Precomputed CDFs

The condition $C_m(\boldsymbol{s}_k) \neq 0$ is fullfilled by computing samples according to the images C_m. This corresponds to the following CDFs for s (i.e., the 1D $\mathrm{c}df_m(s)$) and for t knowing s (i.e., the 2D $\mathrm{c}df_m(t|s)$):

$$\begin{aligned}
\mathrm{c}df_m(s) &= \frac{\int_{-\infty}^{s}\int_{-\infty}^{+\infty} C_m(\sigma,\tau)\mathrm{d}\sigma\,\mathrm{d}\tau}{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} C_m(\sigma,\tau)\mathrm{d}\sigma\,\mathrm{d}\tau} \\
\mathrm{c}df_m(t|s) &= \frac{\int_{-\infty}^{t} C_m(s,\tau)\mathrm{d}\tau}{\int_{-\infty}^{+\infty} C_m(s,\tau)\mathrm{d}\tau}
\end{aligned} \tag{4.10}$$

where $\boldsymbol{s} = (s,t,\delta)$ (cf. Figure 4.3). Assuming C_m is a simple image that is, a piecewise constant and positive function, $\mathrm{c}df_m(s)$ is a 1D piecewise linear function and $\mathrm{c}df_m(t|s)$ is a 2D piecewise function, linear in t and constant in s. Therefore, they can be **exactly represented as precomputed 1D and 2D tables**, relying on hardware linear interpolation. Such a derivation may be easily extended to higher order reconstructions for images like piecewise bilinear ones [Heidrich *et al.* 1998].

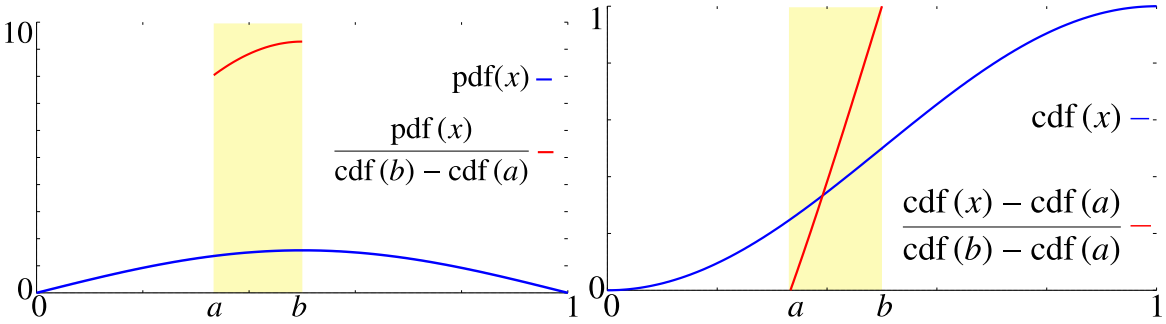


Figure 4.5: *The original pdf (**Left**, in blue) and its corresponding cdf (**Right**, in blue) are defined on the interval* $[0, 1]$. *Restricting the sampling to the interval* $[a, b]$ *is done according to a new pdf (**Left**, in red) that is a rescaled version of the original one. The corresponding new cdf (**Right**, in red) is obtained by an affine transformation.*

4.3.3 Restricted CDFs

We also do not want to generate samples for which $\Phi_m(\boldsymbol{u}_k) = 0$ because they do not convey energy. Since Φ_m are defined as the product of two 1D functions [Goesele *et al.* 2003; Heidrich *et al.* 1998], the validity domain of samples (i.e., $\Phi_m(\boldsymbol{u}_k) \neq 0$) is an axis-aligned bounding box defined by $\boldsymbol{u}_m^{min} < \boldsymbol{u}_k < \boldsymbol{u}_m^{max}$. According to their definition, $\boldsymbol{u}_k$, $\boldsymbol{s}_k$ and $\boldsymbol{p}$ are aligned (cf. Figure 4.4) leading to the position-dependent condition on samples:

$$\boldsymbol{s}_m^{min}(\boldsymbol{p}) < \boldsymbol{s}_k < \boldsymbol{s}_m^{max}(\boldsymbol{p})$$

$$\text{with} \quad \begin{pmatrix} \boldsymbol{s}_m^{min}(\boldsymbol{p}) \\ \boldsymbol{s}_m^{max}(\boldsymbol{p}) \end{pmatrix} = \frac{\delta}{\delta + \Delta(\boldsymbol{p})} \begin{pmatrix} \boldsymbol{p} - \boldsymbol{u}_m^{min} \\ \boldsymbol{p} - \boldsymbol{u}_m^{max} \end{pmatrix} \tag{4.11}$$

where $\boldsymbol{s}_m^{min}(\boldsymbol{p})$ (resp. $\boldsymbol{s}_m^{max}(\boldsymbol{p})$) is the intersection of line $\overline{\boldsymbol{p}\,\boldsymbol{u}_m^{min}}$ (resp. $\overline{\boldsymbol{p}\,\boldsymbol{u}_m^{max}}$) with $\mathcal{I}$. $\boldsymbol{s}_m^{min}$ and $\boldsymbol{s}_m^{max}$ represent the axis-aligned bounding box corners of the restricted sampling domain.

Our main idea is to restrict the sample generation to this domain. This is achieved by a simple position-dependent affine transformation of precomputed CDFs.

To illustrate the core idea, we consider the 1D case illustrated in Figure 4.5. Given a known PDF, denoted pdf, defined on $[0, 1]$ and its corresponding CDF, denoted cdf, it is very easy to restrict the sampling to a sub-interval $[a, b]$. The new sampling strategy corresponds to a new conditional PDF defined on $[a, b]$ which is a rescaled version of the original PDF:

$$\text{pdf}(x|x \in [a, b]) = \frac{\text{pdf}(x)}{cdf(b) - cdf(a)}.$$

The corresponding conditional CDF, also defined on $[a, b]$, is obtained by a simple affine transformation:

$$cdf(x|x \in [a, b]) = \frac{cdf(x) - cdf(a)}{cdf(b) - cdf(a)}.$$

Figure 4.6: *Restricted sampling: restrict areas are shown as red rectangles. Instead of sampling on the whole image, restrict areas are computed dependent on the position of a ray. Blue points are random samples.*

This means that for a given 1D CDF, we can compute exactly its restriction to a sub-interval from its definition domain.

As for higher dimensions [Fishman 1996], the same process may be applied to each conditional 1D CDF. The only condition is that the restricted domain for the pdf is an axis-aligned bounding box. Fortunately, it applies to our case: the restricted domain is bounded by $\boldsymbol{s}_m^{min}(\boldsymbol{p})$ and $\boldsymbol{s}_m^{min}(\boldsymbol{p})$. On this domain, the CDFs are transformed to:

$$\begin{aligned} cdf_m(s|\boldsymbol{p}) &= \frac{cdf_m(s) - cdf_m(s_m^{min}(\boldsymbol{p}))}{cdf_m(s_m^{max}(\boldsymbol{p})) - cdf_m(s_m^{min}(\boldsymbol{p}))}, \\ cdf_m(t|(s,\boldsymbol{p})) &= \frac{cdf_m(t|s) - cdf_m\left(t_m^{min}(\boldsymbol{p})\middle| s\right)}{cdf_m\left(t_m^{max}(\boldsymbol{p})\middle| s\right) - cdf_m\left(t_m^{min}(\boldsymbol{p})\middle| s\right)}. \end{aligned} \tag{4.12}$$

Importance sampling based on these CDFs generates samples that have the two properties (i) $C_m(\boldsymbol{s}_k) \neq 0$ is preserved and (ii) $\Phi_m(\boldsymbol{u}_k) \neq 0$ due to the restriction. Except on the plane $\mathcal{I}$, we thus ensure the two conditions to have an **unbiased and efficient** sampling strategy.

4.3.4 Special case: $\boldsymbol{p}$ on image plane $\mathcal{I}$

We have pointed out that the strategy is not defined for $\boldsymbol{p} \in \mathcal{I}$. Note that due to the lighting model, either the choice of plane $\mathcal{I}$ or $\mathcal{F}$ will lead to such a discontinuity. However, the same intensity as in Equation 4.7 can be computed by integrating on the plane $\mathcal{F}$:

$$I(\boldsymbol{p}) = (\Delta(\boldsymbol{p}) + \delta) \int_{\mathcal{F}} L(\boldsymbol{u} \to \boldsymbol{p}) \frac{1}{|\boldsymbol{u} - \boldsymbol{p}|^3} d\boldsymbol{u}.$$

On the image plane $\mathcal{I}$ where $\Delta(\boldsymbol{p}) = 0$ we thus have

$$I(\boldsymbol{s}) = \delta \int_{\mathcal{F}} L(\boldsymbol{u} \to \boldsymbol{s}) \frac{1}{|\boldsymbol{u} - \boldsymbol{s}|^3} d\boldsymbol{u}.$$

By using the same approach for obtaining Equation 4.5, we obtain

$$I_m(\boldsymbol{s}) = \delta\, C_m(\boldsymbol{s}) \int_{\mathcal{F}} \Psi_m(\boldsymbol{u} \to \boldsymbol{s}) \frac{1}{|\boldsymbol{u}-\boldsymbol{s}|^3} \mathrm{d}\boldsymbol{u}\,. \tag{4.13}$$

With the same approach described in the introduction of the current Section 4.3.1, we can demonstrate that a uniform sampling $\boldsymbol{u}_k$ on the support of Φ_m leads to an efficient and unbiased estimator.

4.3.5 Real-Time Restricted Sampling

The PDF corresponding to the sampling strategy introduced in the previous section is obtained by differentiation of the restricted CDFs: on the axis-aligned box bounded by $\boldsymbol{s}_m^{min}(\boldsymbol{p})$ and $\boldsymbol{s}_m^{max}(\boldsymbol{p})$, pdf_m is defined by:

$$\mathrm{pdf}_m(\boldsymbol{s}_k|\boldsymbol{p}) = \frac{C_m(\boldsymbol{s})}{A_m(\boldsymbol{p})} \text{ with } A_m(\boldsymbol{p}) = \int_{\boldsymbol{s}_m^{min}(\boldsymbol{p})}^{\boldsymbol{s}_m^{max}(\boldsymbol{p})} C_m(\boldsymbol{s})\,\mathrm{d}\boldsymbol{s}\,. \tag{4.14}$$

As demonstrated in Section 4.3.1, this PDF has the required properties to mimic closely the behavior of the optimal one $\mathrm{pdf}_m^\star$. The resulting estimator defined in Equation 4.6 simplifies to

$$I_m(\boldsymbol{p}) = \frac{A_m(\boldsymbol{p})\ \Delta(\boldsymbol{p})}{K_m} \sum_{k=1}^{K_m} \Psi_m(\boldsymbol{u}_k \to \boldsymbol{s}_k)\ \frac{1}{|\boldsymbol{s}_k-\boldsymbol{p}|^3}\,. \tag{4.15}$$

Estimating each I_m using the same number of samples is straightforward and can be easily parallelized. However, if $I_m(\boldsymbol{p}) = 0$, K_m useless samples will still be evaluated leading to unwanted processing. With our sampling strategy $I_m(\boldsymbol{p}) = 0$ if and only if $A_m(\boldsymbol{p}) = 0$. We thus use $A_m(\boldsymbol{p})$ to balance the number of samples among the different images. By introducing

$$K_m(\boldsymbol{p}) = \left\lfloor \frac{A_m(\boldsymbol{p})}{A(\boldsymbol{p})} K \right\rceil \quad \text{with} \quad A(\boldsymbol{p}) = \sum_m A_m(\boldsymbol{p}) \tag{4.16}$$

where $\lfloor x \rceil$ denotes the closest integer approximation of x and K is a global control on the number of samples, we estimate $I_m(\boldsymbol{p})$ as follow:

$$I_m(\boldsymbol{p}) = \frac{A_m(\boldsymbol{p})\ \Delta(\boldsymbol{p})}{K_m(\boldsymbol{p})} \sum_{k=1}^{K_m(\boldsymbol{p})} \Psi_m(\boldsymbol{u}_k \to \boldsymbol{s}_k)\ \frac{1}{|\boldsymbol{s}_k-\boldsymbol{p}|^3}. \tag{4.17}$$

Finally $I(\boldsymbol{p})$ is estimated by accumulating the computed values of $I_m(\boldsymbol{p})$. Readers may note that the balancing strategy is still dependent on the 3D scene position $\boldsymbol{p}$ due to the use of $A_m(\boldsymbol{p})$. The complete sampling strategy is therefore completly adapted to each position dynamically. It is also worth noticing that when $A_m(\boldsymbol{p}) = 0$ for all m no samples will

be generated. This corresponds to regions of the scene that may not be reached by rays emitted from the luminaire (without taking into account visibility). Finally, for the special case where $\boldsymbol{p} = \boldsymbol{s}$, the number of samples is trivially balanced according to $C_m(\boldsymbol{s})$.

4.4 Generic Shading Estimator

Until now, we have assumed that $C_m(\boldsymbol{s})$ is a scalar positive value. However, we can easily extend our approach to colored images where $\boldsymbol{C}_m(\boldsymbol{s})$ is a three-component vector. For this purpose, we set $C_m(\boldsymbol{s})$ to be the luminance of $\boldsymbol{C}_m(\boldsymbol{s})$ and we store an additional texture per image containing $\boldsymbol{c}_m(\boldsymbol{s}) = \boldsymbol{C}_m(\boldsymbol{s})/C_m(\boldsymbol{s})$. During the shading estimation, the intensity conveyed by a sample $\boldsymbol{s}_k$ is scaled by $\boldsymbol{c}_m(\boldsymbol{s}_k)$. Compared to the solution described in this chapter, for a light field of M images, this approach requires M additional 2D RGB textures to represent the light source color.

Finally, combining BRDF ρ_k and visibility v_k evaluated for each sample $\boldsymbol{s}_k$ with luminaire color $\boldsymbol{c}_m(\boldsymbol{s}_k)$ leads to the following **final estimator** for reflected radiance:

$$I_m(\boldsymbol{p}) = \frac{A_m(\boldsymbol{p})\,\Delta(\boldsymbol{p})}{K_m(\boldsymbol{p})} \sum_{k=1}^{K_m(\boldsymbol{p})} \rho_k\, v_k\, \boldsymbol{c}_m(\boldsymbol{s}_k)\, \frac{\Psi_m(\boldsymbol{u}_k \rightarrow \boldsymbol{s}_k)}{|\boldsymbol{s}_k - \boldsymbol{p}|^3} \tag{4.18}$$

where $K_m(\boldsymbol{p})$ is given in Equation 4.16. This estimator does not introduce any sampling bias or spatial bias between pixels as long as a different random sequence is used for each pixel and/or position $\boldsymbol{p}$.

The generation of the $\boldsymbol{s}_k$ samples relies on the inversion of the CDF functions. Instead of using a brute force binary search, we take advantage of the fact that our CDFs are piecewise linear and monotonic functions and we use a modified version of the secant method. We modify it such that the recursion stops when the search interval corresponds to a linear part of the function (i.e., between two neighboring pixels). Due to the CDFs properties, our modified secant method will always stop with the exact solution.

The faster convergence is illustrated in Figure 4.7 where for a given search depth, our adaptation of the algorithm is closer to the converged solution than the classical binary search.

4.5 Dedicated GPU Implementation

In this section we explain the GPU specifics of our sampling statregy when used for direct lighting estimation. Our implementation uses (but not limited to) OpenGL and OpenCL. For each C_m image we precompute its associated CDFs (cf. Equation 4.12). These CDFs

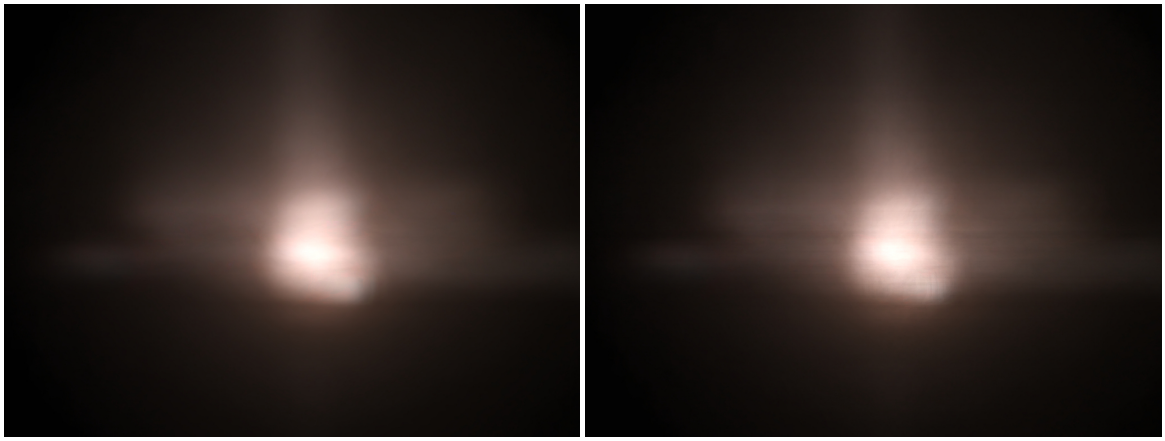


Figure 4.7: *Comparison of (**Left**) our new binary search with (**Right**) the classical binary search for the same search depth. For a search depth of three, the new binary search gives a qualitatively better results (0.89 vs. 0.93 Lab error). Both results are computed in 84ms using 200 samples per pixel (spp). The mean Lab error is computed against a reference solution, shown in Figure 4.15, computed with a precomputed light importance sampling strategy using 25M spp.*

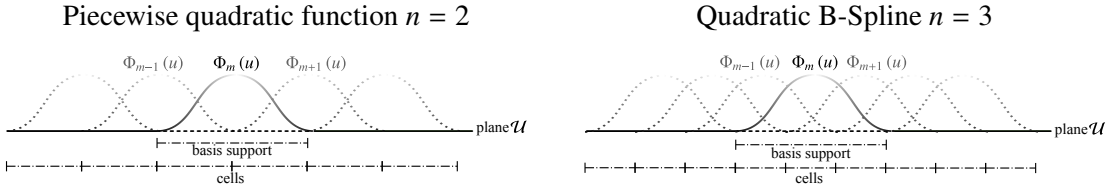


Figure 4.8: *Definition of cells for 1D piecewise quadratic functions Φ_m. **(Left)** For piecewise quadratic function (used in [Goesele* et al. *2003]) the basis support overlaps two cells whereas **(Right)** with quadratic B-Spline the basis support overlaps three cells. Each basis support overlaps n cells that are shared with neighboring basis. For M basis functions, the supporting plane is divided in $M + n - 1$ tiles.*

are stored as textures of the same resolution. $cdf(s)$ (resp. $cdf(t|s)$) is stored as a 1D (resp. 2D) floating textures of the same resolution. Finally, we also transform each C_m into a summed area table sat_m to speed up the computation of $A_m(\boldsymbol{p})$.

4.5.1 Per-frame Rendering

Our rendering pipeline is based on deferred shading. For each frame, the sampling and rendering process is divided into four steps:

Step 0 - OpenGL A first G-Buffer pass is done where we construct and store the pixels' positions and normals into two floating-point textures. The shadow maps are then computed and stored for later visibility approximation (cf. Section 4.5.2).

Step 1 - OpenCL We perform one pass per image C_m to compute $A_m(\boldsymbol{p})$ and we accumu-

late them in a dedicated floating-point buffer to estimate $A(\boldsymbol{p})$. More precisely, for each pixel, we compute in parallel the boundaries $\boldsymbol{s}_m^{min}(\boldsymbol{p})$ and $\boldsymbol{s}_m^{max}(\boldsymbol{p})$ (Equation 4.11) and then use sat_m to evaluate efficiently $A_m(\boldsymbol{p})$ (Equation 4.14).

Step 2 - OpenCL We perform one pass per image C_m and compute per pixel shading. For each pixel in parallel, we recompute $A_m(\boldsymbol{p})$ as in step 1 and use the previously computed $A(\boldsymbol{p})$ to determine the number of samples $K_m(\boldsymbol{p})$. We then generate the random samples $\boldsymbol{s}_k$ according to the $cdf_m(s|\boldsymbol{p})$ and $cdf_m(t|s,\boldsymbol{p})$. For each generated sample, we accumulate its lighting contribution to the pixel by multiplying it to the BRDF and visibility terms. The complete estimator is detailed in the upcoming Equation 4.18.

Step 3 - OpenGL We perform a simple dynamic tone mapping using the exponential operator [Reinhard *et al.* 2010].

4.5.2 Efficient Shadow Approximation

In theory, visibility has to be evaluated for each light sample $\boldsymbol{s}_k$ but it would be too slow for interactive rendering because they are hundreds of dynamic light sources per pixel. Therefore, we introduce a new and fast shadow algorithm that approximates the visibility.

Our approximation is based on the properties of the reconstruction basis functions Φ_m . As illustrated in Figure 4.8, their 1D support overlaps n cells that are shared with neighboring basis functions. For the two-dimension case, each basis support overlaps n^2 cells. For each cell, we select a reference light position from which the shadow map is generated. For a light field composed of $M = W \times H$ images, we generate $(W+n-1)\times(H+n-1)$ shadow maps. Each shadow map is shared between n^2 neighboring basis functions.

To compute the reference light positions, we first select a position $\boldsymbol{s}_m = (s_m, t_m)$ on each C_m where $cdf_m(s_m)$ and $cdf_m(t_m|s_m)$ are both equals to 0.5. This roughly corresponds to the center of the high-intensity region of C_m. The reference light position for a cell is computed as the average of the $\boldsymbol{s}_m$ on the corresponding images C_m whose associated basis functions overlap on the cell. As shown in Figure 4.9, our visibility algorithm introduces approximations compared to the ray-traced reference solution. However, these approximations are visually coherent.

Remember that our rendering pipeline executes one pass for each C_m. For each pass, instead of computing one visibility test per light sample $\boldsymbol{s}_k$, we compute one average visibility $v_m(\boldsymbol{p})$. The later is computed as the average of the shadow tests on $\boldsymbol{p}$ against the n^2 shadow maps from cells overlapping the basis function Φ_m.

Figure 4.9: *Comparison of our (**Left**) approximated visibility to (**Right**) a reference solution computed with a ray-tracer. The light source used the* `car1` *data and our algorithm is implemented with 49 shadow maps (*192×192*). As confirmed by the mean Lab error (1.2) the visual difference between the two images is low.*

4.5.3 Random Sequence Optimizations

For efficiency purpose when implementing on GPU, we use the same random sequence for each pixel. This does not introduce any per-pixel bias but only a spatial bias between pixels and thus reduces the spatial noise. Furthermore, it also improves the cache access coherency of the GPU implementation. To adjust the tradeoff between speed and spatial bias, interleaved sampling [Keller & Heidrich 2001] may be introduced. However, as shown in Figure 4.13 and Figure 4.15, the current simple strategy already gives very good results.

Since we need to distribute different number of samples among the different light field images, we choose the Halton sequence because all prefixes of the sequence are well distributed over the domain. As shown in Figure 4.10, due to their lower discrepancy property (cf. [Niederreiter 1992]), when used with a small number of samples, Halton sequence gives better result than the one generated with Mersenne Twister.

Performance improvements are also obtained by limiting the number of samples per pixel and per image $K_m(\boldsymbol{p})$ to a maximum value $K_{\max}$. As shown in Figure 4.11, this strategy reduces the total number of samples without introducing any bias and with a low impact on the final quality.

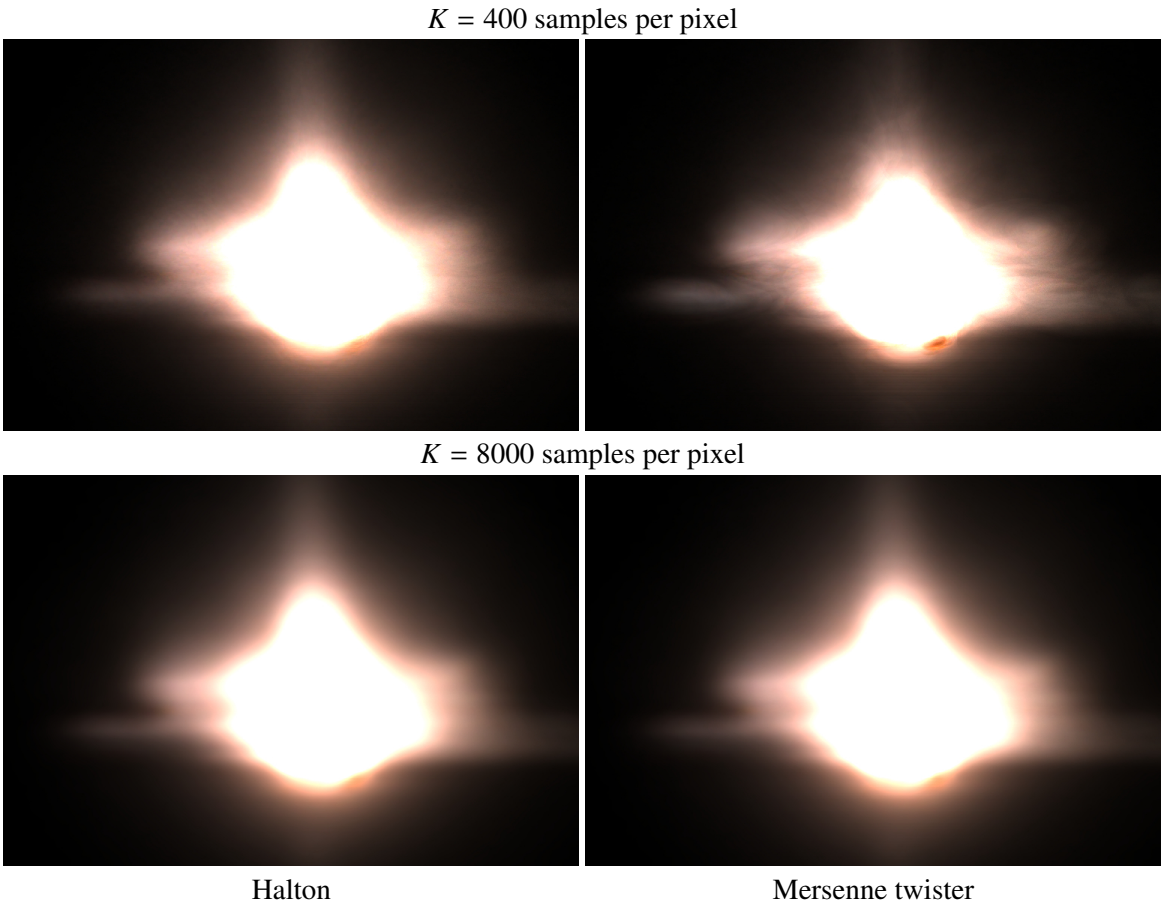


Figure 4.10: *Influence of the random generator according to the number of samples per pixel on the* `car1` *data (cf. Table 4.2).* ***(Top)*** *For a low number of samples, Halton sequence leads more rapidly to better results.* ***(Bottom)*** *When using a large number of samples, the Mersenne twister [Matsumoto & Nishimura 1998] is used as reference due to its recognized quality and long period.*

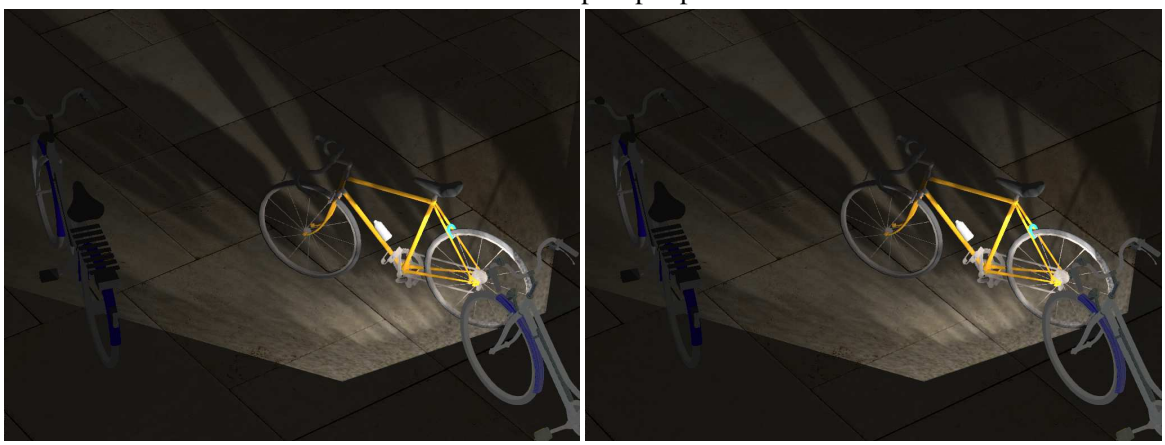
4.6 Results and Discussion

We have tested our solution with three available light data. One of them is the bike light from [Goesele *et al.* 2003]. The other two are new acquired car light source data. The specification of them is shown in Table 4.2.

We have implemented a slightly modified setup. Instead of the original ad-hoc piecewise quadratic function for reconstruction basis $\{\Phi_{mn}\}$ which is not a partition of unity and may thus introduce unwanted oscillations, we prefer to use 2D quadratic B-Splines. The previous setup uses two physical filters to represent the positive and negative part of the dual function. We prefer to directly use the basis function as filter:

$$E_{ij} = \left\langle E, \Phi_{ij} \right\rangle . \tag{4.19}$$

K = 200 samples per pixel



with: 13.5fps without: 11.6fps

Figure 4.11: *Influence of the maximum number of samples. To increase the rendering efficiency, the number of samples per pixel and per images $K_m(\boldsymbol{p}_i)$ may be bounded to a maximum value K_{max}. In this example with the* `bike` *data (cf. Table 4.2), setting $K_{max} = 7$ does not introduce large visual differences since the maximum Lab error is only* 23 *(*0.63 *mean error) between the two images.*

name	description	Nb. - res. of images	basis
`bike`	bike	9×7 - 300×300	quad. 15mm
`car1`	car	5×5 - 256×256	spline 49.5mm
`car2`	car	11×9 - 256×256	spline 49.5mm

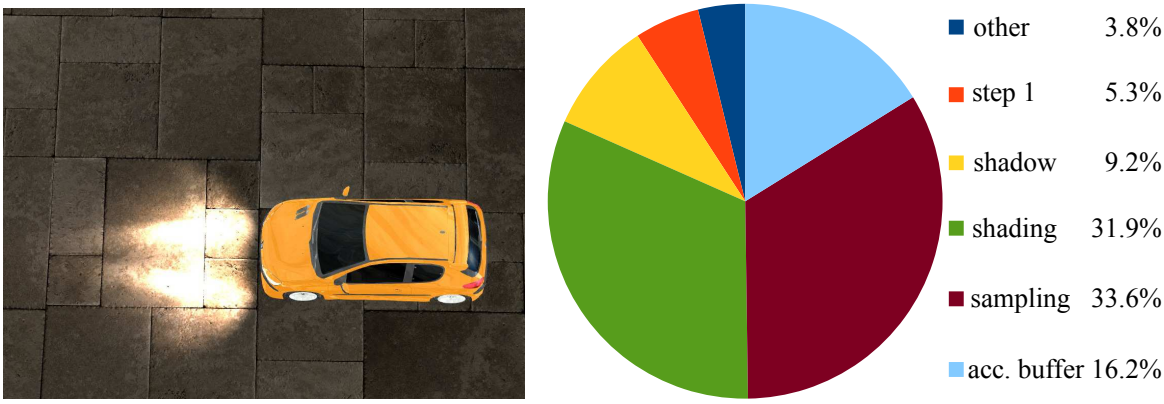
Table 4.2: *Light field data and their associated types for the Φ_m functions used in this chapter. The* `bike` *data are from Goesele et al. [Goesele* et al. *2003] and use their dedicated quadratic functions whereas* `car1` *and* `car2` *are new data that use quadratic B-Splines.*

Since a B-Spline is positive, only one filter is fabricated. The final C_{mn} are solution of the following linear equation that depends on acquired images E_{ij}:

$$E_{ij} = \sum_{m,n} C_{mn} \left\langle \Phi_{ij}, \Phi_{mn} \right\rangle . \tag{4.20}$$

These linear equations are obtained by minimizing the difference $|\widetilde{E} - E|^2$ between the incident energy E and its reconstruction $\widetilde{E}$ on the image plane. After inverting the matrix which coefficient are $\left\langle \Phi_{ij}, \Phi_{mn} \right\rangle$, the process is highly parallelizable and take only 3 minutes for 345 images with resolution 256×256 using a workstation with two processors Xeon E5645 (6 cores, 12 threads, 2.4 GHz).

All presented results of using these three light sources are rendered at 1024×768 resolution, using a GTX 580 with 1.5GB on a workstation with an Intel Core i7 920 with 6 GB. The companion video shows the interactive frame rates, ranging from 7 to 15fps, of



200 spp for two headlights
9fps

Cost decomposition

Figure 4.12: *Repartition of the rendering time for two headlights using* `car2` *data. "Est. A" stands for the computation of each* $A_m(\boldsymbol{p})$ *accumulated to get* $A(\boldsymbol{p})$ *in step 1 of section 4.5.1. "sampling" corresponds to the sampling cost of step 2. "other" includes buffer swapping, memory and context sharing between OpenCL and OpenGL. "acc. buffer" stands for the cost of accumulating the contribution of each image* C_m.

our GPU implementation. The size of the different 3D models are 200K polygons for the bike, 7000 polygons for the car and 70K for Sponza (cf. Figures 4.9).

The precomputation time (CDFs, *sat* and reference light positions for shadow maps) for all light field data listed in Table 4.2 is quite low: 513ms for `bike`, 129ms for `car1` and 527ms for `car2`. Regarding the GPU memory footprint, our technique requires storing the G-Buffers (24MB) and different textures representing the light field and its CDF. More precisely, for a light field of M images the number of floating-point textures stored on the GPU memory is: M 2D textures for $cdf_m(t|s)$, M 1D textures for $cdf_m(s)$ and M 2D textures for sat. For the `car2` data the whole light field and colored $\boldsymbol{C}_m(\boldsymbol{s})$ textures (cf. Section 4.4) sum to 123MB.

To illustrate the advantage of our sampling approach we compare the image quality between **precomputed light importance sampling** and our dynamic sampling strategies both applied on the image plane. Since Precomputed Light Importance sampling is a classical approach when one needs to sample many light sources with different energy, it is an appropriate reference solution to compare our approach with.

As shown in Figure 4.13, our dynamic approach achieves a drastically higher quality with 1000 times less samples per pixel and is almost two orders of magnitude faster (e.g., 62ms vs 4821ms) than the precomputed approach. Moreover, as shown on the right image of Figure 4.13, our technique quickly converges toward the reference solution (63M samples, max search depth 15) with only 1000 samples and is several order of magnitude faster.

Comparisons between a uniform approach and ours are shown in Figure 4.14 with the

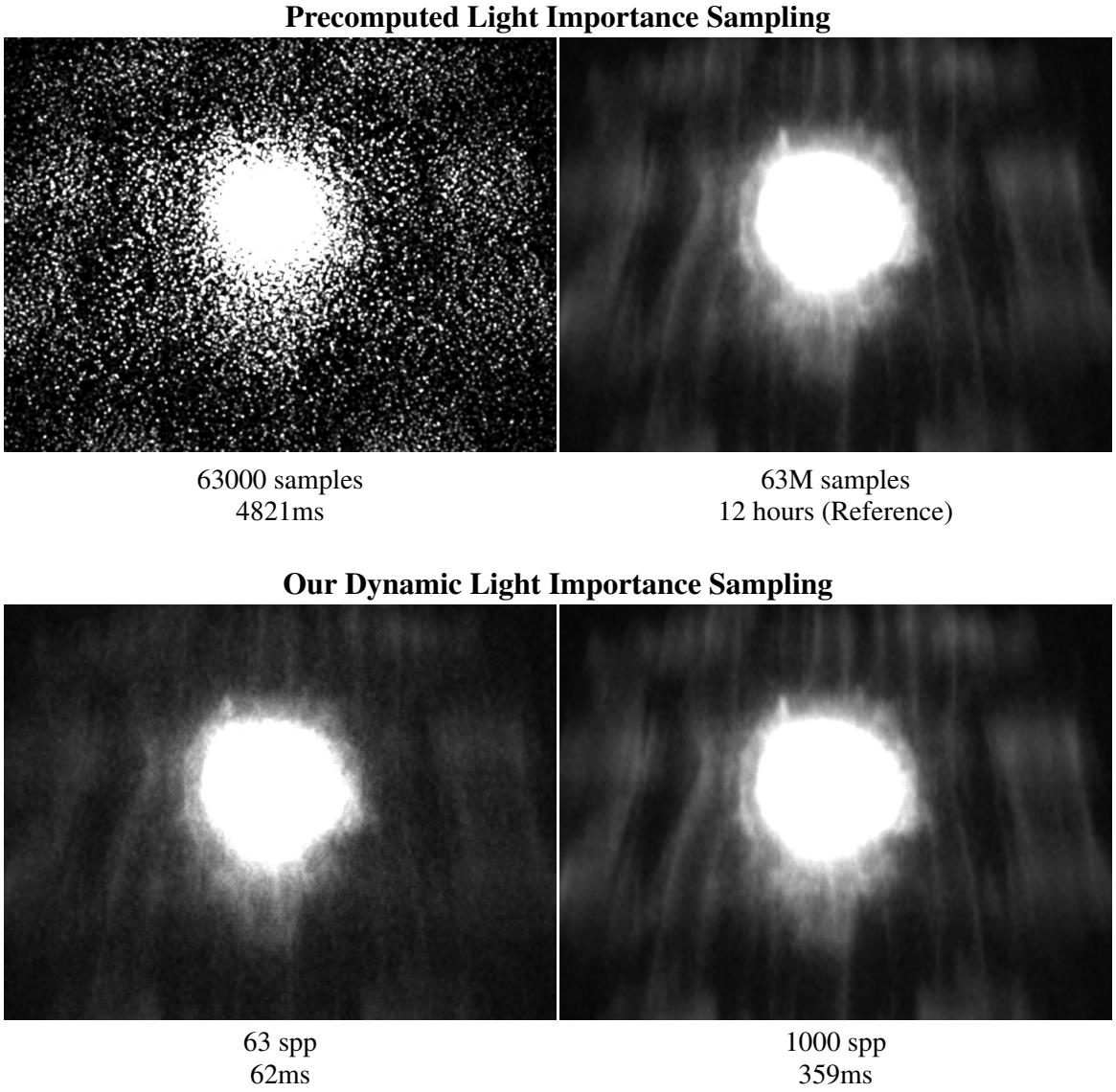


Figure 4.13: *Convergence speed comparisons of precomputed light importance samping versus our approach with per-pixel light importance sampling for the* `bike` *headlight data. We compute mean Lab errors against the image, which is used as reference solution, computed with 63M samples on the texture plane. With the precomputed strategy, samples are generated from global CDF functions (cf. Equation 4.10) and every pixel is shaded with all precomputed samples. On the contrary, with our approach, for each pixel, samples are dynamically generated on a restricted area and distributed among the light field images. Consequently, with less samples per pixel (63 vs 63000), our technique generates a result with better quality (2.3 vs 15 mean Lab error) and is almost two order of magnitude faster (62ms vs 4821ms). Furthermore, our technique achieves almost the quality (0.34 Lab error) of the reference solution at interactive frame rate (359ms) with only 1000 samples.*

bike head data, for the uniform solution, we generate the same number of samples from each texture image and sample the restricted area uniformly on the image plane without taking into account the CDFs. This approach is equivalent to sample uniformly the support

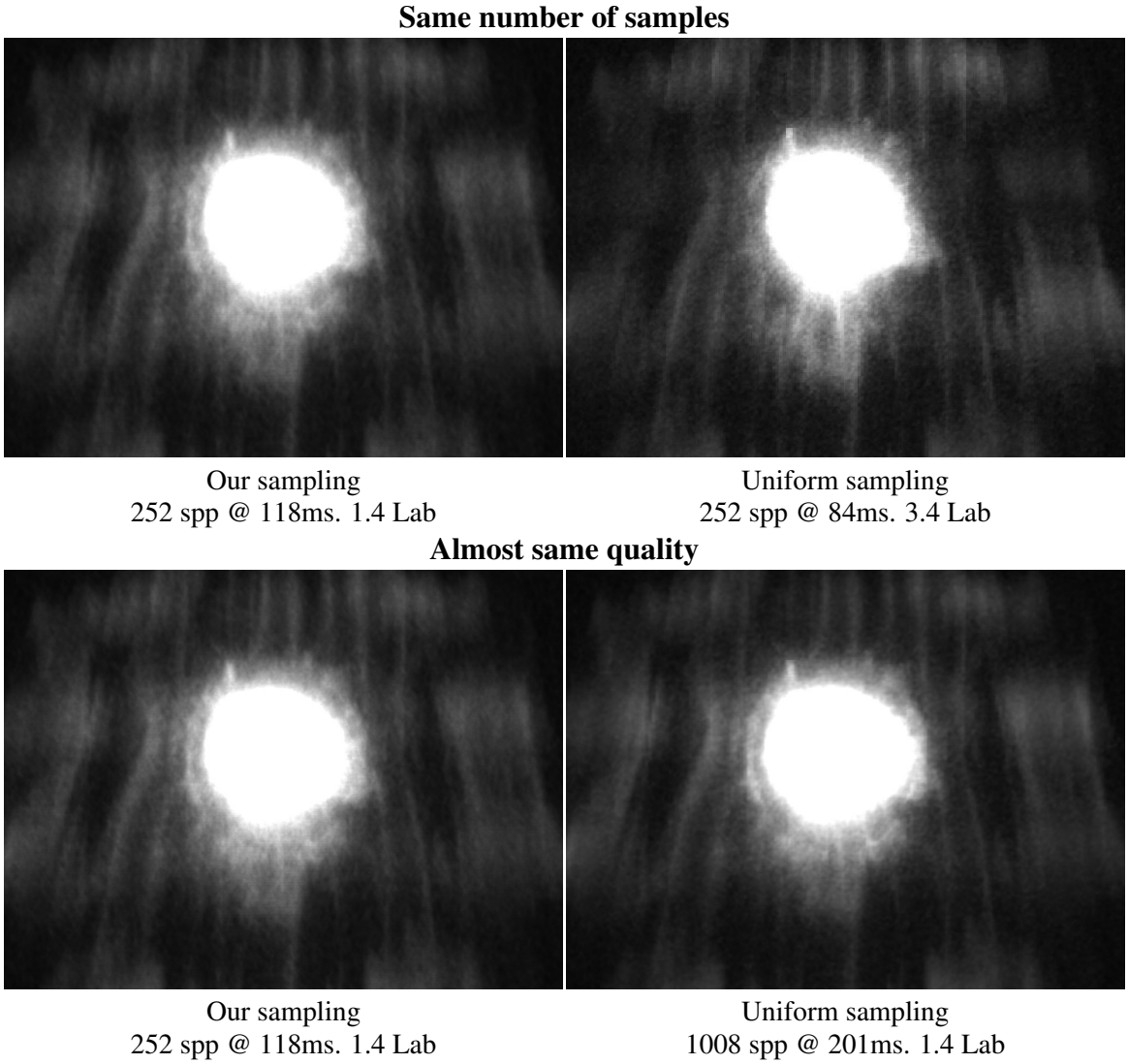


Figure 4.14: *Qualitative comparisons of uniform sampling versus light importance sampling for the* `bike` *headlight data. Both strategies are applied on the restricted area and the mean Lab errors are computed against the reference solution of Figure 4.13. With our importance sampling approach, for each pixel, samples are selected using the restricted CDFs, and therefore, the number of samples varies for each basis support. (**Top**). For the same number of samples our technique is slower than the uniform approach but its Lab error is 2.4 times better (1.4 vs 3.4 Lab), thus showing that our technique converges faster. This is also illustrated in the (**Bottom**) side of the figure where, for almost the same quality (1.4 Lab error), our approach requires 4 times (252 vs 1008) less samples.*

of each basis. As illustrated in the left part of Figure 4.14, when comparing against the reference solution of Figure 4.13, for the same number of samples (252) our approach gives a Lab error 2.4 times smaller. Furthermore, for almost the same quality (1.4 Lab error), our approach requires four times less samples than the uniform approach. Consequently, our dynamic light importance sampling is more suitable when combined with complex BRDF since it requires less shading samples. Finally, convergence speed of our technique

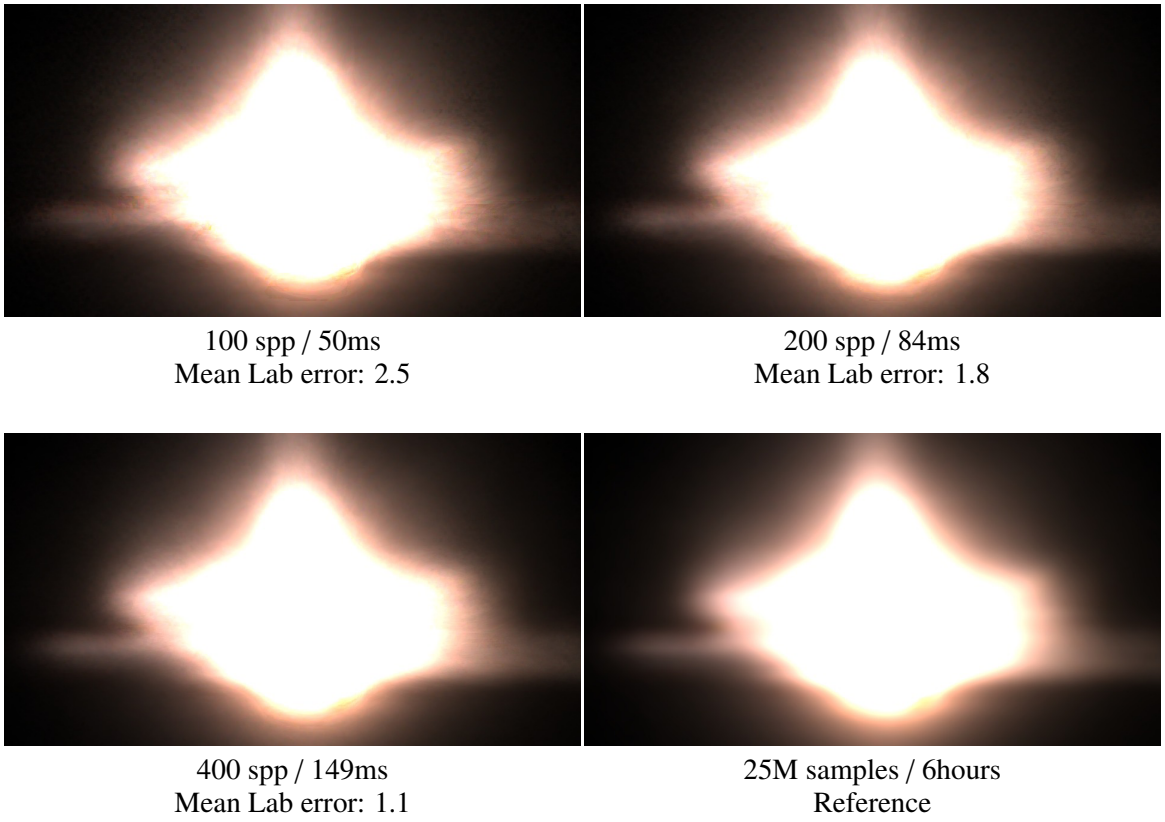


Figure 4.15: *Convergence of our sampling technique when increasing the number of samples per pixel (spp). The Lab errors are computed against the bottom right image. Our technique achieves a very good visual quality with only 200 samples per pixel.*

is illustrated in Figure 4.15 with the `car1` data. It is shown that our technique requires only a low number of samples per pixel to converge toward a high-quality result.

It is important to notice that the different frame rates reported in Figures 4.1, 4.12, and 4.16 for the car scene are due to the dynamic nature of our sampling strategy. For pixels belonging to a part of the scene that will not receive any energy, our dynamic sampling strategy prevents generating samples that would be useless. Therefore, the performance also depends on the percentage of lit pixels. Figure 4.12 details the cost of each rendering step of Section 4.5.1 to display one frame for the two-headligh `car2` scene. For 200 samples per pixel, the cost of our sampling techniques (i.e., about 39% with both step 1 and sampling) is almost the same as the shading time (i.e., 41% with shadow and shading). Although the computation of each $A_m(\boldsymbol{p})$ is duplicated in step 2, its cost remains negligible compared to the shading and sampling cost. This cost is only dependent on the size of the luminaire model, and its relative impact will decrease with the increasing number of samples. However, the shading and sampling costs are dependent on the number of samples and will thus stay similar. It is worth noticing that 16% of the time is spent accumulating the contribution of each light field image C_m of the luminaire model.

Figure 4.16: *Light editing for* `car2` *headlight data. Our technique allows to modify interactively (*8*fps in average) the position and orientation of captured light field.*

Finally, since we achieve interactive frame rate with real world light sources, our approach (cf. Figure 4.16) allows to rotate and translate light field data interactively. It would be also possible to change the parameter δ from Goesele et al. models. However, it is not a user parameter but an acquisition parameter: its modification results in a new luminaire that may not correspond to any feasable physical systems.

Compared to previous techniques, the main strength of our new approach is its ability to deal with 4D light field luminaires and to use acquired data directly. We do not convert the original data into an approximated alternative representation and ensure that our technique is unbiased per pixel.

Our approach is based on the fact that the two parameterizing planes are parallel. However, it can be extended to non-rectangular representation as long as we can compute a bounding box of the projected support. Unfortunately, without a two-plane parallel configuration, the projection of the axis-aligned box of the basis functions support would not be an axis-aligned bounding box. The sampling efficiency is directly related to the ratio of the support area to the bounding box area. Hence, for future acquisition setups, it might be worth to maintain it or to combine it with multiple two-plane configurations.

Although we demonstrate interactive frame rates, we think that there are potential improvements that could be investigated to reach real-time performances. First, as shown in the previous section, 16% of the rendering cost is due to the accumulation of the C_m images, hence reducing the number of images, or their size, would be a valid approach to improve the performances. Another possibility is to use the screen space coherency as recently demonstrated for visibility (e.g., [Clarberg & Akenine-Möller 2008]).

One can also expect that better balancing strategy could also lead to some improvements. We have experimented a solution that multiplies the accumulated energy $A_m(\boldsymbol{p})$ for one image I_m with the solid angle of the basis support viewed from the current position: this is an accurate estimation of the contribution of one image to the final shading. Unfortunately, it increases more the performance penalty than the quality in our test scenes. For instance, we have obtained 8% computation overhead with only 5% improvement in quality for the scene shown in Figure 4.15. However, such an approach may be worth being used for more complex shading scenarios.

Our GPU-dedicated and approximated visibility algorithm based on shadow maps is currently sufficient for interactive purposes and could be improved for better performance. With the current solution, we have introduced a uniform viewpoint selection for shadow map generation. Better strategies might certainly lead to speed and quality improvements. We have shown in Figure 4.9 that the best shadow quality than can be reached when evaluating the visibility for each light sample. Unfortunately, this is currently limited to offline processing.

Finally, our light source sampling approach could be extended by taking into account the influence of BRDF and visibility terms. There are two directions to achieve this goal. The first one would be to incorporate our approach in a Multiple Importance Sampling framework (e.g., [Pajot *et al.* 2011]). The second direction would be to determine how to incorporate the visibility or BRDF influence into the importance function (e.g., [Clarberg & Akenine-Möller 2008; Cline *et al.* 2006]), by adjusting the number of samples per image or by defining a more accurate restriction on the sampling domain.

4.7 Conclusion

In this chapter, we have introduced a position-dependent importance sampling technique that can be used on light field luminaires. This technique is based on a simple position-dependent affine transformation of CDFs that dynamically restricts the sampling domain. Furthermore, our technique distributes the number of samples between the different light field images according to their intensity. Combined together, these two solutions ensure that every generated sample contributes more evenly to the final results. Our new method is unbiased per pixel and requires only a small number of light samples to reach high-quality results. Finally, we have demonstrated a GPU implementation with a new and appropriate shadow algorithm that achieves interactive results.

However, this approach do not take into account BRDFs. We will investigate in the next chapter how to combine light-based and BRDF-based sampling techniques.

Chapter 5

Automatic Variance Reduction for MIS

A good sampling technique leads to fast convergence rate of Monte Carlo estimation (cf. Section 2.3.5). In Chapter 3 and Chapter 4, we have introduced two light-based importance sampling techniques that can generate good samples efficiently. Besides, from the overview in Section 2.3.6, to our method of combining BRDF-based sampling in Chapter 3 and the discussion of combining BRDF-based sampling in Chapter 4, it is clear that combining different sampling strategies is an interesting approach for generating good samples.

Since different sampling strategies have different advantages and disadvantages, it is very important to match these sampling properties to the characteristics of the scene. For example, if the light source contains high frequencies, using light-based importance sampling is more effective, whereas if the material is highly specular, using BRDF-based importance sampling is a better choice. However, in general, most scenes are composed with light sources and materials with arbitrary frequency and glossiness. Therefore, choosing a dedicated strategy is a problem.

Multiple Importance Sampling (MIS) (cf. Section 2.3.6) reduces the impact of choosing a dedicated strategy by balancing the number of samples between them (e.g., light-based or BRDF-based sampling). However, an automatic choice of the optimal balancing remains a difficult problem. Without any scene characteristics knowledge, the default choice is to select the same number of samples from different strategies and to use them with heuristic techniques (e.g., balance, power or maximum).

In this chapter, we introduce a second-order approximation for variance reduction in the context of balance heuristic. Based on this approximation, we introduce an automatic distribution of samples for direct lighting without any prior knowledge of the scene characteristics. We demonstrate that for all test scenes (with different types of materials, light sources and visibility complexity), the assumption of variance reduction is fulfilled in average. We also propose an implementation of this framework for offline applications with a low overhead and for online applications that allows a direct visualization of sample distribution. We hope that these tools will help developing new balancing strategies. This work

has been presented in Pacific Graphics 2013 as a full paper [Lu *et al.* 2013a] and will be in the Computer Graphics Forum.

5.1 Motivation

In order to remind the reader about how to compute radiance, we reintroduce the rendering equation. Note that, in Chapter 3 and Chapter 4 the visibility term is omitted, and in Chapter 3, the lighting is distant lighting which has only directional variations.

In this chapter, we focus on a general case of the equation and take into account the visibility term as well. Once again, we focus on estimating the reflected radiance [Kajiya 1986] expressed as:

$$L(\boldsymbol{p} \rightarrow \boldsymbol{o}) = \int_{\Omega} \mathrm{s}(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega})\, \mathrm{d}\boldsymbol{\omega} \quad \text{with}$$
$$\mathrm{s}(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega}) = \rho(\boldsymbol{o}, \boldsymbol{\omega})\ \langle \boldsymbol{p}, \boldsymbol{\omega} \rangle\ V(\boldsymbol{p}, \boldsymbol{\omega})\, L(\boldsymbol{\omega} \rightarrow \boldsymbol{p})$$

where Ω is the full unit sphere, $\rho(\boldsymbol{o}, \boldsymbol{\omega})$ is the reflectance function (BRDF), $<,>$ the positive clamped dot product operator (aka *cosine factor*), $V(\boldsymbol{p}, \boldsymbol{\omega})$ is the visibility function and, $L(\boldsymbol{\omega} \rightarrow \boldsymbol{p})$ represents the radiance in the direction $\boldsymbol{\omega}$ from the environment map.

The main trend to compute such integral is to use Monte Carlo methods where the estimation of a reflected radiance $L(\boldsymbol{p} \rightarrow \boldsymbol{o})$ is computed from a set of N samples:

$$L(\boldsymbol{p} \rightarrow \boldsymbol{o}) \approx \frac{1}{N} \sum_{n=1}^{N} \frac{\mathrm{s}(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega}_n)}{\mathrm{pdf}(\boldsymbol{\omega}_n)}\,. \tag{5.1}$$

In this equation, $\mathrm{pdf}(\boldsymbol{\omega}_n)$ is a Probability Density Function (PDF), the closer it matches the numerator of the Equation the lower the number of samples N will be required to converge to the solution. Although the best matching is reached when pdf is proportional to $\mathrm{s}(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega})$, it is hard to obtain an exact integrable analytical form in most cases.

5.2 In-depth Related Work on MIS

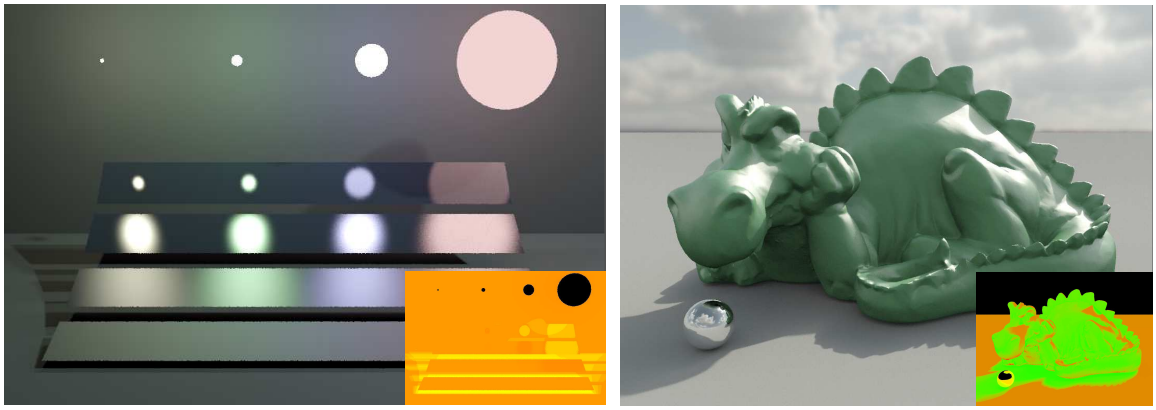
A lot of work has been dedicated to derive an efficient sampling strategy for each component, or a subset, of the function $\mathrm{s}(\boldsymbol{p}, \boldsymbol{o}, \boldsymbol{\omega})$: light sampling [Agarwal *et al.* 2003; Ergun *et al.* 2012; Havran *et al.* 2005; Ostromoukhov *et al.* 2004], or BRDF (e.g., [Ashikhmin & Premoze 2007]) or BRDF corrected by the cosine factor [Lawrence *et al.* 2004; Montes *et al.* 2008; Pacanowski *et al.* 2012]. For visibility, since no analytical form exists, precomputation [Ghosh & Heidrich 2006] is generally required. In general, a product cannot be

accurately approximated with only one of its components.

A better PDF can be obtained by an approximation of the whole integrand. Bidirectional Importance Sampling techniques [Burke *et al.* 2005] approximate this product on a preliminary sampling (either BRDF-based or light-based): a weight is computed for each sample according to the shading functions. Then, a subset of these samples is selected by using these weights. However, these techniques are very sensitive to the initial choice of samples that may introduce some bias. Another potential problem is due to the over-sampling needed to perform the second-step (resampling): some generated samples will not be used for the final estimator. Wang et al. [Wang & Akerlund 2009] improved the techniques by combining lightcuts [Walter *et al.* 2005b] with a BRDF-based sampling to reduce the bias caused by the initial distribution.

More generally, precomputing conservative approximations of lighting environment and BRDF helps computing dynamically PDFs based on the product of the lighting and BRDF. For this purpose, Cline et al. [Cline *et al.* 2006] split the environment map according to BRDF peaks computed from BRDF-based importance sampling. To be more conservative, Clarberg et al. [Clarberg & Akenine-Möller 2008] use a wavelet representation of the environment map, a wavelet approximation of the BRDF, and a fast computation of their product. These techniques require dedicated representations to support a per-pixel dynamic computation of the importance sampling structure. Although photon maps may also be used (e.g., [Pajot *et al.* 2011]), in this chapter we focus on a technique that does not required any additional storage.

Compared to these approaches, Multiple Importance Sampling (MIS) [Veach & Guibas 1995] is still a simple and efficient way to combine different sampling techniques by first, balancing the number of samples between the different strategies and second, combining them into a weighted estimator (balance, power, maximum heuristics [Veach & Guibas 1995] or α-max heuristic [Georgiev *et al.* 2012]). Despite the average improvement over the use of a unique strategy, the optimal balancing may differ from pixel to pixel in an unpredictable way [Owen & Zhou 2000]. To reduce this problem, Pajot et al. [Pajot *et al.* 2011] have introduced the empirical notion of *representativity* that estimates the concentration information for BRDF or photon maps. Intuitively, highly concentrated phenomena will result in higher number of samples. General guidelines to develop new representativity functions are introduced, but problems may occur when multiple concentrations exist such as in environment maps.



Scene 1 Scene 2

Figure 5.1: *Our method on two different scenes: (**Left**) Low to high glossy materials with five diffuse area light sources. (**Right**) Glossy materials with high-frequency environment map lighting.*

5.3 Second-Order Approximation for Variance Reduction

As shown in Figure 5.1, our per-pixel second-order approximation of the variance leads to a new and automatic approach for balancing the number of samples between two different sampling strategies. Except for light sources, the inset images show the sample distribution for each pixel: red (resp. green) means that samples were generated using light (resp. BRDF) importance sampling and perfect yellow represents the default balance heuristic strategy [Veach & Guibas 1995]. The validity of our approximation is demonstrated by the fact that compared to the balance heuristic, the variance is reduced by (**Left**) 26% and (**Right**) 20% in average (14% and 11% for the standard deviation).

Contributions In this chapter, we introduce a second-order approximation for variance reduction when using MIS approach. We demonstrate that our approach leads to a new and automatic way to improve in average samples distribution between different strategies. We show that for our different test scenes and for direct lighting, our approximation also improves the average variance, validating the accuracy of our approximation. We also demonstrate that we can use our approach for offline rendering for which its efficiency surpasses previous balancing techniques. Its accuracy also permits to visualize what should be the dominant strategy for future investigations.

5.3.1 Theoretical Variance Reduction

To estimate the integral of a function $L = \int \mathrm{f}(\omega)\,\mathrm{d}\omega$ using random samples $\{\omega_{i=\mathrm{B|L},n=1..N_i}\}$ from two PDFs pdf_B (for BRDF-based strategy to generate N_B samples) and pdf_L (for light-based strategy to generate N_L samples), Veach & Guibas [1995] have introduced the fol-

lowing MIS primary estimator

$$L_{N_\mathrm{L},N_\mathrm{B}} = \sum_{i=\mathrm{B}|\mathrm{L}} \frac{1}{N_i} \sum_{n=1}^{N_i} w_i\left(\boldsymbol{\omega}_{i,n}\right) \frac{\mathrm{f}\left(\boldsymbol{\omega}_{i,n}\right)}{\mathrm{pdf}_i\left(\boldsymbol{\omega}_{i,n}\right)} \quad , \tag{5.2}$$

where $w_i(\boldsymbol{\omega}_{i,n})$ is a weighting function. For balance heuristic,

$$w_i\left(\boldsymbol{\omega}_{i,n}\right) = \frac{N_i \,\mathrm{pdf}_i\left(\boldsymbol{\omega}_{i,n}\right)}{N_\mathrm{B}\mathrm{pdf}_\mathrm{B}\left(\boldsymbol{\omega}_{i,n}\right) + N_\mathrm{L}\mathrm{pdf}_\mathrm{L}\left(\boldsymbol{\omega}_{i,n}\right)} \quad .$$

By denoting $N = N_\mathrm{B} + N_\mathrm{L}$ (N is the total number of samples) and $N_\mathrm{B} = \alpha N$ with $\alpha \in [0, 1]$, and as pointed by Georgiev et al. [Georgiev *et al.* 2012], this estimator is equivalent to the secondary estimator

$$L_{N,\alpha} = \frac{1}{N} \sum_{n=1}^{N} \frac{\mathrm{f}\left(\boldsymbol{\omega}_n\right)}{\alpha\,\mathrm{pdf}_\mathrm{B}\left(\boldsymbol{\omega}_n\right) + (1-\alpha)\,\mathrm{pdf}_\mathrm{L}\left(\boldsymbol{\omega}_n\right)} \quad . \tag{5.3}$$

This is the estimator for a Defensive Importance Sampling strategy [Hesterberg 1995] where a set of N samples $\boldsymbol{\omega}_n$ is generated according to $\alpha\,\mathrm{pdf}_\mathrm{B} + (1-\alpha)\,\mathrm{pdf}_\mathrm{L}$. In this chapter, we focus on minimizing the variance of this estimator to improve the sampling quality.

5.3.2 Second-Order Approximation of Variance

Finding the best α that minimizes the variance of $L_{N,\alpha}$ (i.e., $\mathrm{V}\left[L_{N,\alpha}\right]$) is equivalent to minimize the sampling variance of the strategy (i.e., $\mathrm{V}\left[L_{1,\alpha}\right]$) since

$$\mathrm{V}\left[L_{N,\alpha}\right] = \frac{1}{N}\mathrm{V}\left[L_{1,\alpha}\right] \quad .$$

If $\alpha\,\mathrm{pdf}_\mathrm{B} + (1-\alpha)\,\mathrm{pdf}_\mathrm{L}$ leads to a unbiased strategy for all α (this is the case when both pdf_B and pdf_L generate unbiased samples), the expected value of $L_{1,\alpha}$ is independent of α since, by definition of unbiased techniques, $\mathrm{E}\left[L_{1,\alpha}\right] = L$. Minimizing the sampling variance is thus similar to minimize $\mathrm{E}\left[L_{1,\alpha}^2\right]$ since $\mathrm{V}\left[L_{1,\alpha}\right] = \mathrm{E}\left[L_{1,\alpha}^2\right] - \mathrm{E}\left[L_{1,\alpha}\right]^2 = \mathrm{E}\left[L_{1,\alpha}^2\right] - L^2$.

Nonetheless, the analytical form of the minimum for such a function is too complex to obtain. Therefore, we approximate the problem using a Taylor expansion around $\alpha = 1/2$:

$$\mathrm{E}\left[L_{1,\alpha}^2\right] \simeq \sum_{n=0}^{N} (-1)^n \left(2\,\alpha - 1\right)^n \int \frac{\mathrm{f}^2 \Delta\mathrm{pdf}^n}{\bar{\mathrm{pdf}}^{n+1}} \quad ,$$

where $\bar{\mathrm{pdf}} = (\mathrm{pdf}_\mathrm{B} + \mathrm{pdf}_\mathrm{L})/2$, $\Delta\mathrm{pdf} = (\mathrm{pdf}_\mathrm{B} - \mathrm{pdf}_\mathrm{L})/2$, and N is the order of Taylor expansion.

To minimize $\mathrm{E}\left[L_{1,\alpha}^2\right]$, we have to compute α such as

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\mathrm{E}\left[L_{1,\alpha}^2\right] = 0 \quad .$$

Using the second-order approximation

$$\mathrm{E}\left[L_{1,\alpha}^2\right] \simeq \sum_{n=0}^{2}(-1)^n\,(2\,\alpha-1)^n \int \frac{\mathrm{f}^2\Delta\mathrm{pdf}^n}{\bar{\mathrm{pdf}}^{n+1}} \quad ,$$

leads to the linear form

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\mathrm{E}\left[L_{1,\alpha}^2\right] \simeq \sum_{n=1}^{2}(-1)^n\,2\,n\;(2\,\alpha-1)^{n-1} \int \frac{\mathrm{f}^2\Delta\mathrm{pdf}^n}{\bar{\mathrm{pdf}}^{n+1}}\,,$$

or more simply

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\mathrm{E}\left[L_{1,\alpha}^2\right] \simeq -2\int \frac{\mathrm{f}^2\Delta\mathrm{pdf}}{\bar{\mathrm{pdf}}^2} + 4\;(2\,\alpha-1)\int \frac{\mathrm{f}^2\Delta\mathrm{pdf}^2}{\bar{\mathrm{pdf}}^3} \quad .$$

This approximation is equal to zero when

$$2\alpha - 1 = \frac{1}{2}\left(\int \frac{\mathrm{f}^2\Delta\mathrm{pdf}}{\bar{\mathrm{pdf}}^2} \Big/ \int \frac{\mathrm{f}^2\Delta\mathrm{pdf}^2}{\bar{\mathrm{pdf}}^3}\right) \quad .$$

Therefore, this quadratic objective function is minimized when

$$\alpha = \frac{1}{4}\left(2 + \int \frac{\mathrm{f}^2\Delta\mathrm{pdf}}{\bar{\mathrm{pdf}}^2} \Big/ \int \frac{\mathrm{f}^2\Delta\mathrm{pdf}^2}{\bar{\mathrm{pdf}}^3}\right) \quad . \tag{5.4}$$

The choice of Taylor expansion around $\alpha = 1/2$ is motivated by the following reasons. First, it ensures that the best approximation is centered in the definition domain of α, which is $[0, 1]$. Similarly, an approximation around $\alpha = 0$ or $\alpha = 1$ will give a strong priority to one particular sampling technique. In our case, the priority is given to the default balance heuristic. As discussed in Section 5.3.4, when the best sampling strategy is to use a unique strategy (i.e., $\alpha = 0$ or $\alpha = 1$), our approximation becomes less valid. However, as demonstrated in the Section 5.4, our estimation of α still reduces the variance compared to the default balance heuristic.

5.3.3 Estimation of α

Despite the fact that we have an analytical formula (cf. Equation 5.4) to compute what would the best α according to a second-order approximation of the sampling variance, two

numerical integrations have to be computed. For this purpose, we use a MIS approach by classically selecting the same number of samples from each strategy, and combining them with balance heuristic. Then, the two integrals are estimated using M samples:

$$\begin{aligned}\int \frac{\mathrm{f}^2\Delta\mathrm{pdf}}{\bar{\mathrm{pdf}}^2} &\simeq \frac{1}{M}\sum_{m=1}^{M}\left(\frac{\mathrm{f}(\omega_m)}{\bar{\mathrm{pdf}}(\omega_m)}\right)^2 \Delta\mathrm{pdf}(\omega_m)\frac{1}{\bar{\mathrm{pdf}}(\omega_m)} \\ \int \frac{\mathrm{f}^2\Delta\mathrm{pdf}^2}{\bar{\mathrm{pdf}}^3} &\simeq \frac{1}{M}\sum_{m=1}^{M}\left(\frac{\mathrm{f}(\omega_m)}{\bar{\mathrm{pdf}}(\omega_m)}\right)^2 \frac{\Delta\mathrm{pdf}^2(\omega_m)}{\bar{\mathrm{pdf}}(\omega_m)}\frac{1}{\bar{\mathrm{pdf}}(\omega_m)} \quad .\end{aligned} \tag{5.5}$$

5.3.4 Theoritical Accuracy for Two Extreme Cases

Since it is based on an approximation, our approach has a limited accuracy. We estimate it for two extreme cases: perfect mirror BRDF where α has to be 1 and Dirac light source where α has to be 0.

For highly specular cases, BRDFs are zero in almost every direction excepted in a small direction domain. And the shading function follows the same behavior. In this domain, $\mathrm{pdf}_\mathrm{B}(\omega)$ is very large, therefore, $\Delta\mathrm{pdf}$ and $\bar{\mathrm{pdf}}$ are well approximated by $\mathrm{pdf}_\mathrm{B}(\omega)$. For simplicity we denote $\mathrm{pdf}_\mathrm{B}(\omega)$ as pdf_B. Consequently we get,

$$2\alpha - 1 = \frac{1}{2}\left(\int \frac{\mathrm{f}^2\mathrm{pdf}_\mathrm{B}}{\mathrm{pdf}_\mathrm{B}^2} \Big/ \int \frac{\mathrm{f}^2\mathrm{pdf}_\mathrm{B}^2}{\mathrm{pdf}_\mathrm{B}^3}\right) \quad ,$$

or

$$2\alpha - 1 = \frac{1}{2}\left(\int \frac{\mathrm{f}^2}{\mathrm{pdf}_\mathrm{B}} \Big/ \int \frac{\mathrm{f}^2}{\mathrm{pdf}_\mathrm{B}}\right) \quad .$$

Finally

$$2\alpha - 1 = \frac{1}{2} \Leftrightarrow \alpha = \frac{3}{4} \quad .$$

we found that $\alpha = 3/4$ instead of $\alpha = 1$ for mirror BRDF. Reciprocally, the same demonstration for very concentrated light source leads to

$$2\alpha - 1 = -\frac{1}{2} \quad .$$

It shows that with extremely directional light sources, α becomes equal to $1/4$ for our approximation whereas $\alpha = 0$ is expected. Therefore, we know that for these extreme cases we cannot find the optimal value for α. However, classically for MIS, it is usual to not use either $\alpha = 0$ or $\alpha = 1$ to ensure that the resulting estimator remains unbiased even if one of the two techniques is biased. Furthermore, the value of α still clearly indicates the dominant strategy.

Method \ Scene	1	2	3	4	5	6	7	8	9	10	11
BRDF	74.8	966	251	10.9	1.14	10.5	0.94	48.1	**0.08**	347	6.87
Light	0.52	**0.54**	**0.05**	**0.52**	10.1	0.73	10.7	0.84	472	1.93	2.31
Balanced heuristic	1	1	1	1	1	1	1	1	1	1	1
Power heuristic	1.10	1.02	1.01	1.11	1.10	1.15	1.11	1.07	1.02	1.09	1.07
Maximum heuristic	1.38	1.07	1.01	1.30	1.34	1.31	1.70	1.42	1.02	1.54	1.36
Preprocessed α	**0.47**	0.62	0.36	0.59	**0.66**	**0.39**	**0.88**	**0.77**	0.38	**0.75**	**0.82**

Table 5.1: *Relative comparisons with default balance heuristic, for each scene, of different balancing methods in terms of average sampling variance. The average variances are computed over all the pixels using the same number of samples $N = 256$ for each light source. The absolute values are available in appendix (Table 1).*

5.4 Numerical Validation

We have implemented our α estimation into PBRT [Pharr & Humphreys 2004] and tested our technique on many different scenes (cf. Figures 5.1 and 5.2). All the results are computed using 4 camera rays per pixel. For each camera ray, we use the default PBRT approach and sample uniformly all light sources and accumulate the contribution of each light source to estimate the final radiance. In the following, we will indicate the number of samples per light source.

5.4.1 Variance Reduction

To demonstrate the accuracy limits of our approach, we compare (cf. Table 5.1) the use of a reliable estimation of α, which is computed using $M = 4096$ samples for each light source, against different sampling and combination strategies: only light-based importance sampling, only BRDF-based one, and the half-half strategies that use the same number of samples between the two sampling techniques, combined with balance, power, or maximum heuristics. We do not compare with the α-max heuristic [Georgiev *et al.* 2012] since it requires additional knowledge about which strategy achieves more often a better sampling than the others.

As shown in Table 5.1, in most cases our technique reduces the variance compared to previous heuristics. As discussed in Section 5.3.4, since our second-order approximation is done around $\alpha = 1/2$, the approximation is getting worse for $\alpha = 0$ or $\alpha = 1$. Therefore, we cannot guarantee to correctly detect if using only light-based or BRDF-based strategy is the best solution. This is demonstrated in scenes 2, 3, and 4 where only light-based sampling will perform better in average, and in scene 9 where only BRDF-based sampling will be better. However, our approach is always better than balance/power/maximum heuristics: this demonstrates that our second-order approximation is sufficient to improve over

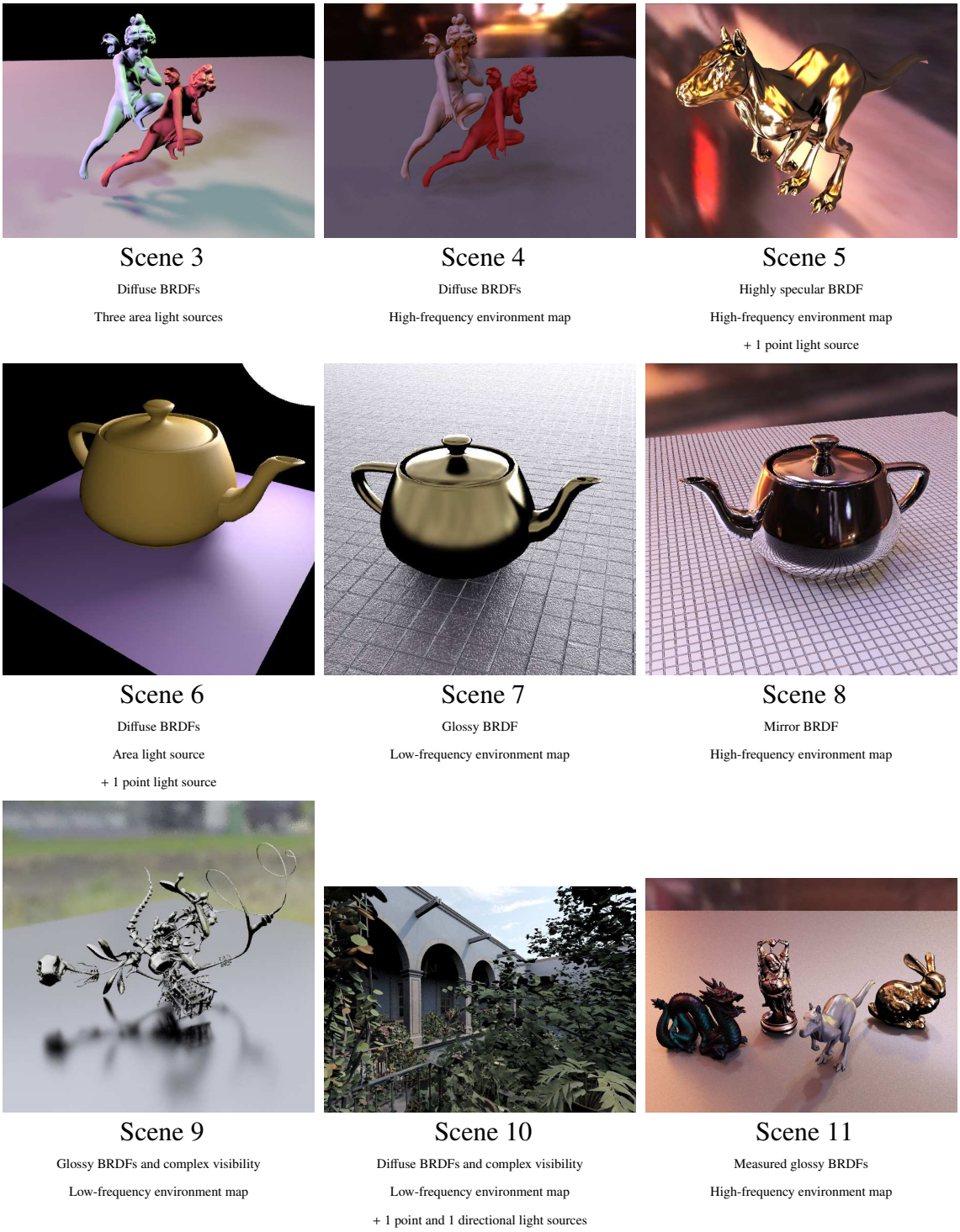


Figure 5.2: *Different test scenes, ranging from diffuse (scenes 3, 4 and 6) to specular (scenes 8 and 2) materials including complex visibility (scene 10) with different types of illumination such as area light sources (scenes 1, 3 and 6) and environment maps. Since only one sample is required for point and directional light sources, they are added in a post-process and removed from the variance computation.*

these strategies. Moreover, selecting either light-based or BRDF-based requires an a priori knowledge, which is unavailable in most practical cases, but our α estimation provide an

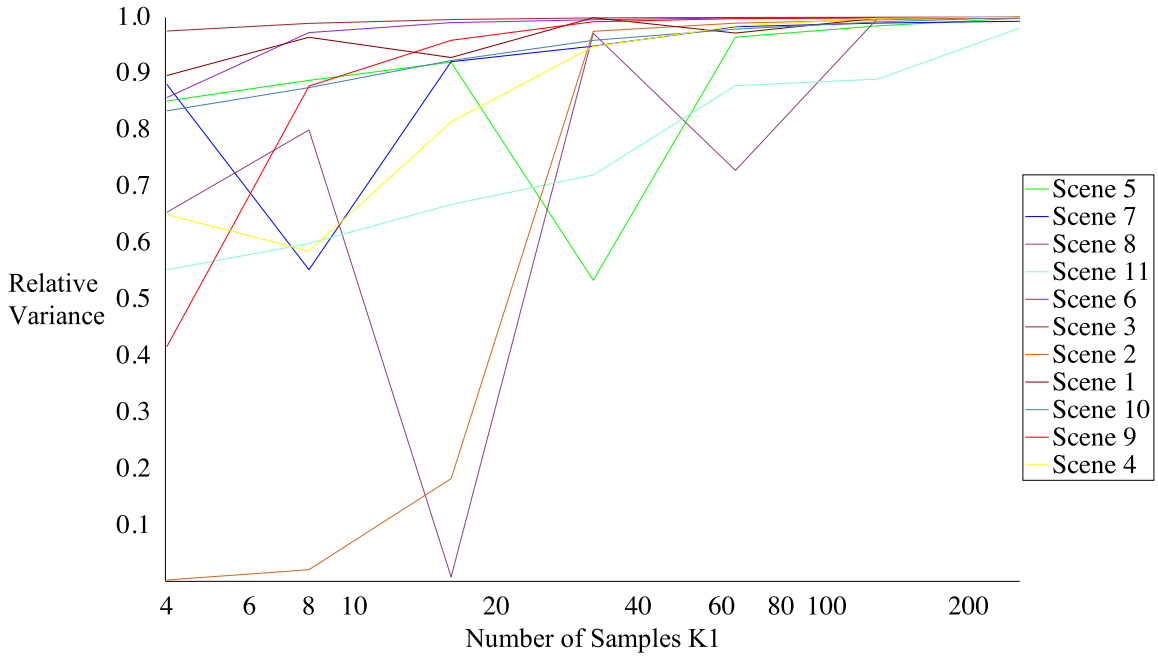


Figure 5.3: *Number of samples M to estimate α per scene. The y-axis represents the variance for $L_{N,\alpha}$ ($N = 256$), relatively to the best variance obtained by our technique with different value of M for a given scene, thus illustrates the convergence of α estimation.*

indication of what may be the dominant strategy.

5.4.2 Convergence Speed for α

For practical reasons, the number of samples to estimate α should be as low as possible. In Figure 5.3 we investigate how many samples are required to bring a satisfying variance reduction. Depending on the scene, the number of samples per light source can be low (upper curve) but generally speaking, at least 128 samples per light source are required to compute a good estimation of α that will improve the variance of the $L_{N,\alpha}$ estimator. We will use this number for the following tests.

5.4.3 Clamping Interval for α

We already know from Section 5.3.4 that we have, in some extreme cases, a theoretical and limited accuracy when estimating α. To investigate accuracy in a more general context, we have tested (cf. Figure 5.4) the effect of clamping the estimated α. This figure shows that for some scenes the variance increases if α may get a value over the interval $[0.025, 0.975]$ (i.e., with a size of 0.95). This experience confirms two facts. First, the second-order approximation is not sufficient to capture some variance oscillations in the interval $[0, 1]$ since clamping may lead to better results. However, it also confirms that our estimation still provide a good hint of what is the dominant strategy. It also justifies that we clamp α to

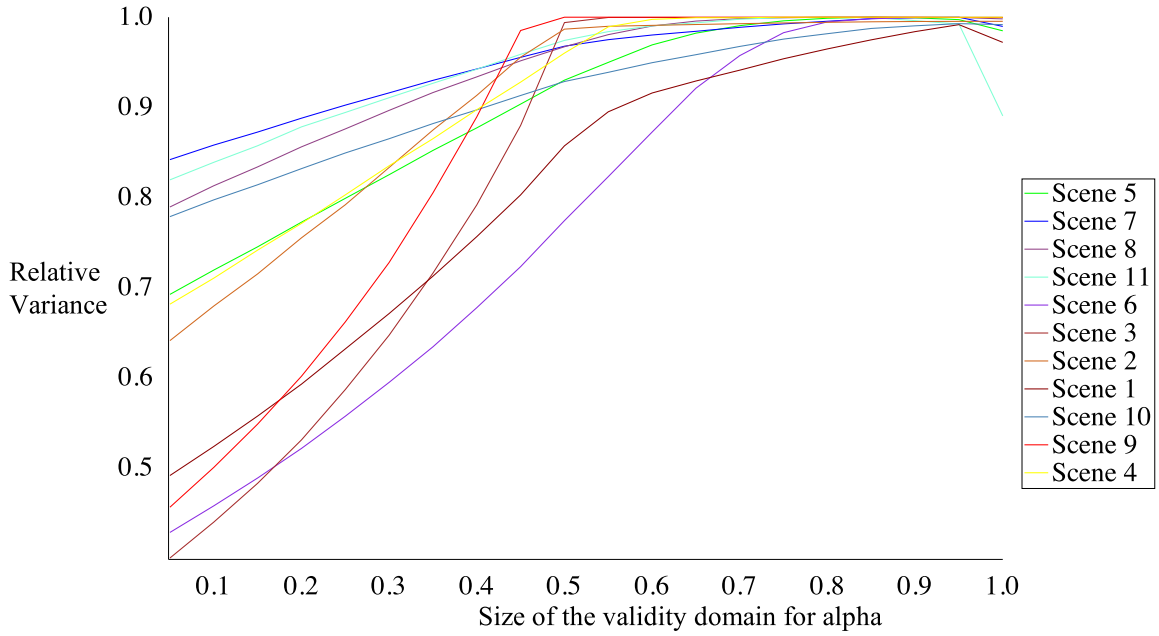


Figure 5.4: *Effect of clamping α for each scene. The y-axis represents the variance for $L_{N,\alpha}$, relatively to the best variance (cf. Figure 5.3). The x-axis represents the size of an interval centered at $1/2$ that restricts the value of α. When the size if the validity domain is equal to zero, this corresponds to the default balance heuristic. For each scene, the number of samples per light source to estimate α is set to $M = 128$ and the one to compute the variance to $N = 256$.*

[0.025, 0.975] for all the following results.

5.5 Application and Results

After having demonstrated the validity of our approximation, we illustrate its potential applications.

5.5.1 Online α Estimation

A first application is to integrate our estimation directly into an offline rendering system with MIS, to automatically detect the balancing strategy without extra storage and without prior knowledge of a scene. For this purpose, we use a two-step approach. We first estimate α using a low number of M samples using half-half balancing with default balance heuristic as in Equation 5.5. At the same time, we estimate a first radiance value $L_{M/2,M/2}$ as in Equation 5.2. Once α is known, we estimate the second radiance value $L_{\alpha N,(1-\alpha)N}$. In order to not waste any samples, we combine these two estimators using

$$L_{N+M} = \frac{M}{M+N} L_{M/2,M/2} + \frac{N}{M+N} L_{\alpha N,(1-\alpha)N} \,. \tag{5.6}$$

With this formula, when $\alpha = 0.5$, our estimator is exactly the same as the default balance heuristic one. Another solution to combine the two steps would be to store all the samples of the first step and to integrate them in the second one using the balancing weights of MIS. However, this would require an extra-storage of size M.

Variance Reduction Compared to previous techniques (cf. Table 5.2-top), our approach still preserves the variance reduction shown in Section 5.4.1. For a more perceptual comparison, we have computed Lab errors against reference solutions rendered with 16000 samples per camera ray and per light source using default balance heuristic technique (cf. Table 5.2-middle). In many cases, our approach achieves the lowest Lab error. It is interesting to note that the power heuristic reaches the lowest Lab error on a scene very similar to the one introduced by Veach.

Efficiency and Computation Time We have also investigated the overhead of the required α estimation on an Intel i7-3930K@3.2 GHz PC with 16GB of memory (cf. Table 5.2-center). This overhead takes place only during the first step and is consistently about 1% of the rendering time when using the default balance heuristic. This comes from the fact that most of the cost is due to shading computation PDF evaluation: the estimation of α, based on Equation 5.5, simply reuses these values.

	Method \ Scene	1	2	3	4	5	6	7	8	9	10	11
Variance	BRDF	74.8	966	251	10.9	1.14	10.5	0.94	48.1	**0.08**	347	6.87
	Light	0.52	**0.54**	**0.05**	**0.52**	10.1	0.73	10.7	0.84	472	1.93	2.31
	Balanced heuristic	1	1	1	1	1	1	1	1	1	1	1
	Power heuristic	1.10	1.02	1.01	1.11	1.10	1.15	1.11	1.07	1.02	1.09	1.07
	Maximum heuristic	1.38	1.07	1.01	1.30	1.34	1.31	1.70	1.42	1.02	1.54	1.36
	Preprocessed α	**0.47**	0.62	0.36	0.59	**0.66**	**0.39**	**0.88**	**0.77**	0.38	**0.75**	**0.82**
	Online estimation of α	0.73	0.82	0.68	0.81	**0.85**	**0.72**	0.98	0.86	0.68	**0.89**	**0.93**
Lab error	BRDF	14.9	6.5	10.5	2.80	1.10	22.9	2.00	3.85	**0.36**	5.26	3.50
	Light	0.14	1.11	**0.25**	**0.78**	7.50	0.23	3.60	1.16	15.9	2.00	2.28
	Balance heuristic	0.14	0.81	0.31	0.91	0.77	0.28	1.04	1.21	0.47	1.65	1.60
	Power heuristic	**0.11**	0.83	0.31	0.93	0.78	0.28	1.06	1.22	0.46	1.70	1.59
	Maximum heuristic	0.12	0.95	0.31	0.99	0.85	0.28	1.30	1.37	0.45	1.98	1.71
	Combined estimator	0.13	**0.78**	0.29	0.86	**0.74**	**0.26**	**1.02**	**1.15**	0.44	**1.60**	**1.56**
Time (sec.)	BRDF	**159**	138	**40.6**	55.1	33.7	**16.9**	59.1	36.5	49.2	354	91.0
	Light	300	**103**	147	**41.8**	**18.6**	33.8	**41.7**	**34.2**	**27.1**	**178**	**53.0**
	Balance heuristic	238	124	95.5	48.9	28.0	25.5	52.0	36.2	40.0	281	74.0
	Power heuristic	240	122	95.7	48.8	27.1	24.9	49.9	35.9	39.2	275	74.0
	Maximum heuristic	239	122	95.4	48.1	26.0	24.5	49.5	34.9	38.5	264	71.1
	Combined estimator	251	128	107	45.8	28.3	26.9	51.1	34.5	40.9	281	72.0
Efficiency	BRDF	0.02	0.00	0.01	0.08	0.73	0.14	0.94	0.02	**10.3**	0.00	0.12
	Light	**1.51**	**2.24**	**12.4**	**2.24**	0.15	1.03	0.12	**1.26**	0.00	0.81	0.60
	Balance heuristic	1	1	1	1	1	1	1	1	1	1	1
	Power heuristic	0.89	1	0.99	0.90	0.94	0.89	0.93	0.94	1	0.93	0.93
	Maximum heuristic	0.81	0.94	1.00	0.87	0.85	0.89	0.68	0.78	1.02	0.73	0.82
	Combined estimator	1.43	1.18	1.34	1.45	**1.23**	**1.49**	**1.10**	1.29	1.44	**1.19**	**1.19**

Table 5.2: *Relative average variance and Lab error, rendering time and relative efficiency for each scene and for different balancing methods. We use $N = 256$ samples per light source for each method. Combined estimator refers to the one introduced in Equation 5.6 where we use half of the total samples to evaluate α ($M = 128$ in Equation 5.6) and the remaining half to evaluate the radiance with the estimated α ($N = 128$ in Equation 5.6). For comparison purpose, we copy the variance of other strategies and our preprocessed one from Table 5.1. The relative variance is computed as the variance of each strategy against the variance of the default balance heuristic. The Lab error is computed against the reference solution. The efficiency [Veach 1998] of each strategy is computed as the inverse of the product of the variance and the rendering time. The relative efficiency is computed as the efficiency of each strategy against the efficiency of the default balance heuristic. Absolute values are in appendix (Table 1).*

The bottom part of Table 5.2 shows the efficiency [Veach 1998] of each sampling strategy. It is computed as the inverse of the product of the variance and the rendering time. For an efficiency point of view, and for all tested scenes, our strategy outperforms the previous ones (balance, power and max heuristics). Moreover, for some scenes (e.g., 5-7), our

method is even better than the two extreme strategies: BRDF-based or light-based sampling. Nevertheless and as already mentioned, selecting without prior knowledge of the scene one of the two extreme strategies remains difficult.

5.5.2 Visual Feedbacks

Since we have demonstrated that, in average, our approach outperforms balance heuristic and gives a good hint of which sampling technique is better suited for variance reduction, another potential application is the direct visualization of the balancing strategy. Direct visualization of what would be a good balancing strategy might help in a better understanding of the behavior and thus future development of direct and efficient distribution of samples.

Furthermore, the low overhead of our approach is suitable for a GPU implementation. It could help exploring new balancing strategies for dynamic scenes. For this purpose, we have implemented our two-step approach, described in the previous section, into a full dedicated GPU solution for dynamic environment maps [Lu *et al.* 2013b] lighting. The solution is implemented on a NVIDIA 580 GTX with 1.5 GB of memory. For one environment map and without any visibility computation, we achieve a frame rate of 67 fps with 256 samples per pixels at a 1024×768 resolution.

Visualization of Variance Reduction We have demonstrated that we outperform the balance/power/maximum heuristics in most cases. Visualizing the per-pixel improvement on variance reduction might help in understanding more precisely how samples have to be distributed. To visualize the variance reduction for each pixel, we compute the difference of variance between traditional balance heuristic and our approach (cf. Figure 5.5). As shown by the preponderance of the red color in Figure 5.5, our method outperforms the default balance heuristic for most pixels. When our technique fails to be better (green pixels), the value of α oscillates around $\alpha = 1/2$ (cf. the inset image in Figure 5.1-left): this illustrates the slow convergence for α-estimation in some cases as discussed in the next Section.

Dynamic Visualization of α The GPU implementation is a convenient tools to visualize in real-time the effects of different parameters on the sample distribution: we can dynamically change the view point, the BRDF parameters, and the number of samples M used to estimate α. We can even observe the evolution of α for a dynamic light source. In the left image of Figure 5.6, the sun is behind the dragon, which leads to a large priority to BRDF-based sampling, whereas on the right image the sun is facing the dragon and the sampling priority has shifted to the light source. Finally, this also illustrates that our approach is perfectly suited for dynamic scenes where it is difficult to have a priori knowledge of the scene characteristics.

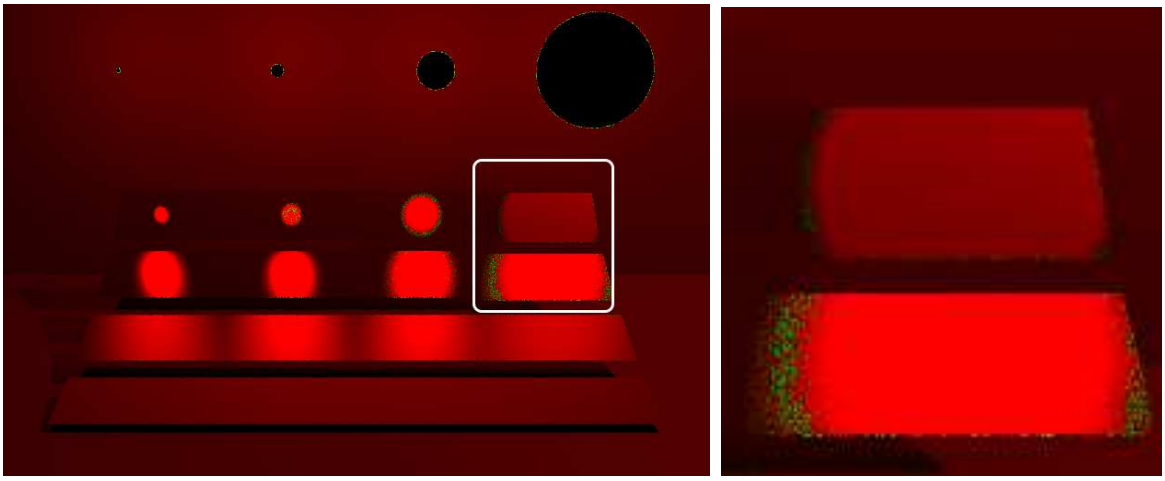
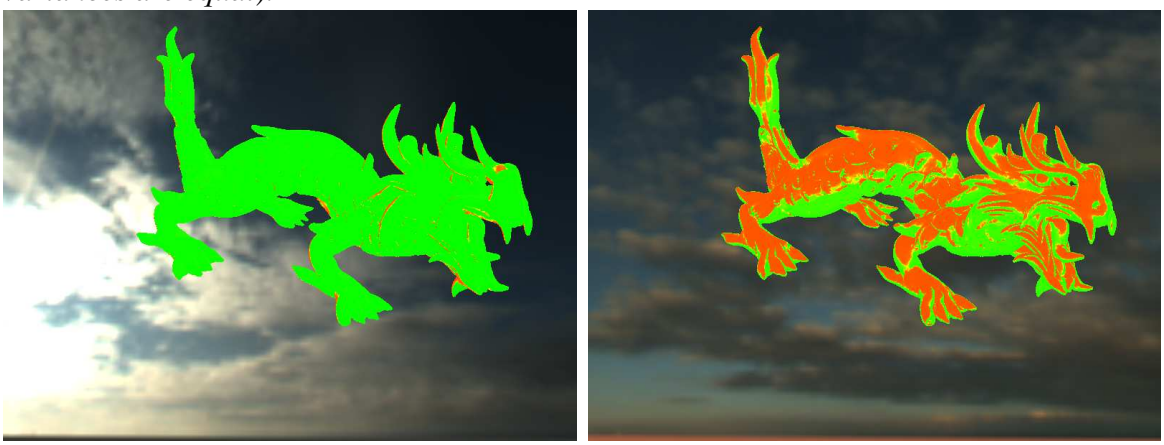


Figure 5.5: *Visualization of per-pixel variance. These images are computed as the difference of variance between the default balance heuristic and our method.* ***Red*** *corresponds to pixels where our method is better whereas* ***Green*** *corresponds to those where the balance heuristic is better.* ***Black*** *means that none of the methods outperforms the other (i.e., variances are equal).*



Morning sky Afternoon sky

Figure 5.6: *Dynamic visualization of α. The red channel represents α (samples selected from light sources), whereas the green channel represents $1-\alpha$ (samples selected from BRDF). These two frames are rendered around 142 fps (without visibility) with* 100 *samples per pixel for α estimation.*

5.5.3 Per-pixel α values

Per-pixel α values (cf. Figures 5.7 and 5.8) are presented as RGB colors, the RGB color for each pixel is computed as:

$$(R,G,B) = \left(\frac{\alpha}{\max(\alpha, 1-\alpha)}, \frac{1-\alpha}{\max(\alpha, 1-\alpha)}, 0\right)$$

where the perfect red means samples are all selected from light sources, the perfect green means from BRDFs and the perfect yellow means the classical half-half strategy.

The α value is computed for each light source and can thus be also visualized separately.

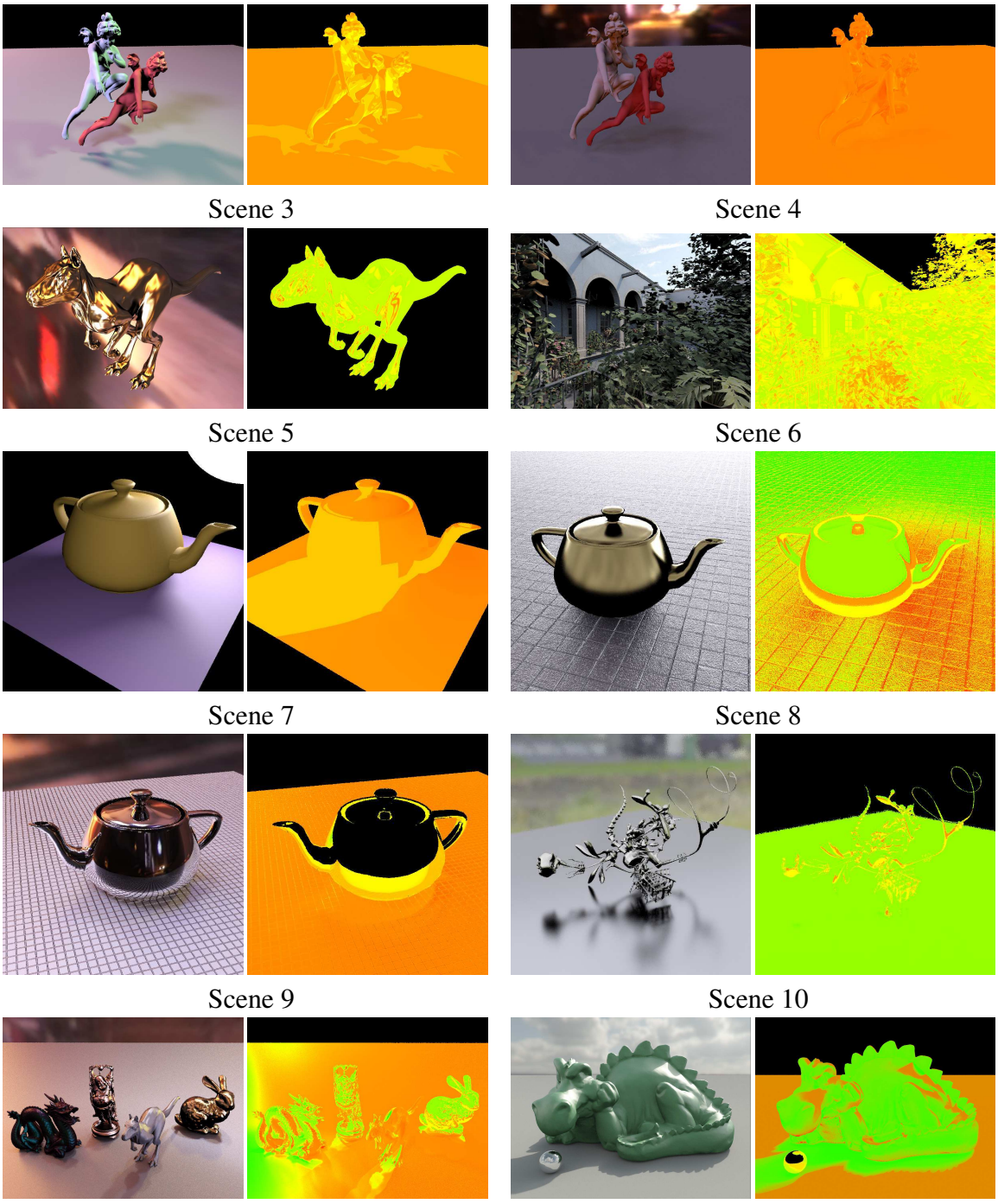


Figure 5.7: *Per-pixel α values. For each scene we show (**Left**) our result and (**Right**) the balancing map which represents how the α value varies spatially. Scene 4 shows light-based sampling only is a good solution whereas scene 10 shows the opposite case. Scene 8 and 11 show large variation of α over pixels.*

We illustrate this feature in Figure 5.8 on scene 1. It clearly shows that the larger the light source is and the closer it is to a glossy BRDF, the more samples from BRDF will be used.

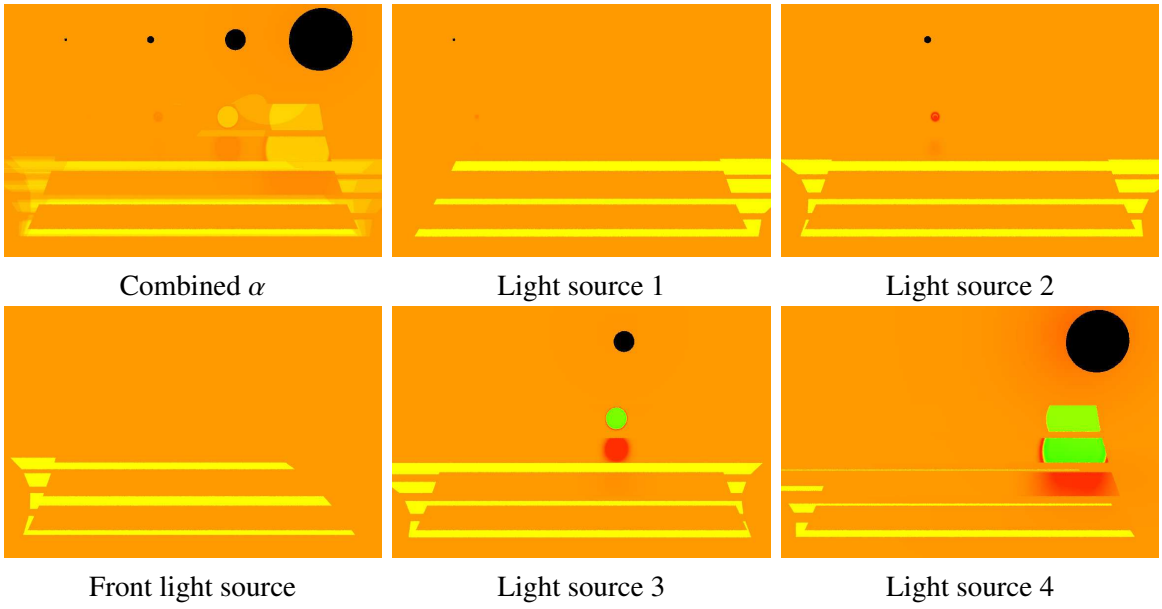


Figure 5.8: *Per-pixel α for each of the five light sources in scene 1. One light source is not directly visible since it is facing the whole scene.*

5.5.4 Lab Difference Images

Figure 5.9 presents, for each scene, Lab difference images between our balancing method and the corresponding reference result rendered with 16000 samples per camera ray and per light source using default balance heuristic technique. Each Lab image is computed from an LDR image obtained after applying the tone-mapping operator (gamma correction) to the corresponding original HDR image. The results show the possibility of arbitrary variations of α over pixels.

For scene 4 and 10, MIS doesn't help much for variance reduction since only the light-based or the BRDF-based sampling technique is efficient for most of pixels. Scene 6 results in default balancing as a good choice and scene 7 shows another static α is appropriate for most of pixels. For scene 2, 8 and 11, α varies significantly. In scene 2, although the dragon is modeled with a glossy material, highlight parts benefit more from light-based sampling. Moreover, although, the ground is rather diffuse, the BRDF-based sampling is preferred for the pixels in the shadow. Scene 8 and 11 give more complex results. Although, their grounds have an unique martial, the α varies significantly between pixels due to the changing of light directions, view directions and occlusions.

Apparently, for high-frequency light sources, the light-based sampling contributes more for highlight parts, even if the marital is quite glossy. However, low-frequency ones lead to opposite results. Occluded pixels benefit more from the BRDF-based sampling compared to the similar pixels that are not occluded.

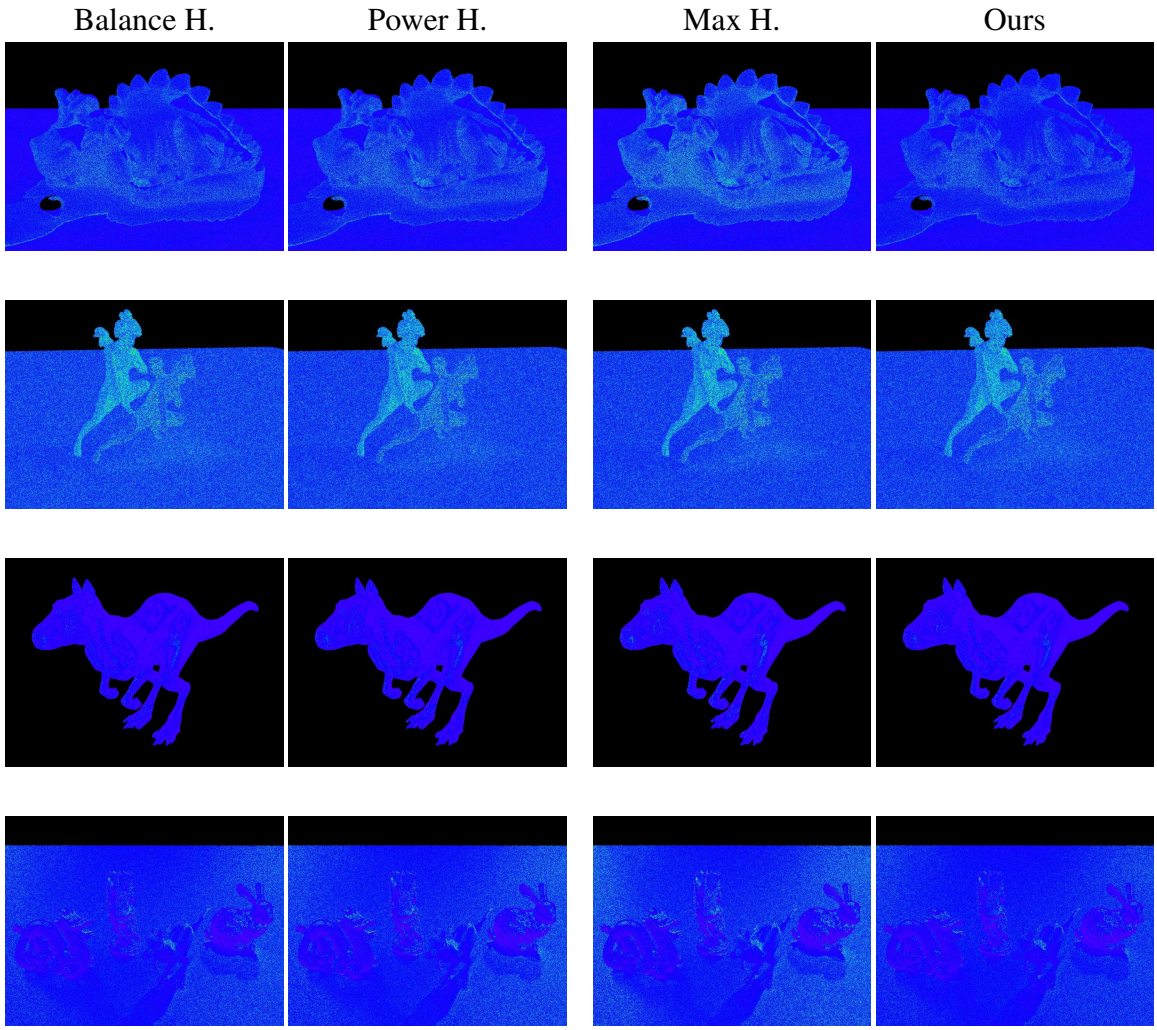


Figure 5.9: ***From top to bottom***. *Lab difference images for Scene 2, 4, 5, 11. The others are in the appendix. The images are computed against corresponding reference solutions.*

5.6 Discussion

Limits of α Estimation As pointed out in Section 5.3.4, the main limitation of our approach is due to the second-order approximation of the variance around $\alpha = 1/2$. This is also illustrated by the fact that restricting the search interval for α to $[0.025, 0.975]$ generally improves the results on our test scenes.

We have investigated third-order approximation but there is generally no real solution to the objective equation. In fact, approximating a positive function (the variance) with a function that varies from -∞ to +∞ (third-order approximation) is very inefficient. In the case of forth-order approximation, the theoretical upper bound is only slightly improved to $\alpha \simeq 0.8$ at the price of a larger computational cost. Finally, we have shown that the use of a non-converged estimation of α may lead to some variance improvements.

Despite these limits and since our technique exhibits some consistent improvements for variance reduction, we believe that it might be a good framework for future studies on balancing criteria. We think that our second-order approximation provides a good trade-off between accuracy and computational cost. In may also be used to progressively estimate α like with a gradient descent: for each step $i + 1$, a second-order Taylor expansion around the previously estimated value α_i will be computed and used to estimate the next α_{i+1}.

More Sampling Strategies We have experimented our approach for balancing only between two strategies that do not require a precomputation step (BRDF-based and light-based importance sampling). The same approach might be use to combine them with other strategies such as the ones based on photon mapping as in [Pajot *et al.* 2011] or visibility-based ones [Ghosh & Heidrich 2006]. For more than two strategies, the minimization of the second-order approximation will result in solving a linear system to find the strategy weights.

Since our approach reduces the sampling variance, it can also benefit to Bidirectional Importance Sampling strategies. For example, Burke et al. [Burke *et al.* 2005] use either BRDF-based or light-based strategies as initialization step. Improved sampling would certainly lead to a lower starting bias. Similarly, our approach can serve as a support for Control Variate techniques (e.g., [Clarberg & Akenine-Möller 2008]) since they can be used on top of state-of-the-art schemes for importance sampling.

Other Future Work Finally, we would like to point out that the recent advances in image denoising (e.g., [Li *et al.* 2012; Rousselle *et al.* 2012; Sen & Darabi 2012]) are complementary to our method. Indeed, these techniques apply a filter on the result of the Monte Carlo renderer and since our technique improves the quality of the Monte Carlo generated image, it could also benefit from these approaches.

5.7 Conclusion

In this chapter, we have introduced a second-order approximation of variance in the context of Multiple Importance Sampling. This approximation leads to an automatic estimation of sample distribution between different sampling strategies. We have demonstrated that our balancing technique reduces, for most cases, the variance and the Lab errors compared to previous MIS approaches (balance, power, and maximum heuristic). Furthermore, for all tested scenes, the efficiency of our method outperforms all previous heuristic techniques. Finally, we have also shown that, it fits into existing MIS approaches and can be implemented on GPU, as well as leads to new visualization tools (static on CPU and dynamic

on GPU). We believe that our approach combined with these new tools will help further investigations on how to develop improved balancing strategies.

Chapter 6

Conclusion and Future Work

6.1 Summary and Contributions

Motivation Realistic rendering results are achieved by simulating light transport. Since light sources initiate the propagation, they are very important for the rendering. Although, the use of classical light source models in physically-based rendering techniques yields visually plausible results, these models are only coarse approximations of real light sources. Generally, the most effective way to capture and integrate realistic light sources is to acquire and model them with a set of images. However, captured data are sometimes too heavy for practical use.

In order to make them more computational friendly, people either compress them with basis or stochastically select only part of them that are important for the illumination.

Problem The light transport problem in non-participating media is modeled by the rendering equation. However, there is generally no analytical solution. Therefore, numerical solutions such as Monte Carlo methods are needed to estimate the equation. Convergence is a criterion to describe the accuracy of Monte Carlo estimation. The convergence rate of these techniques highly depends on the quality of the sampling. A good sampling technique should consider all the factors of the integrand. In the case of global illumination, they are incident lighting, BRDF, visibility and incident cosine. However, it is not easy to do so since all the factors are independent from each other and thus leading to too may numerous combinations.

Contribution In this thesis, we have introduced two importance sampling techniques for realistic light sources. The first one **generates efficiently samples from distant light sources** that have only 2D directional variations. We achieve this in two steps. First, we generate samples according to light intensity. Second, we select a part of the samples that have high contribution due to the cosine factor. To balance between accuracy and efficiency,

we introduced a "pseudo form factor" that approximates the cosine of samples by the cosine of the four corners of the face that they belong to. Our approximation generates more samples from a face where the cosine is positive. Our method is efficient. With a GPU implementation, it achieves interactive performance with 600 dynamic samples per pixel at 1024×768 resolution.

The second importance sampling technique is to **generate importance samples for 4D light field light sources** which have 2D directional and 2D spatial variations. Due to the two-plane parametrization of such light sources, we noticed that only a small part of the lighting contributes to the shading of a point in the scene. Since these small parts change over pixels, we introduced a position-dependent importance sampling technique that generates light samples for each pixel dynamically. A CDF affine transformation is also introduced to generate samples with high intensities. In practical, since these small parts are much smaller than the whole light field, the rendering speed is increased significantly while retaining the details. Our method is efficient, an interactive performance of sampling as well as rendering is achieved with 200 dynamic samples per pixel at 1024×768 resolution when using 99 light fields whose images are composed with 256×256 pixels.

Different type of importance sampling techniques have different advantages. Therefore, combining different sampling techniques together to generate samples is a reasonable way. One direction is to use MIS that combines different techniques with different weights. The more efficient the sampling technique is, the higher its weight should be. However, due to the complexity of a scene, it is very hard to know which sampling technique is better. Moreover, in the case of interactive applications, the characteristics of a scene may be unknown. Therefore, we introduced a "black-box" approach that **approximates the balancing weights for MIS without prior knowledge**. We use second-order approximation for variance minimization. The result of the minimization gives good weights for each sampling techniques leading to a variance reduction. Our approximation is simple, efficient and the whole framework of our method can be applied to both CPU and GPU (interactive). In addition, our method will help to discover new balance strategies.

6.2 Short-term Future work

Our method for far-field environment lighting is limited in terms of visibility. Since we do not take visibility into account to generate samples, some samples are wasted when rendering with occlusions. Therefore, we would like to introduce the visibility to the sampling technique in the future. In order to preserve real-time rendering performance, approximations of visibility might be needed. We can try to follow the idea of [Inger *et al.* 2013] that divide the environment map to small blocks, then use a differential representation of

the visibility function over each block. The sampling takes place for each block afterwards. Additionally, indirect lighting is also an interesting direction for future work. Since our approach achieves quite high performance, there is a remaining large computational budget that can be spent for indirect lighting effects.

In computer games or movies, environment lighting is also very important. For example, in RPG game or racing game, the PC character and the PC car are more important than the other objects in these games. Capturing environment per frame and computing environment lighting with PC objects would bring nice shading effects. Since our method is efficient for realistic lighting, it is also valid for this virtual-realistic lighting. However, there are two remaining problems. The first problem is shadow mapping and visibility coherency. This problem is nearly inevitable since shadows are very important for 3D games with realistic style. As we have discussed in the previous paragraph, the visibility is a remaining problem of our method. Admittedly, for games, shadow mapping techniques are preferred. However, since light samples are dynamically generated for each pixel, generating shadow maps from these dynamic light sources lead to huge computation overhead. Therefore, we have to generate some static light samples for shadow mapping rendering, or we have to select small number of light samples from dynamically generated ones with a new problem that coherency between pixels and frames. Moreover, visibility coherency problem happens between PC objects and NPC objects, since we render them with different light sources. How to make the shadow effects coherent is still a open problem. The second problem is how to capture a plausible lighting environment from these games. This problem is inherent since NPC objects are not infinitive far away from PC objects and sometimes they are even attached to each others (e.g., the car touches the road or the car hits the barrier).

Our method is also helpful to tone mapping study, since we can share the realistic lighting in virtual and real world. If we also design and place the same objects with similar materials, a same rendering result becomes possible. The real scene becomes a true reference. Therefore, tone mapping operator and other lighting effects can be studied with this reference.

Our method for near-field environment lighting is also limited in visibility and indirect lighting. Although we can achieve real-time performance with shadow mapping techniques, the cost is still too much. Using a coarse approximation of the geometry and more shadow maps but with lower resolution (cf. [Ritschel *et al.* 2008]) may better balance the quality and the speed. Furthermore, as presented in the papers [Ritschel *et al.* 2008, 2011], the imperfect shadow map technique has been proved to be a successful solution for indirect light visibility. However, support for indirect lighting is not trivial, since the performance of our method for only direct lighting is already not high enough. A more

efficient solution is needed before extending the method for indirect lighting.

Our method for balancing MIS cannot guarantee variance reduction for each pixel. In the future, we would like to introduce a new simple method to detect these pixels and use default balancing instead or to improve the original method to reduce variance for each pixel.

The current results of our method give some hints on the distribution of samples for each pixel but they are not deeply analyzed. Apparently, for high-frequency light sources, the light-based sampling contributes more for highlight parts whereas low-frequency ones lead to opposite results. However, we still do not know where is the boundary for this change. The same question exists for scene 1 (cf.Figure 5.8). For glossy materials, BRDF-based sampling is preferred but the case suddenly changed when a material is less glossy or the geometry is father. Besides, there is another kind of pattern (cf.Figure 5.8) that green pixels surrounded by red pixels. We still do not know the reason for this phenomenon. It might be caused by limited accuracy of the second-order approximation or might be another boundary for the change of sampling distribution. A reverse way of our method that analyzing the these patterns to find the relationship between the distribution of samples and the rendering factors (lighting, BRDFs and Cosine), would help people find some new balancing strategies. For example, we may find low-frequency distribution patterns for complex illumination. These patterns may be well represented by basis and lead to variance reduction before minimizing variance for each pixel.

It is also possible to integrate our method into multi-pass sampling techniques to improve the quality of the initial samples. Finally, the contribution of indirect illumination is not considered yet when minimizing the variance. As a future work, taking the indirect lighting into account may lead to a better balancing strategy for global illumination. Besides, balancing for photon mapping techniques is also an interesting future work.

6.3 Medium-term Future work

Except for the dedicated future work for far-field and near-field lighting, and balancing scheme for MIS, there are also future research directions for the global background: narrow the gap between virtual world and real world. **The first direction** is to introduce spatial information to environment maps. Since classical environment maps have only 2D directional variations, they hardly represent the spatial variations. For the purpose of recording spatial variations, one way is to sample spatially illumination information by capturing light probes at different positions in the scene (cf. [Tatarchuk 2005; Unger *et al.* 2008]). However, these approaches are too time consuming and have the remaining problem on how to interpolate properly local lighting among recored information. Another way is to use

EnvyDepth [Banterle *et al.* 2013] to reconstruct depth information from a classical environment map. A user guiding step is needed to specify the horizontal plane and the extruded surface. Although the reconstructed environment lighting achieves plausible results (spatial lighting between different objects in the scene), the depth information is still not enough that the reconstructed environment is always a convex bounding box. To the spirit of 4D light field light sources, it is possible to use the same idea for environment map that captures an environment illumination information using the two-plane setup (e.g., one setup for each face). With light field representations, spatial information is able to be recorded for near-field environment lighting.

The second direction is to find a simple and efficient way to capture environment lighting or only part of environment lighting with common devices such as web-cam. Since sampling and rendering with environment map in real time is already done, combining with real-time capturing will lead to some new applications: "let the light come into your computer" which would share the same lighting both in real world and virtual world in real time. The main problem is how to capture without any probes. Since some devices such as smart phones have two cameras, they might be able to capture the environment by following the similar idea as the lady bug camera (cf. Figure 2.18).

The third direction is to speed up 4D light field rendering process. Although real-time rendering using 4D light sources is already done, it is still too costly for practical use. A better engineering work may lead to higher performance. Furthermore, a better representation or even with approximation may be easier balancing the accuracy and speed. For example, we can try to preprocess the acquired data and represent it with other basis that has low computation cost for rendering. Although approximation is introduced, since the frequency of a light source is not always high for each pixel of the light field, the accuracy reduction might be acceptable.

The fourth direction is to capture new data with new devices. Obviously, the original two-plane setup is limited in its projection direction. Artifacts appear on the boundary of the plane. Using other devices, such as cube or even cylinder can remove this limitation and represent more spatial variations. This capture device may also suit the first research direction to capture the whole environment illumination.

6.4 Long-term Future work

Since we would like to narrow the gap between virtual world and real world, capturing only real world information and using it to virtual world is not enough. An echo behavior is also needed: project the information from virtual world to real world. For example, the projected imagery [Raskar *et al.* 1999, 2001] changes or replaces the appearance of physical object

in real world with a projector to provide hybrid visualizations with real physical models and computer graphics techniques. In this research direction, one of the main motivation is also to narrow the gap between the two worlds [Bimber & Raskar 2005]. There are already quite a lot successful applications in this direction [Audet *et al.* 2010; Benko *et al.* 2012; Spindler *et al.* 2009].

A combination of these two research directions (real to virtual and virtual to real) would lead to some interesting applications: we could capture realistic lighting from somewhere, and then use this lighting to illuminate a real object placed somewhere else. Computers would become the media to connect "two" real world together. An ideal system would be like that, a dedicated device (e.g., a camera) is used to capture the illumination of real world lighting, and then transfer the data to another computer placed somewhere with a projector. The projector projects the lighting to a real physical model.

For example, the camera (e.g., the Curiosity [nasa 2013]) is placed on the Mars, the lighting environment can be captured and then transfered to a projected imagery system on the earth, the lighting environment of Mars can be tested in laboratories.

For another example, a two-plane setup is used to measure a very expensive stage light, the lighting of this light is captured and then transfered to some cheaper projectors in the club, then the lighting can be duplicated in different rooms. Since this system is in real time, the stage light can rotate, change color or anything else, and the lighting in other rooms can keep synchronous.

Moreover, this application could also help people to better understanding the human perception and the lighting of the nature.

Since the realistic lighting can be captured and processed in real time, interactive modification is possible. The modification can be implemented as a filter and the realistic lighting is modified by this filter in real time, then the modified lighting is used to lit real physical objects to change the appearance. Since the new appearance can be observed or even captured again, it will be quire helpful for perception study.

To the spirit of the rendering equation, the radiance is computed by BSDF and Lighting. Since lighting can be changed or modified by our system, it will be very helpful to BSDF study. For example, the property of the BSDF can be studied progressively while changing the lighting.

Once we know the property of a material, it will be quite interesting and possible to represent the outgoing radiance with our system. We can first deactivate the real world lighting (in a black room), then design a proper input lighting form the captured realistic lighting. Since the system is in real time, it is easy to adjust different filters to represent required outgoing radiance. For example, project a cartoon character to a real physical model, then this virtual character lives in real world (the latest result related to this objective

is limited to transparent material and the character is thus transparent, and cannot be touch ed since the real physical object is limited to granule only). If the projector is very very bright, it is also possible to do this out of the black room. Some interesting applications may appear: change the object to mirror, change the object to metal, or even change the object to transparent. However, the original lighting still impacts the appearance as an inherent problem.

APPENDIX

A Additional derivation for the second-order Approximation

A.1 From Equation 2 to Equation 3

To estimate the integral of a function $L = \int \mathrm{f}(\boldsymbol{\omega})\,\mathrm{d}\boldsymbol{\omega}$ using random samples $\{\boldsymbol{\omega}_{i=\mathrm{B|L},k=1..K_i}\}$ from two PDFs pdf_B (for BRDF-based strategy to generate K_B samples) and pdf_L (for light-based strategy to generate K_L samples), Veach [Veach & Guibas 1995] has introduced the MIS estimator

$$L_{K_\mathrm{L},K_\mathrm{B}} = \sum_{i=\mathrm{B|L}} \frac{1}{K_i} \sum_{k=1}^{K_i} w_i\left(\boldsymbol{\omega}_{i,k}\right) \frac{\mathrm{f}\left(\boldsymbol{\omega}_{i,k}\right)}{\mathrm{pdf}_i\left(\boldsymbol{\omega}_{i,k}\right)}, \tag{2}$$

where $w_i(\boldsymbol{\omega}_{i,k})$ is a weighting function. For balance heuristic,

$$w_i\left(\boldsymbol{\omega}_{i,k}\right) = \frac{K_i\,\mathrm{pdf}_i\left(\boldsymbol{\omega}_{i,k}\right)}{K_\mathrm{B}\mathrm{pdf}_\mathrm{B}\left(\boldsymbol{\omega}_{i,k}\right) + K_\mathrm{L}\mathrm{pdf}_\mathrm{L}\left(\boldsymbol{\omega}_{i,k}\right)}.$$

By denoting $K = K_\mathrm{B} + K_\mathrm{L}$ (K is the total number of samples) and $K_\mathrm{B} = \alpha K$,

$$w_\mathrm{B}\left(\boldsymbol{\omega}_{\mathrm{B},k}\right) = \frac{\alpha\,\mathrm{pdf}_\mathrm{B}\left(\boldsymbol{\omega}_{\mathrm{B},k}\right)}{\alpha\,\mathrm{pdf}_\mathrm{B}\left(\boldsymbol{\omega}_{\mathrm{B},k}\right) + (1-\alpha)\,\mathrm{pdf}_\mathrm{L}\left(\boldsymbol{\omega}_{\mathrm{B},k}\right)}.$$

and

$$w_\mathrm{L}\left(\boldsymbol{\omega}_{\mathrm{L},k}\right) = \frac{(1-\alpha)\,\mathrm{pdf}_\mathrm{L}\left(\boldsymbol{\omega}_{\mathrm{L},k}\right)}{\alpha\,\mathrm{pdf}_\mathrm{L}\left(\boldsymbol{\omega}_{\mathrm{L},k}\right) + (1-\alpha)\,\mathrm{pdf}_\mathrm{L}\left(\boldsymbol{\omega}_{\mathrm{L},k}\right)}.$$

Introducing everything in Equation 2 leads to

$$L_{\alpha K,(1-\alpha)K} = \frac{1}{\alpha K} \sum_{k=1}^{\alpha K} \frac{\alpha\,\mathrm{f}\left(\boldsymbol{\omega}_{\mathrm{B},k}\right)}{\alpha\,\mathrm{pdf}_\mathrm{B}\left(\boldsymbol{\omega}_{\mathrm{B},k}\right) + (1-\alpha)\,\mathrm{pdf}_\mathrm{L}\left(\boldsymbol{\omega}_{\mathrm{B},k}\right)} + \frac{1}{(1-\alpha)K} \sum_{k=1}^{(1-\alpha)K} \frac{(1-\alpha)\,\mathrm{f}\left(\boldsymbol{\omega}_{\mathrm{L},k}\right)}{\alpha\,\mathrm{pdf}_\mathrm{L}\left(\boldsymbol{\omega}_{\mathrm{L},k}\right) + (1-\alpha)\,\mathrm{pdf}_\mathrm{L}\left(\boldsymbol{\omega}_{\mathrm{L},k}\right)}$$

or

$$L_{K,\alpha} = \frac{1}{K}\sum_{k=1}^{\alpha K}\frac{\mathrm{f}(\boldsymbol{\omega}_{\mathrm{B},k})}{\alpha\,\mathrm{pdf}_{\mathrm{B}}(\boldsymbol{\omega}_{\mathrm{B},k}) + (1-\alpha)\,\mathrm{pdf}_{\mathrm{L}}(\boldsymbol{\omega}_{\mathrm{B},k})} + \frac{1}{K}\sum_{k=1}^{(1-\alpha)K}\frac{\mathrm{f}(\boldsymbol{\omega}_{\mathrm{L},k})}{\alpha\,\mathrm{pdf}_{\mathrm{L}}(\boldsymbol{\omega}_{\mathrm{L},k}) + (1-\alpha)\,\mathrm{pdf}_{\mathrm{L}}(\boldsymbol{\omega}_{\mathrm{L},k})}\,.$$

A simple change of indices from $\{(\mathrm{B}, k = 1..\alpha K)\}$ and $\{(\mathrm{L}, k = 1..(1-\alpha)K)\}$ to $\{i = 1..K\}$ leads to the Defensive Importance Sampling (DIS) [Hesterberg 1995] formula

$$L_{K,\alpha} = \frac{1}{K}\sum_{i=1}^{K}\frac{\mathrm{f}(\boldsymbol{\omega}_i)}{\alpha\,\mathrm{pdf}_{\mathrm{B}}(\boldsymbol{\omega}_i) + (1-\alpha)\,\mathrm{pdf}_{\mathrm{L}}(\boldsymbol{\omega}_i)}. \tag{3}$$

A.2 Computing and Approximating $\mathrm{E}\left[L_{1,\alpha}^2\right]$

Starting from DIS formulation in Equation 3, we have

$$L_{1,\alpha} = \frac{\mathrm{f}(\boldsymbol{\omega}_1)}{\alpha\,\mathrm{pdf}_{\mathrm{B}}(\boldsymbol{\omega}_1) + (1-\alpha)\,\mathrm{pdf}_{\mathrm{L}}(\boldsymbol{\omega}_1)}\,.$$

Since in the case of DIS, the PDF corresponding to the sampling strategy is $\mathrm{pdf}_\alpha(\boldsymbol{\omega}) = \alpha\,\mathrm{pdf}_{\mathrm{B}}(\boldsymbol{\omega}) + (1-\alpha)\,\mathrm{pdf}_{\mathrm{L}}(\boldsymbol{\omega})$, we get

$$\mathrm{E}\left[L_{1,\alpha}^2\right] = \int\left(\frac{\mathrm{f}(\boldsymbol{\omega})}{\mathrm{pdf}_\alpha(\boldsymbol{\omega})}\right)^2 \mathrm{pdf}_\alpha(\boldsymbol{\omega})\;\mathrm{d}\boldsymbol{\omega}\,.$$

It is equivalent to

$$\mathrm{E}\left[L_{1,\alpha}^2\right] = \int\frac{\mathrm{f}^2(\boldsymbol{\omega})}{\mathrm{pdf}_\alpha(\boldsymbol{\omega})}\,\mathrm{d}\boldsymbol{\omega}\,.$$

In the following, for legibility reasons, we will omit $\boldsymbol{\omega}$.

Introducing $\bar{\mathrm{pdf}} = (\mathrm{pdf}_{\mathrm{B}} + \mathrm{pdf}_{\mathrm{L}})/2$ and $\Delta\mathrm{pdf} = (\mathrm{pdf}_{\mathrm{B}} - \mathrm{pdf}_{\mathrm{L}})/2$, we have

$$\mathrm{pdf}_\alpha = \bar{\mathrm{pdf}} + (2\,\alpha - 1)\;\Delta\mathrm{pdf}\,.$$

The Taylor expansion of the N-differentiable rational $1/\mathrm{pdf}_\alpha$ is given by

$$\frac{1}{\mathrm{pdf}_\alpha} \simeq \sum_{n=0}^{N}(-1)^n\,(2\,\alpha - 1)^n\,\frac{1}{\bar{\mathrm{pdf}}^{n+1}}\,.$$

Therefore, we get the final Nth order Taylor expansion

$$\mathrm{E}\left[L_{1,\alpha}^2\right] \simeq \sum_{n=0}^{N}(-1)^n\,(2\,\alpha - 1)^n\int\frac{\mathrm{f}^2}{\bar{\mathrm{pdf}}^{n+1}}\,.$$

B All Results of Lab Images

	Method	*Scene*										
		1	2	3	4	5	6	7	8	9	10	11
Var.	cf. Table 5.1											
	BRDF	67.3	4597	24.14	0.294	0.469	1.260	5.39	2.070	**0.0198**	139.75	0.378
	Light	0.47	**2.56**	**0.005**	**0.014**	4.131	0.088	61.7	0.036	118.20	0.776	0.127
	Balanced H.	0.90	4.76	0.096	0.027	0.410	0.120	5.76	0.043	0.250	0.402	0.055
	Power H.	0.99	4.87	0.097	0.030	0.450	0.138	6.42	0.046	0.255	0.440	0.059
	Max. H.	1.24	5.11	0.097	0.035	0.550	0.157	9.78	0.061	0.256	0.618	0.075
	Pre. α	**0.42**	2.95	0.035	0.016	**0.27**	**0.047**	**5.09**	**0.033**	0.095	**0.301**	**0.045**
	Online α	0.66	3.88	0.065	0.022	**0.35**	**0.086**	5.68	0.037	0.1700	**0.359**	**0.051**
Lab error	BRDF	14.9	6.5	10.5	2.80	1.10	22.9	2.00	3.85	**0.36**	5.26	3.38
	Light	0.14	1.11	**0.25**	**0.78**	7.50	0.23	3.60	1.16	15.9	2.00	2.25
	Balance H.	0.14	0.81	0.31	0.91	0.77	0.28	1.04	1.21	0.47	1.65	1.54
	Power H.	**0.11**	0.83	0.31	0.93	0.78	0.28	1.06	1.22	0.46	1.70	1.59
	Max. H.	0.12	0.95	0.31	0.99	0.85	0.28	1.30	1.37	0.45	1.98	1.71
	Online α	0.13	**0.78**	0.29	0.86	**0.74**	**0.26**	**1.02**	**1.15**	0.44	**1.60**	**1.51**
Time (sec.)	BRDF	**158.6**	138.1	**40.6**	55.1	33.7	**16.9**	59.1	36.5	49.2	354	91.0
	Light	299.7	**102.8**	147.4	**41.8**	**18.6**	33.8	**41.7**	**34.2**	**27.1**	**178**	**53.0**
	Balance H.	237.7	123.5	95.5	48.9	28.0	25.5	52.0	36.2	40.0	281	74.0
	Power H.	239.9	122.3	95.7	48.8	27.1	24.9	49.9	35.9	39.2	275	74.0
	Max. H.	239.0	121.6	95.4	48.1	26.0	24.5	49.5	34.9	38.5	264	71.1
	Online α	250.8	128.3	106.9	45.8	28.3	26.9	51.1	34.5	40.9	281	72.0
Efficiency	BRDF	0.0001	0.0000	0.0010	0.0617	0.0633	0.0470	0.0031	0.0132	**1.0265**	0.0000	0.0291
	Light	**0.0071**	**0.0038**	**1.3569**	**1.7088**	0.0130	0.3362	0.0004	**0.8122**	0.0003	0.0072	0.1486
	Balance H.	0.0047	0.0017	0.1091	0.7574	0.0871	0.3268	0.0033	0.6424	0.1000	0.0089	0.2457
	Power H.	0.0042	0.0017	0.1077	0.6831	0.0820	0.2910	0.0031	0.6055	0.1000	0.0083	0.2290
	Max. H.	0.0034	0.0016	0.1081	0.5940	0.0699	0.2600	0.0021	0.4697	0.1015	0.0061	0.1875
	Online α	0.0060	0.0020	0.1439	0.9925	**0.1010**	**0.4323**	**0.0034**	0.7834	0.1438	**0.0099**	**0.2723**

Table 1: *Average variance and Lab error, rendering time and efficiency for each scene and for different balancing methods. We use $N = 256$ samples per light source for each method. Integrated estimation of α refers to the estimator introduced in Equation 7 in the paper where we use half of the total samples to evaluate α ($M = 128$) and the remaining half to evaluate the radiance with the estimated α. The Lab error is computed against the reference solution. The efficiency [Veach 1998] of each strategy is computed as the inverse of the product of the variance and the rendering time.*

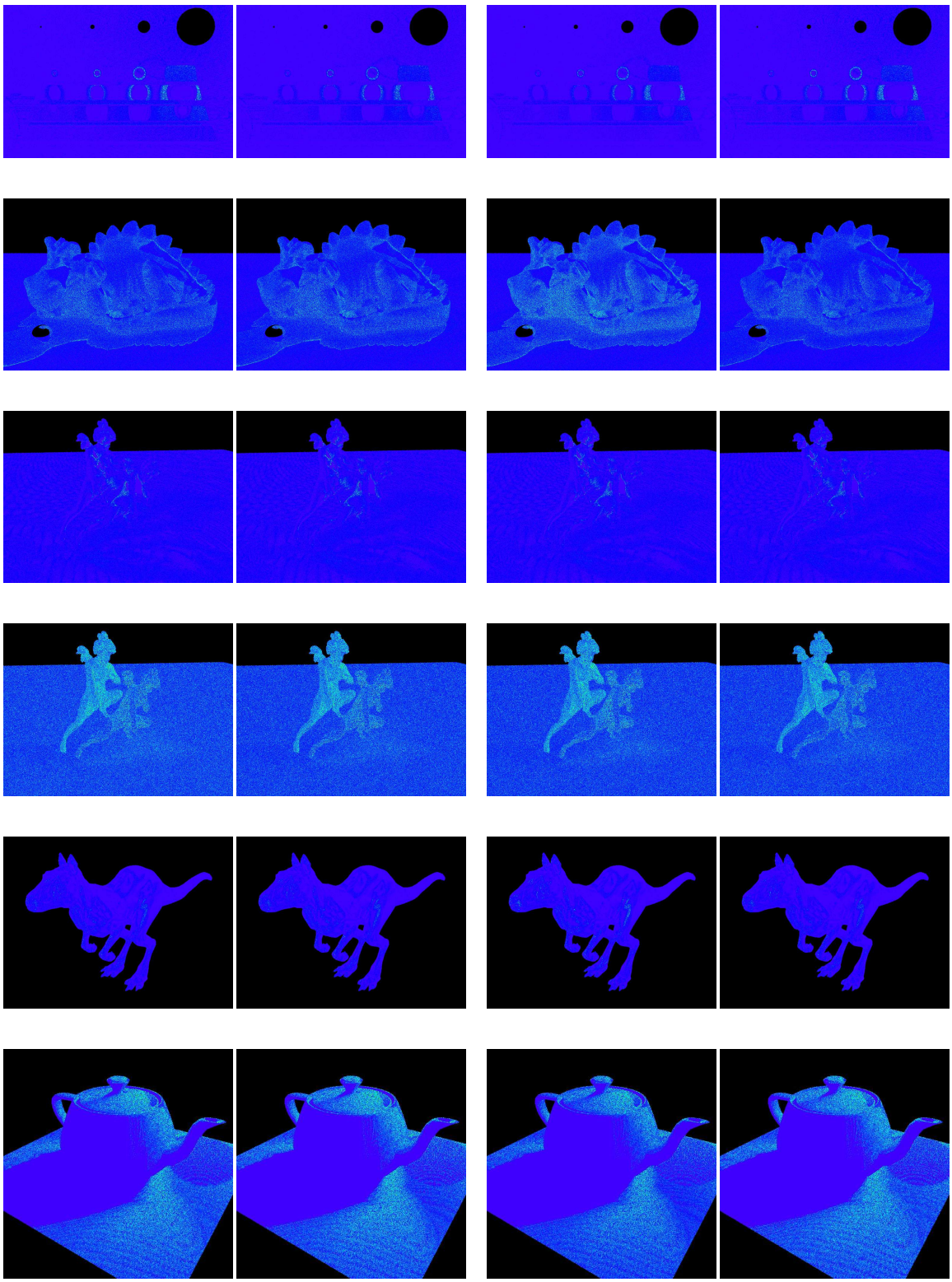


Figure 1: ***From left to right****. Lab difference images for the balance, power, max heuristics and ours (scenes 1 to 6).*

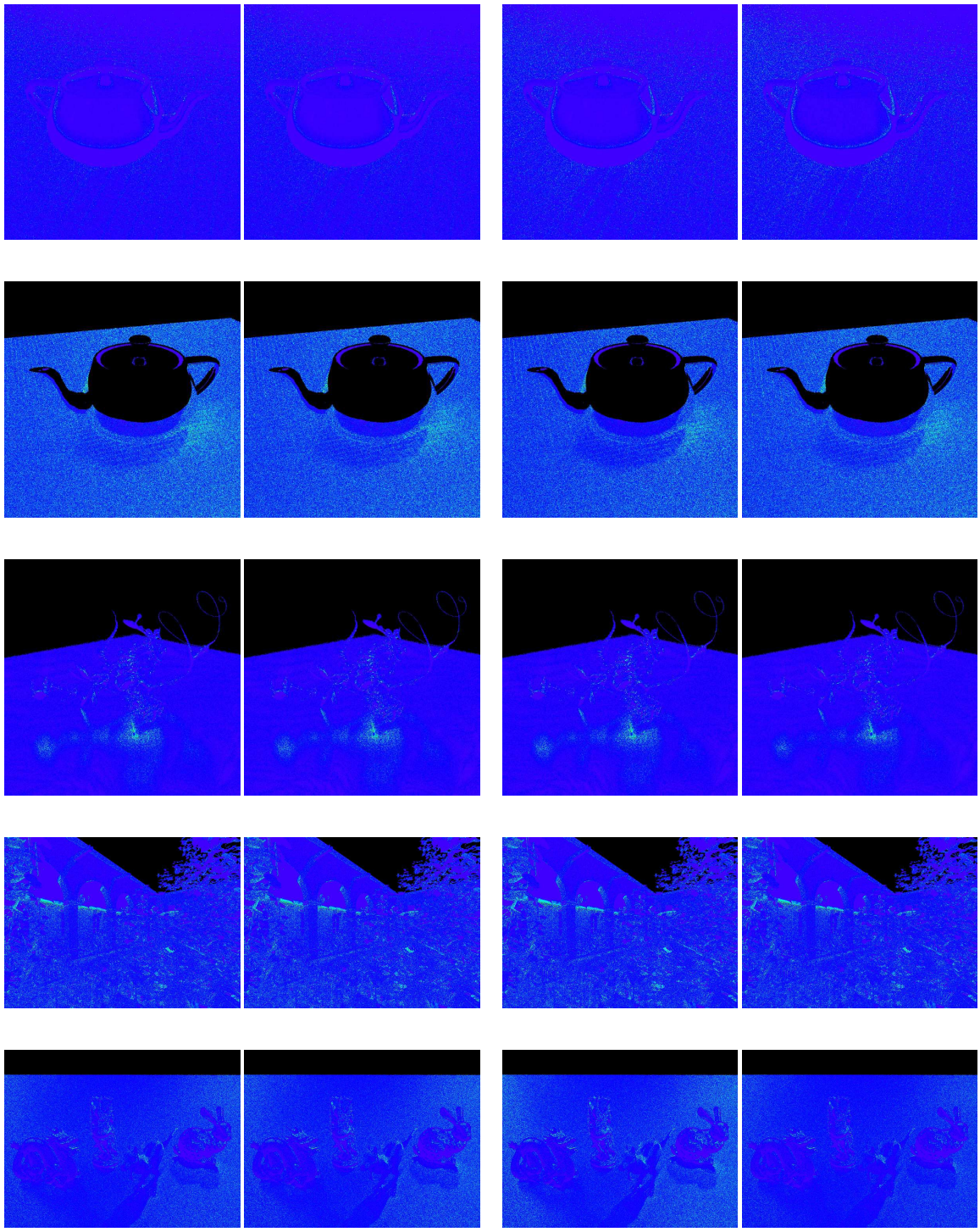


Figure 2: ***From left to right****. Lab difference images for the balance, power, max heuristics and ours (scenes 7 to 11).*

REFERENCES

AGARWAL, S., RAMAMOORTHI, R., BELONGIE, S. & JENSEN, H.W. (2003). Structured importance sampling of environment maps. *ACM Transactions on Graphics*, **22**, 605–612. 69, 92

AMANATIDES, J. (1984). Ray tracing with cones. *SIGGRAPH Comput. Graph.*, **18**, 129–135. 32

ANNEN, T., MERTENS, T., SEIDEL, H.P., FLERACKERS, E. & KAUTZ, J. (2008). Exponential shadow maps. In *Proceedings of Graphics Interface*, 155–161, Canadian Information Processing Society. 32

APPLE (2007). OpenCL Licensing and Trademarks. 4

ARBREE, A., WALTER, B. & BALA, K. (2011). Heterogeneous subsurface scattering using the finite element method. *IEEE Transactions on Visualization and Computer Graphics*, **17**, 956–969. 13

ARVO, J., TORRANCE, K. & SMITS, B. (1994). A framework for the analysis of error in global illumination algorithms. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, SIGGRAPH '94, 75–84, ACM. 9

ASHDOWN, I. (1993). Near-Field Photometry: A New Approach. *J. Illum. Eng. Soc.*, **22**, 163–180. 44

ASHDOWN, I. (1995). Near-Field Photometry: Measuring and Modeling Complex 3-D Light Sources. In *SIGGRAPH 95 Course Notes - Realistic Input for Realistic Images*, 1–15, ACM Press. 44, 68

ASHDOWN, I. & RYKOWSKI, R. (1997). Making Near-Field Photometry Practical. In *1997 IESNA Conference Proceedings*, 368–389, Illuminating Engineering Society of North America. 2, 39, 68

ASHIKHMIN, M. & PREMOZE, S. (2007). Distribution-based BRDFs. Tech. rep., Univ. of Utah, `http://www.cs.utah.edu/~premoze/dbrdf`. 92

Audet, S., Okutomi, M. & Tanaka, M. (2010). Direct image alignment of projector-camera systems with planar surfaces. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 303–310. 116

Banterle, F., Callieri, M., Dellepiane, M., Corsini, M., Pellacini, F. & Scopigno, R. (2013). Envydepth: An interface for recovering local natural illumination from environment maps. In *Pacific Graphics 2013*, Eurographics Association and Blackwell. 115

Benko, H., Jota, R. & Wilson, A. (2012). Miragetable: freehand interaction on a projected augmented reality tabletop. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, 199–208, ACM. 116

Bimber, O. & Raskar, R. (2005). *Spatial Augmented Reality: Merging Real and Virtual Worlds*. A. K. Peters, Ltd. 116

Burke, D., Ghosh, A. & Heidrich, W. (2005). Bidirectional Importance Sampling for Direct Illumination. In *Proceedings of Eurographics Symposium on Rendering*, 147–156. 16, 23, 93, 109

Carr, N.A., Hall, J.D. & Hart, J.C. (2003). Gpu algorithms for radiosity and subsurface scattering. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, HWWS '03, 51–59, Eurographics Association. 13

Chen, S.E., Rushmeier, H.E., Miller, G. & Turner, D. (1991). A progressive multi-pass method for global illumination. *SIGGRAPH Comput. Graph.*, **25**, 165–174. 30

Cheslack-Postava, E., Wang, R., Akerlund, O. & Pellacini, F. (2008). Fast, realistic lighting and material design using nonlinear cut approximation. *ACM Transactions on Graphics*, **27**, 128:1–128:10. 23

Chow, A. (2010). *Orthogonal and Symmetric Haar Wavelets on the Three-Dimensional Ball*. Ph.D. thesis, University of Toronto. 34

Clarberg, P. & Akenine-Möller, T. (2008). Exploiting Visibility Correlation in Direct Illumination. *Computer Graphics Forum*, **27**, 1125–1136. 89, 109

Clarberg, P. & Akenine-Möller, T. (2008). Practical Product Importance Sampling for Direct Illumination. *Computer Graphics Forum*, **27**. 69, 90, 93

Clarberg, P., Jarosz, W., Akenine-Möller, T. & Jensen, H.W. (2005). Wavelet importance sampling: efficiently evaluating products of complex functions. *ACM Transactions on Graphics*, **24**, 1166–1175. 23, 69

Cline, D., Egbert, P.K., Talbot, J. & Cardon, D.L. (2006). Two Stage Importance Sampling for Direct Lighting. In *Proceedings of Eurographics Symposium on Rendering*, 103–113. 23, 69, 70, 90, 93

Cohen, M.F. & Greenberg, D.P. (1985). The hemi-cube: a radiosity solution for complex environments. *SIGGRAPH Comput. Graph.*, **19**, 31–40. 32

Cohen, M.F., Chen, S.E., Wallace, J.R. & Greenberg, D.P. (1988). A progressive refinement approach to fast radiosity image generation. *SIGGRAPH Comput. Graph.*, **22**, 75–84. 13

Dachsbacher, C., Křivánek, J., Hašan, M., Arbree, A., Walter, B. & Novák, J. (2013). Scalable realistic rendering with many-light methods. *Computer Graphics Forum*, **32**. 10, 36

Davidovič, T., Křivánek, J., Hašan, M., Slusallek, P. & Bala, K. (2010). Combining global and local virtual lights for detailed glossy illumination. *ACM Transactions on Graphics*, **29**, 143:1–143:8. 37

Debevec, P. (1998). Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, 189–198, ACM. 3, 41

Debevec, P. & Malik, J. (1997). Recovering High Dynamic Range Radiance Maps from Photographs. In *Proceedings of SIGGRAPH '97*, 369–378, ACM Press/Addison-Wesley Publishing Co. 49

Engelhardt, T., Novák, J., Schmidt, T.W. & Dachsbacher, C. (2009). Approximate bias compensation for rendering scenes with heterogeneous participating media. *Computer Graphics Forum (Proceedings of Pacific Graphics 2012)*, **31**, 141:1–141:8. 37

Ergun, S., Kurt, M. & Öztürk, A. (2012). Real-time Kd-tree Based Importance Sampling of Environment Maps. In *Proceedings of Spring Conference on Computer Graphics*, 84–91, Comenius University Press. 92

Fishman, G.S. (1996). *Monte Carlo: Concepts, algorithms, and applications*. Springer Series in Operations Research, Springer-Verlag. 77

Georgiev, I., Křivánek, J., Popov, S. & Slusallek, P. (2012). Importance caching for complex illumination. *Computer Graphics Forum*, **31**, 701–710. 25, 26, 93, 95, 98

Ghosh, A. & Heidrich, W. (2006). Correlated visibility sampling for direct illumination. *Vis. Comput.*, **22**, 693–701. 92, 109

Goesele, M., Granier, X., Heidrich, W. & Seidel, H.P. (2003). Accurate light source acquisition and rendering. *ACM Transactions on Graphics*, **22**, 621–630. 2, 6, 39, 44, 45, 46, 47, 68, 69, 71, 72, 73, 74, 76, 80, 83, 84

Goral, C.M., Torrance, K.E., Greenberg, D.P. & Battaile, B. (1984). Modeling the interaction of light between diffuse surfaces. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '84, 213–222, ACM. 12

Gortler, S.J., Grzeszczuk, R., Szelinski, R. & Cohen, M.F. (1996). The Lumigraph. In *Proceedings of ACM SIGGRAPH*, 43–54, ACM Press/Addison-Wesley Publishing Co. 43, 71

Granier, X., Drettakis, G. & Walter, B. (2000). Fast Global Illumination Including Specular Effects. In B. Peroche & H. Rushmeier, eds., *Rendering Techniques 2000 (Proceedings of the Eleventh Eurographics Workshop on Rendering)*, 47 – 59, Eurographics, Springer Wien. 30

Granier, X., Goesele, M., Heidrich, W. & Seidel, H.P. (2003). Interactive Visualization of Complex Real-World Light Sources. In *Proceedings of Pacific Conference on Computer Graphics and Applications*, 59–66, IEEE Computer Society. 6, 10, 47, 68, 69

Green, P., Kautz, J., Matusik, W. & Durand, F. (2006). View-dependent precomputed light transport using nonlinear gaussian function approximations. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games*, I3D '06, 7–14, ACM. 35

Greenberg, D.P., Cohen, M.F. & Torrance, K.E. (1986). Radiosity: A method for computing global illumination. *The Visual Computer*, **2**, 291–297. 11, 13

Hachisuka, T. & Jensen, H.W. (2009). Stochastic progressive photon mapping. *ACM Transactions on Graphics*, **28**, 141:1–141:8. 27, 29

Hachisuka, T., Ogaki, S. & Jensen, H.W. (2008). Progressive photon mapping. *ACM Transactions on Graphics*, **27**, 130:1–130:8. 1, 27, 29

Halle, M.W. (1994). Holographic stereograms as discrete imaging systems. In S.A. Benton, ed., *Practical Holography VIII*, vol. 2176, 73–84, SPIE. 45

Halton, J.H. (1964). Algorithm 247: Radical-inverse quasi-random point sequence. *Commun. ACM*, **7**, 701–702. 18

Hammersley, J.M. (1960). Monte carlo methods for solving multivariable problems. *Annals of the New York Academy of Sciences*, **86**, 844–874. 18

Harris, M., Sengupta, S. & Owens, J. (2007). Parallel prefix sum (scan) with CUDA. *GPU Gems*, **3**, 851–876. 53, 60

Havran, V., Smyk, M., Krawczyk, G., Myszkowski, K. & Seidel, H.P. (2005). Interactive System for Dynamic Scene Lighting using Captured Video Environment Maps. In *Proceedings of Eurographics Symposium on Rendering*, 31–42. 36, 51, 52, 92

Hašan, M., Pellacini, F. & Bala, K. (2006). Direct-to-indirect transfer for cinematic relighting. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, 1089–1097, ACM. 34

Hašan, M., Pellacini, F. & Bala, K. (2007). Matrix row-column sampling for the many-light problem. *ACM Transactions on Graphics*, **26**. 37, 38

Hašan, M., Velázquez-Armendariz, E., Pellacini, F. & Bala, K. (2008). Tensor clustering for rendering many-light animations. In *Proceedings of the Nineteenth Eurographics conference on Rendering*, Proceedings of Eurographics Symposium on Rendering'08, 1105–1114, Eurographics Association. 38

Hašan, M., Křivánek, J., Walter, B. & Bala, K. (2009). Virtual spherical lights for many-light rendering of glossy scenes. *ACM Transactions on Graphics*, **28**, 143:1–143:6. 36, 37, 38, 69

Heckbert, P.S. (1990). Adaptive radiosity textures for bidirectional ray tracing. In *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '90, 145–154, ACM. 29

Heckbert, P.S. (1991). *Simulating global illumination using adaptive meshing*. Ph.D. thesis, University of California, Berkeley. 11

Heidrich, W., Kautz, J., Slusallek, P. & Seidel, H.P. (1998). Canned Lightsources. In *Rendering Techniques '98*, 293–300, Eurographics. 71, 72, 73, 74, 75, 76

Hesterberg, T. (1995). Weighted Average Importance Sampling and Defensive Mixture Distributions. *Technometrics*, **37**, 185–194. 95, 120

IESNA Committee (2001). *Standard File Format for Electronic Transfer of Data: LM-63*. Lighting Measurements Series, Illuminating Engineering Society of North America. 68

Inger, Y., Farbman, Z. & Lischinski, D. (2013). Locally Adaptive Products for All-Frequency Relighting. *Computer Graphics Forum*, **32**, 73–82. 112

Jakob, W. & Marschner, S. (2012). Manifold exploration: A markov chain monte carlo technique for rendering scenes with difficult specular transport. *ACM Transactions on Graphics*, **31**, 58:1–58:13. 27

Jakob, W., Arbree, A., Moon, J.T., Bala, K. & Marschner, S. (2010). A radiative transfer framework for rendering materials with anisotropic structure. *ACM Transactions on Graphics*, **29**, 53:1–53:13. 13

Jenkins, D.R. & Mönch, H. (2000). Source Imaging Goniometer Method of Light Source Characterization for Accurate Projection System Design. In *Proceedings of SID (Society for Information Display) Conference*, 862–865. 44

Jensen, H.W. (1996a). Global Illumination Using Photon Maps. In *Proceedings of Eurographics Workshop on Rendering)*, 21–30, Eurographics, Springer-Verlag/Wien. 1

Jensen, H.W. (1996b). Global illumination using photon maps. In *Proceedings of the eurographics workshop on Rendering techniques '96*, 21–30, Springer-Verlag. 11, 29

Jensen, H.W. (2001). *Realistic Image Synthesis Using Photon Mapping*. A. K. Peters. 69

Kajiya, J.T. (1986). The rendering equation. In *Proceedings of SIGGRAPH '86*, 143–150, ACM. 1, 10, 11, 12, 26, 92

Kaplanyan, A. & Dachsbacher, C. (2010). Cascaded light propagation volumes for real-time indirect illumination. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, I3D '10, 99–107, ACM. 33

Kaplanyan, A.S. & Dachsbacher, C. (2013a). Adaptive progressive photon mapping. *ACM Transactions on Graphics*, **32**, 16:1–16:13. 1, 29

Kaplanyan, A.S. & Dachsbacher, C. (2013b). Path space regularization for holistic and robust light transport. *Computer Graphics Forum (Proceedings of Eurographics 2013)*, **32**. 27, 38

Kautz, J., Sloan, P.P. & Snyder, J. (2002). Fast, arbitrary brdf shading for low-frequency lighting using spherical harmonics. In *Proceedings of the 13th Eurographics workshop on Rendering*, EGRW '02, 291–296, Eurographics Association. 32

Keller, A. (1997). Instant radiosity. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, 49–56, ACM Press/Addison-Wesley Publishing Co. 36

Keller, A. & Heidrich, W. (2001). Interleaved sampling. In *Proceedings of Eurographics Workshop on Rendering*. 60, 82

Kerr, W.B., Pellacini, F. & Denning, J.D. (2010). Bendylights: Artistic control of direct illumination by curving light rays. *Comput. Graph. Forum*, **29**, 1451–1459. 31

khronos (2008). OpenCL-The open standard for parallel programming of heterogeneous systems. 4

Kim, C., Zimmer, H., Pritch, Y., Sorkine-Hornung, A. & Gross, M. (2013). Scene reconstruction from high spatio-angular resolution light fields. *ACM Transactions on Graphics*, **32**, 73:1–73:12. 44

Knaus, C. & Zwicker, M. (2011). Progressive photon mapping: A probabilistic approach. *ACM Transactions on Graphics*, **30**, 25:1–25:13. 69

Kollig, T. & Keller, A. (2004). Illumination in the presence of weak singularities. In *Monte Carlo and Quasi-Monte Carlo Methods 2004*, 245–257, Springer. 38

Lafortune, E.P. & Willems, Y.D. (1993). Bi-directional Path Tracing. In *Proceedings of International Conference on Computational Graphics and Visualization Techniques (Compugraphics '93)*, 145–153. 1, 27

Lafortune, E.P. & Willems, Y.D. (1994). Using the modified phong reflectance model for physically based rendering. *Report CW*, **197**, 2–4. 21, 58

Laine, S., Saransaari, H., Kontkanen, J., Lehtinen, J. & Aila, T. (2007). Incremental instant radiosity for real-time indirect illumination. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, Proceedings of Eurographics Symposium on Rendering'07, 277–286, Eurographics Association. 38

Lawrence, J., Rusinkiewicz, S. & Ramamoorthi, R. (2004). Efficient BRDF importance sampling using a factored representation. In *Proceedings of ACM SIGGRAPH '04*, 496–505. 92

Lehtinen, J., Karras, T., Laine, S., Aittala, M., Durand, F. & Aila, T. (2013). Gradient-domain metropolis light transport. *ACM Transactions on Graphics*, **32**, 95:1–95:12. 28

Lessig, C. & Fiume, E. (2008). Soho: Orthogonal and symmetric haar wavelets on the sphere. *ACM Transactions on Graphics*, **27**, 4:1–4:11. 34

Levoy, M. & Hanrahan, P. (1996). Light Field Rendering. In *Proceedings of ACM SIGGRAPH*, 31–42, ACM Press/Addison-Wesley Publishing Co. 45

Li, S., Guennebaud, G., Yang, B. & Feng, J. (2011). Predicted Virtual Soft Shadow Maps with High Quality Filtering. *Computer Graphics Forum*, **30**, bird associated team. 32

Li, T.M., Wu, Y.T. & Chuang, Y.Y. (2012). Sure-based optimization for adaptive sampling and reconstruction. *ACM Transactions on Graphics*, **31**, 194:1–194:9. 109

Liu, X., Sloan, P.P., Shum, H.Y. & Snyder, J. (2004). All-frequency precomputed radiance transfer for glossy objects. In *Proceedings of the Fifteenth Eurographics conference on Rendering Techniques*, Proceedings of Eurographics Symposium on Rendering'04, 337–344, Eurographics Association. 34, 35

Loos, B.J., Antani, L., Mitchell, K., Nowrouzezahrai, D., Jarosz, W. & Sloan, P.P. (2011). Modular radiance transfer. *ACM Transactions on Graphics*, **30**, 178:1–178:10. 35

Loos, B.J., Nowrouzezahrai, D., Jarosz, W. & Sloan, P.P. (2012). Delta radiance transfer. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '12, 191–196, ACM. 35

Lu, H., Bao, P. & Feng, J. (2012). OpenCL-based Real-time KD-Tree and Raytracing for Dynamic Scene. In *CAD/CG'2012 conference*, Tsinghua University. 28

Lu, H., Pacanowski, R. & Granier, X. (2013a). A Second-Order Approximation for Variance Reduction in Multiple Importance Sampling. In *Pacific Graphics 2013*, Eurographics Association and Blackwell. 92

Lu, H., Pacanowski, R. & Granier, X. (2013b). Real-Time Importance Sampling of Dynamic Environment Maps. In *Eurographics Annual Conference (Short Papers)*, Eurographics. 49, 104

Mahajan, D., Tseng, Y.T. & Ramamoorthi, R. (2008). An Analysis of the In-Out BRDF Factorization for View-Dependent Relighting. *Comput. Graph. Forum*, **27**, 1137–1145. 33, 51

Mara, M., McGuire, M. & Luebke, D. (2013). Toward practical real-time photon mapping: Efficient gpu density estimation. In *Interactive 3D Graphics and Games 2013*. 30

Mas, A., Martin, I. & Patow, G. (2008). Compression and Importance Sampling of Near-Field Light Sources. *Computer Graphics Forum*, **27**, 2013–2027. 39, 68, 69

Matsumoto, M. & Nishimura, T. (1998). Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *Acm Transactions on Modeling and Computer Simulation*, **8**, 3–30. 18, 83

McMillan, L. & Gortler, S. (1999). Image-based rendering: A new interface between computer vision and computer graphics. *SIGGRAPH Comput. Graph.*, **33**, 61–64. 45

Microsoft (2013). Direct3D 11 Graphics. 4

Montes, R., Urena, C., Garcia, R. & Lastra, M. (2008). Generic BRDF sampling: A sampling method for global illumination. In *Proceedings of GRAPP '08*, 191–198. 92

nasa (2013). Curiosity - mars science laboratory. Accessed: 2013-10-08. 116

Neumann, C. (1877). *Untersuchungen über das Logarithmische und Newton'sche potential*. Druck und Verlag von B. G. Teubner. 12

Ng, R., Ramamoorthi, R. & Hanrahan, P. (2003). All-frequency shadows using non-linear wavelet lighting approximation. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, 376–381, ACM. 33

Ng, R., Ramamoorthi, R. & Hanrahan, P. (2004). Triple product wavelet integrals for all-frequency relighting. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, 477–487, ACM. 35

Nicodemus, F.E. (1965). Directional reflectance and emissivity of an opaque surface. *Applied Optics*, **4**, 767–773. 11

Niederreiter, H. (1992). *Random number generation and quasi-Monte Carlo methods*. Society for Industrial and Applied Mathematics. 82

Nishita, T., Okamura, I. & Nakamae, E. (1985). Shading models for point and linear sources. *ACM Transactions on Graphics*, **4**, 124–146. 31

NVIDIA (2006a). CUDA Parallel Computing Platform. 4

NVIDIA (2006b). Gefore 8. 3

Ostromoukhov, V., Donohue, C. & Jodoin, P.M. (2004). Fast hierarchical importance sampling with blue noise properties. In *Proceedings of SIGGRAPH '04*, 488–495, ACM. 52, 92

Ou, J. & Pellacini, F. (2011). Lightslice: matrix slice sampling for the many-lights problem. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, SA '11, 179:1–179:8, ACM. 37

Owen, A. & Zhou, Y. (2000). Safe and Effective Importance Sampling. *Journal of the American Statistical Association*, **95**, 135–143. 93

Pacanowski, R., Celis, O.S., Schlick, C., Granier, X., Poulin, P. & Cuyt, A. (2012). Rational BRDF. *IEEE Transactions Visualization and Computer Graphics*, **18**, 1824–1835. 58, 92

Pajot, A., Barthe, L., Paulin, M. & Poulin, P. (2011). Representativity for robust and adaptive multiple importance sampling. *IEEE Transactions Visualization and Computer Graphics*, **17**, 1108–1121. 16, 24, 90, 93, 109

Papoulis, A. (1978). *Signal analysis*, vol. 191. McGraw-Hill. 32

Pharr, M. & Fernando, R. (2005). *GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation (Gpu Gems)*. Addison-Wesley Professional. 13

Pharr, M. & Humphreys, G. (2004). *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers. 1, 7, 9, 18, 19, 32, 60, 98

Picott, K.P. (1992). Extensions of the linear and area lighting models. *IEEE Comput. Graph. Appl.*, **12**, 31–38. 32

Poulin, P. & Amanatides, J. (1991). Shading and shadowing with linear light sources. *Computers and Graphics*, **15**, 259–265. 32

Press, O.U. (2013). Oxford dictionaries. `http://oxforddictionaries.com/`, accessed: 2013-08-08. 9

Purcell, T.J., Buck, I., Mark, W.R. & Hanrahan, P. (2002). Ray tracing on programmable graphics hardware. *ACM Transactions on Graphics*, **21**, 703–712. 28

Ramamoorthi, R. & Hanrahan, P. (2001). An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, 497–500, ACM. 32

Raskar, R., Welch, G. & Chen, W.C. (1999). Table-top spatially-augmented reality: Bringing physical models to life with projected imagery. In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, IWAR '99, 64–, IEEE Computer Society. 115

Raskar, R., Welch, G., Low, K.L. & Bandyopadhyay, D. (2001). Shader lamps: Animating real objects with image-based illumination. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, 89–102, Springer-Verlag. 115

Reinhard, E., Heidrich, W., Debevec, P., Pattanaik, S., Ward, G. & Myszkowski, K. (2010). *High Dynamic Range Imaging: Acquisition, Display and Image-Based Lighting*. Morgan Kaufmann Publishers, 2nd edition. 40, 81

Ritschel, T., Grosch, T., Kim, M.H., Seidel, H.P., Dachsbacher, C. & Kautz, J. (2008). Imperfect Shadow Maps for Efficient Computation of Indirect Illumination. *ACM Transactions on Graphics*, **27**. 32, 113

Ritschel, T., Eisemann, E., Ha, I., Kim, J.D. & Seidel, H.P. (2011). Making imperfect shadow maps view-adaptive: High-quality global illumination in large dynamic scenes. *Computer Graphics Forum (presented at EGSR 2011)*. 37, 113

Ritschel, T., Dachsbacher, C., Grosch, T. & Kautz, J. (2012). The State of the Art in Interactive Global Illumination. *Computer Graphics Forum*, **31**, 160–188. 1, 68

Roger, D., Assarsson, U. & Holzschuch, N. (2007). Efficient Stream Reduction on the GPU. In *Workshop on General Purpose Processing on Graphics Processing Units*. 60

Rousselle, F., Knaus, C. & Zwicker, M. (2012). Adaptive rendering with non-local means filtering. *ACM Transactions on Graphics*, **31**, 195:1–195:11. 109

Rykowski, R.F. & Wooley, C.B. (1997). Source Modeling for Illumination Design. In R.E. Fischer, R.B. Johnson, R.C. Juergens, W.J. Smith & P.R.Y. Jr., eds., *Lens Design, Illumination, and Optomechanical Modeling*, vol. 3130, 204–208, SPIE. 44

Schmidt, T.W., Novak, J., Meng, J., Kaplanyan, A.S., Reiner, T., Nowrouzezahrai, D. & Dachsbacher, C. (2013). Path-space manipulation of physically-based light transport. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2013)*, **32**. 27

Schröder, P. & Sweldens, W. (1995). Spherical wavelets: efficiently representing functions on the sphere. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, 161–172, ACM. 34

Segal, M. & Akeley, K. (2013). The OpenGL® Graphics System:A Specification. 4

Sen, P. & Darabi, S. (2012). On filtering the noise from the random parameters in monte carlo rendering. *ACM Transactions on Graphics*, **31**, 18:1–18:15. 109

Sheng, Y., Shi, Y., Wang, L. & Narasimhan, S.G. (2013). A practical analytic model for the radiosity of translucent scenes. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '13, 63–70, ACM. 13

Shevtsov, M., Soupikov, A. & Kapustin, E. (2007). Highly parallel fast kd-tree construction for interactive ray tracing of dynamic scenes. In *EUROGRAPHICS 2007*, Eurographics Association. 28

Shirley, P., Wade, B., Hubbard, P.M., Zareski, D., Walter, B. & Greenberg, D.P. (1995). Global illumination via density-estimation. In *Proceedings of 6th Eurographics Workshop on Rendering*, 219–230, Springer. 29

Siege1, R. & Howell, J.R. (1981). *Thermal radiation heat transfer*. hemisphere publishing corporation. 12

Siegel, M.W. & Stock, R.D. (1996). A General Near-Zone Light Source Model and its Application to Computer Automated Reflector Design. *SPIE Optical Engineering*, **35**, 2661–2679. 44

Sillion, F.X. & Puech, C. (1994). *Radiosity and Global Illumination*, vol. 11. Morgan Kaufmann Publishers. 13

Sloan, P.P., Kautz, J. & Snyder, J. (2002a). Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Transactions on Graphics*, **21**, 527–536. 34

Sloan, P.P., Kautz, J. & Snyder, J. (2002b). Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Proceedings of SIGGRAPH '02*, 527–536, ACM. 51

Spindler, M., Stellmach, S. & Dachselt, R. (2009). Paperlens: advanced magic lens interaction above the tabletop. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, 69–76, ACM. 116

Stumpfel, J., Tchou, C., Jones, A., Hawkins, T., Wenger, A. & Debevec, P. (2004). Direct HDR capture of the sun and sky. In *Proceedings of AFRIGRAPH '04*, 145–149, ACM. 2, 10, 41, 42, 49, 50, 60, 65

Sun, B. & Ramamoorthi, R. (2009). Affine double- and triple-product wavelet integrals for rendering. *ACM Transactions on Graphics*, **28**, 14:1–14:17. 35

Szirmay-Kalos, L. (2008). *Monte Carlo Methods in Global Illumination - Photo-realistic Rendering with Randomization*. VDM Verlag. 2

Talbot, J.F., Cline, D. & Egbert, P. (2005). Importance resampling for global illumination. In *Proceedings of the Sixteenth Eurographics conference on Rendering Techniques*,

Proceedings of Eurographics Symposium on Rendering'05, 139–146, Eurographics Association. 23, 69

Tatarchuk, N. (2005). Irradiance volumes for games. http://developer.amd.com/wordpress/media/2012/10/Tatarchuk_Irradiance_Volumes.pdf, accessed: 2013-10-14. 114

Tatarchuk, N. (2010). Advances in real-time rendering in 3d graphics and games i. In *ACM SIGGRAPH 2010 Courses*, SIGGRAPH '10, 4:1–4:1, ACM. 13

Tsai, Y.T. & Shih, Z.C. (2006). All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, 967–976, ACM. 34, 36

Unger, J., Gustavson, S., Larsson, P. & Ynnerman, A. (2008). Free form incident light fields. *Computer Graphics Forum*, **27**, 1293–1301. 114

Veach, E. (1998). *Robust monte carlo methods for light transport simulation*. Ph.D. thesis, Stanford University. 7, 16, 23, 24, 25, 52, 103, 122

Veach, E. & Guibas, L. (1994). Bidirectional estimators for light transport. In *Eurographics Rendering Workshop 1994 Proceedings*, 147–162. 27

Veach, E. & Guibas, L.J. (1995). Optimally combining sampling techniques for Monte Carlo rendering. In *Proceedings of ACM SIGGRAPH'95*, 419–428, ACM. 93, 94, 119

Veach, E. & Guibas, L.J. (1997). Metropolis light transport. In *Proceedings of ACM SIGGRAPH*, 65–76, ACM Press/Addison-Wesley Publishing Co. 1, 11, 27

Venkataraman, S. (2012). Programming multi-gpus for scalable rendering. 10

Verbeck, C.P. & Greenberg, D.P. (1984). A comprehensive light source description for computer graphics. *IEEE Computer Graphics & Applications*, **4**, 66–75. 2, 31, 39, 68

Wallace, J.R., Cohen, M.F. & Greenberg, D.P. (1987). A two-pass solution to the rendering equation: A synthesis of ray tracing and radiosity methods. *SIGGRAPH Comput. Graph.*, **21**, 311–320. 30

Walter, B., Alppay, G., Lafortune, E., Fernandez, S. & Greenberg, D.P. (1997a). Fitting virtual lights for non-diffuse walkthroughs. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, 45–48, ACM Press/Addison-Wesley Publishing Co. 36

Walter, B., Hubbard, P.M., Shirley, P. & Greenberg, D.P. (1997b). Global illumination using local linear density estimation. *ACM Transactions on Graphics*, **16**, 217–259. 29

Walter, B., Fernandez, S., Arbree, A., Bala, K., Donikian, M. & Greenberg, D.P. (2005a). Lightcuts: a scalable approach to illumination. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, 1098–1107, ACM. 37

Walter, B., Fernandez, S., Arbree, A., Bala, K., Donikian, M. & Greenberg, D.P. (2005b). Lightcuts: a scalable approach to illumination. *ACM Transactions on Graphics*, **24**, 1098–1107. 69, 93

Walter, B., Arbree, A., Bala, K. & Greenberg, D.P. (2006). Multidimensional lightcuts. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, 1081–1088, ACM. 37

Walter, B., Zhao, S., Holzschuch, N. & Bala, K. (2009). Single scattering in refractive media with triangle mesh boundaries. *ACM Transactions on Graphics*, **28**, 92:1–92:8. 27

Walter, B., Khungurn, P. & Bala, K. (2012). Bidirectional lightcuts. *ACM Transactions on Graphics*, **31**, 59:1–59:11. 37

Wang, R. & Akerlund, O. (2009). Bidirectional Importance Sampling for Unstructured Direct Illumination. *Computer Graphics Forum*, **28**, 269–278. 23, 69, 93

Wang, R., Ng, R., Luebke, D. & Humphreys, G. (2006a). Efficient wavelet rotation for environment map rendering. In *Proceedings of the 17th Eurographics conference on Rendering Techniques*, Proceedings of Eurographics Symposium on Rendering'06, 173–182, Eurographics Association. 33

Wang, R., Tran, J. & Luebke, D. (2006b). All-frequency relighting of glossy objects. *ACM Transactions on Graphics*, **25**, 293–318. 51

Wang, R., Wang, R., Zhou, K., Pan, M. & Bao, H. (2009). An efficient GPU-based approach for interactive global illumination. *ACM Transactions on Graphics*, **28**, 91:1–91:8. 69

Weiss, M. & Grosch, T. (2012). Stochastic progressive photon mapping for dynamic scenes. *Comp. Graph. Forum*, **31**, 719–726. 29

Yao, C., Wang, B., Chan, B., Yong, J. & Paul, J.C. (2010). Multi-Image Based Photon Tracing for Interactive Global Illumination of Dynamic Scenes. *Computer Graphics Forum*, **29**, 1315–1324. 69

Zhao, S., Hašan, M., Ramamoorthi, R. & Bala, K. (2013). Modular flux transfer: Efficient rendering of high-resolution volumes with repeated structures. *ACM Transactions on Graphics*, **32**, 131:1–131:12. 35

Zhou, K., Hou, Q., Wang, R. & Guo, B. (2008). Real-time KD-tree construction on graphics hardware. *ACM Transactions on Graphics*, **27**, 1–15. 28, 30