Efficient Global Illumination for Dynamic Scenes

Takehiro Tawara
Max-Planck-Institut für Informatik
Saarbrücken, Germany

Dissertation zur Erlangung des Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
der Naturwissenschaftlich-Technischen Fakultät I
der Universität des Saarlandes

Eingereicht am 29. Mai 2006 in Saarbrücken

Betreuender Hochschullehrer – Supervisor
Dr.-Ing. habil. Karol Myszkowski, MPI für Informatik, Saarbrücken, Germany

Gutachter – Reviewers
Dr.-Ing. habil. Karol Myszkowski, MPI für Informatik, Saarbrücken, Germany
Prof. Dr. Xavier Pueyo, Universitat de Girona, Spain
Prof. Dr. Hans-Peter Seidel, MPI für Informatik, Saarbrücken, Germany

Dekan – Dean
Prof. Dr. Thorsten Herfet, Universität des Saarlandes, Saarbrücken, Germany

Datum des Kolloquiums – Date of Defense
. . .

Takehiro Tawara
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken, Germany
`tawara@mpi-sb.mpg.de`

Abstract

The production of high quality animations which feature compelling lighting effects is computationally a very heavy task when traditional rendering approaches are used where each frame is computed separately. The fact that most of the computation must be restarted from scratch for each frame leads to unnecessary redundancy. Since temporal coherence is typically not exploited, temporal aliasing problems are also more difficult to address. Many small errors in lighting distribution cannot be perceived by human observers when they are coherent in temporal domain. However, when such a coherence is lost, the resulting animations suffer from unpleasant flickering effects.

In this thesis, we propose global illumination and rendering algorithms, which are designed specifically to combat those problems. We achieve this goal by exploiting temporal coherence in the lighting distribution between the subsequent animation frames. Our strategy relies on extending into temporal domain well-known global illumination and rendering techniques such as density estimation path tracing, photon mapping, ray tracing, and irradiance caching, which have been originally designed to handle static scenes only. Our techniques mainly focus on the computation of indirect illumination, which is the most expensive part of global illumination modelling.

Kurzfassung

Die Erstellung von hochqualitativen 3D-Animationen mit anspruchsvollen Lichteffekten ist für traditionelle Renderinganwendungen, bei denen jedes Bild separat berechnet wird, sehr aufwendig. Die Tatsache jedes Bild komplett neu zu berechnen führt zu unnötiger Redundanz. Wenn temporale Koherenz vernachlässigt wird, treten unter anderem auch schwierig zu behandelnde temporale Aliasingprobleme auf. Viele kleine Fehler in der Beleuchtungsberechnung eines Bildes können normalerweise nicht wahr genommen werden. Wenn jedoch die temporale Koherenz zwischen aufeinanderfolgenden Bildern fehlt, treten störende Flimmereffekte auf.

In dieser Arbeit stellen wir globale Beleuchtungsalgorithmen vor, die die oben genannten Probleme behandeln. Dies erreichen wir durch Ausnutzung von temporaler Koherenz zwischen aufeinanderfolgenden Einzelbildern einer Animation. Unsere Strategy baut auf die klassischen globalen Beleuchtungsalgorithmen wie ”Path tracing”, ”Photon Mapping” und ”Irradiance Caching” auf und erweitert diese in die temporale Domäne. Dabei beschränken sich unsere Methoden hauptsächlich auf die Berechnung indirekter Beleuchtung, welche den zeitintensivsten Teil der globalen Beleuchtungsberechnung darstellt.

Summary

Synthesis of images predicting the appearance of the real world has many important engineering applications including product design, architecture, and interior design. One of the major components of such predictive image synthesis is global illumination, which is very costly to compute. The reduction of those costs is an important practical problem in particular for the production of animated sequences because a vast majority of the existing global illumination algorithms were designed for rendering static scenes. In practice this means that when such algorithms are used for a dynamic scene, all computations have to be repeated from scratch even for minor changes in the scene. This leads to redundant computations which could be mostly avoided by taking into account the temporal coherence of global illumination in the sequence of animation frames. Another important problem is the temporal aliasing, which is more difficult to combat efficiently if temporal processing of global illumination is not performed. Many small errors in lighting distribution cannot be perceived by the human observer when they are coherent in the temporal domain. However, they may cause unpleasant flickering and shimmering effects when such a coherence is lost.

In this thesis, we propose global illumination and rendering algorithms, which are designed specifically to combat those problems. We achieve this goal by exploiting temporal coherence in the lighting distribution between the subsequent animation frames. Our strategy relies on extending into temporal domain well-known global illumination and rendering techniques such as density estimation path tracing, photon mapping, ray tracing, and irradiance caching, which have been originally designed to handle static scenes only. Our techniques mainly focus on the computation of indirect illumination, which is the most expensive part of global illumination modelling.

Parts of this thesis have already been published at different conferences or in journals. This thesis is based on these contributions as shown in the followings:

- As the first contribution, we present accelerated rendering of walkthrough animation sequences using a combination of ray tracing and Image-Based Rendering (IBR) techniques. A proper number of keyframes and their placement within an animation sequence are decided using the perception-based Animation Quality Metric (AQM).

- As the second contribution, we introduce a new framework for efficient rendering of global illumination in dynamic environments. We propose a combination of energy and perception based error metrics AQM to guide efficient lighting computation. In addition, we introduce spatio-temporal processing of photons at each triangle mesh element.

- As the third contribution, we propose an efficient technique for high-quality animation rendering which improves the efficiency of final gathering method and which is computationally expensive. We achieve rendering speedup by localizing in the scene space costly recomputation of the final gathering using extended photon maps and irradiance cache techniques.

- As the fourth contribution, we further extend the irradiance cache algorithm to the underlying level of the incoming radiance field. Incoming radiance directional samples, which contribute to the irradiance cache value, are stored and reused in the subsequent frames. Each incoming radiance sample is updated uniformly in space and time. The algorithm can handle a general animation which includes motion of camera, objects, and light sources.

- As the last contribution, we extend the final gathering technique to handle scenes efficiently with significant variations of lighting distribution. In such a case, commonly used uniform sampling leads to poor convergence of the irradiance integration. We propose a different sampling strategy for strong indirect illumination using an extended photon map.

Finally, we summarize our newly developed solutions for efficient rendering of global illumination in dynamic scenes and conclude this thesis with a discussion of possible directions for future work.

Zusammenfassung

Die realistische Bildsynthese, die dazu gedacht ist, das Erscheinungsbild der realen Welt zu simulieren, hat viele wichtige Einsatzbereiche im heutigen Ingenieurwesen wie zum Beispiel im Produktdesign, in Architektur und Innenarchitektur. Eine der bedeutendsten Komponenten der realistischen Bildsynthese ist die globale Beleuchtung, die jedoch sehr zeitaufwendig zu berechnen ist. Jede Reduktion dieser Zeitkosten ist von großem praktischen Nutzen insbesondere für die Erstellung von animierten Sequenzen, da die meisten existierenden globalen Beleuchtungsalgorithmen für die Berechnung statischer Szenen konzipiert sind. Für dynamische Szenen bedeutet das in der Praxis, dass alle Berechnungen selbst bei kleinsten Änderungen an der Szene für jedes Einzelbild wiederholt werden müssen. Die Folge sind redundante Berechnungen, die zum Großteil vermieden werden können, wenn die temporale Kohärenz in einer Animationssequenz ausgenutzt wird. Ein weiteres Problem in der realistischen Bildsynthese von Animationssequenzen sind temporale Aliasingeffekte, die mit viel mehr Rechenaufwand behoben werden müssen, wenn temporale Kohärenz vernachlässigt wird. Viele kleine Fehler in der Beleuchtungsverteilung können, solange sie kohärent in der temporalen Domäne sind, vom menschlichen Betrachter nicht wahrgenommen werden.

Diese Dissertation behandelt speziell diese Art von Problemen und stellt effiziente globale Beleuchtungsalgorithmen vor. Unsere Strategie liegt darin, bekannte globale Beleuchtungsalgorithmen und Techniken für statische Szenen wie zum Beispiel ”Path tracing”, ”Photon mapping”, ”Irradiance caching” in die temporale Domäne zu erweitern. Der Schwerpunkt unserer Techniken liegt dabei hauptsächlich in der indirekten Beleuchtungsberechnung, welche den größten Anteil einer globalen Beleuchtungsberechnung darstellt.

Teile dieser Dissertation wurden bereits in verschiedenen Konferenzen oder Journalen publiziert. Die Disseration basiert auf diese Publikationen die wie folgt gegliedert sind:

- Als ersten Beitrag präsentieren wir ein beschleunigtes Verfahren zum Rendern von ”Walkthrough Animationen” mittels einer Kombination von Raytracing und ”Image-Based Rendering” (IBR) Techniken.

- Als zweite Kontribution stellen wir ein neues Grundgerüst zur effizienten Bildsynthese von globaler Beleuchtung in dynamischen Umgebungen vor.

- Der dritte Beitrag zeigt einen effizienteren Ansatz für ”Final Gathering” Methoden, die zur Erstellung von hochqualitativen Animationssequenzen eingesetzt werden.

- Als vierten Beitrag erweitern wir den klassischen ”Irradiance Cache” Algorithmus zu einem ”Radiance Cache” Algorithmus. Proben der eintreffenden Strahlungsdichte (”radiance”), die zu den ”Irradiance Cache” Werten beitragen, werden separate gespeichert und in aufeinander folgenden Bildern wieder verwendet. Jede richtungsabhängige Probe wird gleichmässig in Raum und Zeit aktualisiert. Der Algorithmus kann dabei für jede Art von Animationen eingesetzt werden, die Kamerafahrten, dynamische Objekte und Lichtquellen beinhalten.

- Als letztes zeigen wir, wie die klassische ”final gathering” Technik erweitert werden kann, um Szenen mit starken Beleuchtungsvariationen effizient zu handhaben. In solchen Fällen ist die Konvergenz der Integration über die Strahlungsdichte (Berechnung der Beleuchtungsdichte) mittels klassischer uniformer Abtastmethoden (uniform sampling) sehr langsam. Wir schlagen deshalb eine andere Abtaststrategie vor, die zur Berechnung von stark variierender indirekter Beleuchtung eingesetzt werden kann und auf einer erweiterten ”Photon map” basiert.

Zum Schluß fassen wir die neu entwickelten Lösungen zum effizienten Berechnen der globalen Beleuchtung in dynamischen Szenen zusammen und schliessen diese Disseration mit einer Diskussion über mögliche Richtungen für zukünftige Arbeiten ab.

Acknowledgements

First of all, I wish to express my gratitude to my supervisor, Dr.-Ing. habil. Karol Myszkowski, Max-Planck-Institut für Informatik, Saarbrücken, for his attentive guidance and huge support. Without his continuous and patient help since I was an undergraduate student, I would have never finished this work.

Furthermore, I would like to express my appreciation to Prof. Dr. Hans-Peter Seidel, the director of the AG4 (computer graphics) group, Max-Planck-Institut für Informatik, Saarbrücken, for providing me a great opportunity to work within such an excellent research group.

I would like to thank my co-researchers Przemyslaw Rokita and Hiroyuki Akamine, who spent a lot of time to create animations with me.

I was always lucky to be surrounded with extremely creative and competent colleagues from the AG4 group. I cannot mention them all, but my warmest thanks go to people from the global illumination sub-group (in alphabetical order): Philippe Bekaert, Cyrille Damez, Kirill Alexandrovich Dmitriev, Vlastimil Havran, and Annette Scheel. Special thanks go to Kirill, who gave me his code for generating quasi-random sequences.

I have to give credit to International Max Planck Research School for Computer Science (IMPRS), which offers PhD programs and fellowships for graduate students of all nationalities, for providing me such a great opportunity. Special thanks also go to Kerstin Meyer-Ross, the IMPRS coordinator, who cared about us and planned amusing events.

I would also like to thank my external reviewer, Prof. Dr. Xavier Pueyo, Universitat de Girona, for kindly accepting to review this thesis.

I also owe thanks to Akiko Yoshida and Miloslaw Smyk for proofreading this thesis.

Contents

CHAPTER 1

Introduction

Realistic image synthesis has been a prominent research topic in computer graphics for over thirty years. Realistic images are used in many important applications such as architecture and interior design, illumination engineering, environmental assessment, special effects and film production. One of the basic conditions to achieve high level realism is the lighting modeling, which is based on physics laws governing light propagation in space and light reflection on scene surfaces. Such physically-based modeling involves extremely high computational cost because, for each point in a scene, light coming not only directly from light sources but also reflected (possibly multiple times) from surrounding surfaces must be considered. This problem is called the global illumination problem. One of the main goals of the global illumination research is to bring this complex modeling to tractable computation level for scenes with complex geometry, lighting, and reflectance properties. Additionally, the rendering quality cannot be compromised and images indistinguishable from the real world appearance should be synthesized.

The problem of computational cost becomes more pronounced in high quality animation rendering, when thousands of images must be generated to obtain a very short animation.

1.1 Interactive vs. Off-Line Rendering

Existing global illumination solutions for dynamic environments can be roughly categorized as interactive and off-line. The interactive techniques are designed to trade the image quality for the response speed in order to secure a sufficient frame rate (many applications require constant frame rate, e.g., 30 or 60 Hz for broadcasting applications). The state-of-the-art approach in the interactive global illumination is the Instant Global Illumination by using the Real Time Ray Tracing technology which recomputes each frame from scratch and requires a PC cluster to achieve interactive performance [85, 82, 81] [1]. The main objective of the interactive techniques is to provide fast response to frame-to-frame changes in the environment, but not to a sequence of such changes. The temporal coherence of lighting can be exploited much better when longer image sequences are considered. This requires the knowledge of changes in the environment for such sequences in advance. Those conditions are met for the off-line global illumination algorithms that are used in the final production of high quality animations.

1.2 Off-Line Rendering

Although the off-line computation of top quality computer animations makes it possible to include costly lighting simulation techniques, it rarely happens in industrial practice [1]. For example, in the film industry, a common approach is to use only very simple rendering algorithms, which completely ignore indirect lighting and complex light scattering functions [1]. Since scenes illuminated only by direct lighting look unrealistic, much effort is spent to place additional light sources which imitate the appearance of indirect lighting. Although this approach requires significant experience and is done manually, it is still more efficient than using fully automatic but computationally expensive existing global illumination techniques. Even in the very recent movie “Shrek 2” (produced by DreamWorks), global illumination has been used in a limited way to a single bounce of indirect lighting [69]. Another leading company of the movie production exclusively using 3D graphics, PIXAR, plans to add global illumination to their future films as well. Their current research focuses on seamless inclusion of global illumination in their RenderMan system to handle complex scenes [10, 9].

The main problem of existing global illumination solutions is poor scaling of the computation load with increasing scene complexity. It is often caused by wasting computational efforts on unimportant scene details that cannot be perceived in the final animation by human observers [58, 26]. Additionally, existing animation rendering techniques process every single frame independently, therefore they cannot account properly for the eye sensitivity variations resulting from temporal considerations [49]. While some efficient techniques of perception-based guidance of the global illumination computation for static images have been proposed recently [20, 4, 56], relatively little attention has been paid to the temporal aspect of such a guidance for dynamically changing environments. In particular, indirect lighting often changes slowly from frame to frame and many of those changes might be too subtle to be perceived (changes in direct lighting are usually quite well noticeable due to high-contrast shadows and highlights accompanying

[1] for more complete surveys of interactive global illumination techniques, refer to [15, 84, 83].

the moving objects). Since it is an obvious waste to repeat the computation of indirect lighting for every frame, the performance of animation rendering could be significantly improved by exploiting the temporal coherence of indirect lighting. Ideally, lighting information acquired for the preceding and following frames should be reused as much as possible for reconstructing indirect lighting in a given frame; however, the quality of the resulting animation as perceived by human observers should not be compromised.

1.3 Problem Statement

The production of high quality animations featuring compelling lighting effects is very time consuming using traditional rendering approaches in which each frame is computed separately. It leads to unnecessary, redundant computation because a vast majority of computation must be started from scratch for each frame. Since temporal coherence is typically not exploited, temporal aliasing problems are also more difficult to combat. Many small errors in lighting distribution cannot be perceived by human observers when they are coherent in temporal domain. However, when such a coherence is lost, the result often becomes unpleasant flickering effects.

In this thesis, we propose global illumination and rendering algorithms, which are designed specifically to combat those problems. We achieve this goal by exploiting temporal coherence in the lighting distribution between the subsequent animation frames. Our strategy relies on extending into temporal domain well-known global illumination and rendering techniques such as density estimation path tracing, photon mapping, ray tracing, and irradiance caching. All those techniques have been originally designed to handle static scenes only. Our techniques mainly focus on the computation of indirect illumination which is the most computationally expensive part of global illumination.

1.4 Chapter Overview

Parts of this thesis have already been published at different conferences or in journals [50, 51, 52, 71, 70, 73, 72]. This thesis is based on these contributions and organized as follows:

- In Chapter 1 we introduce the general idea of global illumination and discuss briefly its major applications. We identify some problems with existing algorithms and summarize our solutions designed to combat those problems.

- In Chapter 2 we explain basic terminology used in global illumination. Then we introduce the rendering equation which is the core of all global illumination algorithms. We also present briefly a number of relevant works of global illumination and rendering algorithms.

- In Chapter 3 we discuss previous work related to this thesis. We focus mainly on off-line global illumination and rendering solutions which lead to high quality animations.

- In Chapter 4 we present accelerated rendering of walkthrough animation sequences using a combination of ray tracing and Image-Based Rendering (IBR) techniques. A proper number of keyframes and their placement within an animation sequence are decided using the perception-based Animation Quality Metric (AQM) [50].

- In Chapter 5 we introduce a new framework for efficient rendering of global illumination in dynamic environments. We propose a combination of energy and perception based error metrics AQM to guide efficient lighting computation. In addition, we introduce spatio-temporal processing of photons at each triangle mesh element [51, 52].

- In Chapter 6 we propose an efficient technique for high-quality animation rendering which improves the efficiency of final gathering method and which is computationally expensive. We achieve rendering speedup by localizing in the scene space costly recomputation of the final gathering using extended photon maps and irradiance cache techniques [71].

- In Chapter 7 we further extend the irradiance cache algorithm to the underlying level of the incoming radiance field. Incoming radiance directional samples, which contribute to the irradiance cache value, are stored and reused in the subsequent frames. Each incoming radiance sample is updated uniformly in space and time. The algorithm can handle a general animation which includes motion of camera, objects, and light sources [73].

- In Chapter 8 we extend the final gathering technique to handle scenes efficiently with significant variations of lighting distribution. In such a case, commonly used uniform sampling leads to poor convergence of the irradiance integration. We propose a different sampling strategy for strong indirect illumination using an extended photon map [72].

- In Chapter 9 we summarize our newly developed solutions for efficient rendering of global illumination in dynamic scenes and conclude this thesis with a discussion of possible directions for future work.

CHAPTER 2

Background

This chapter introduces the physical and mathematical fundamentals behind global illumination algorithms. For more complete explanations, refer to well established textbooks [12, 65, 25, 33, 19, 54]. Table 2.1 summarizes symbols used in this chapter.

2.1 Lighting Terminology

The goal of global illumination algorithms is to simulate all possible paths of lighting transport in a scene. The transport is formulated using physical quantities introduced in radiometry, which is a scientific discipline dealing with the physical measurement of light. In this section, we briefly explain the basic radiometric terminology used in global illumination algorithms.

Flux
Flux (Φ) is the radiant power in *Watt (W)*. It is the total energy leaving from / arriving at a surface per unit time (J/sec).

Irradiance and Radiosity
Irradiance (E) is the incident flux per unit surface area (W/m^2) at a surface location x:

$$E(x) = \frac{d\Phi}{dA}. \tag{2.1}$$

Radiosity (B) is the outgoing flux per unit surface area (W/m^2) at a surface location x:

$$B(x) = \frac{d\Phi}{dA}. \tag{2.2}$$

Symbol	Description	Unit
x	Position	
$\vec{n}$	Normal	
$\vec{\omega}_i$	Direction of incoming radiance (away from surface)	
$\vec{\omega}_o$	Direction of outgoing radiance (away from surface)	
$d\vec{\omega}$	Differential solid angle	
Ω	Hemisphere of directions	
Φ	Flux	W
E	Irradiance	W/m^2
B	Radiosity	W/m^2
L	Radiance	$W/m^2/sr$
L_e	Emitted radiance	$W/m^2/sr$
L_i	Incoming radiance	$W/m^2/sr$
L_o	Outgoing radiance	$W/m^2/sr$
f_r	BRDF	
ρ	reflectance	
η	Index of refraction	

Table 2.1: Symbols used in this chapter.

Radiance
Radiance (L) is flux per unit projected area per unit solid angle ($W/m^2/sr$):

$$L(x, \vec{\omega}) = \frac{d^2\Phi}{\cos\theta dA d\vec{\omega}} \tag{2.3}$$

where $d\vec{\omega}$ denotes the differential solid angle in the direction $\vec{\omega}$, and θ is the angle between the direction $\vec{\omega}$ and the surface's normal. Radiance is the most important quantity in the image synthesis because it does not change when light is traveling in space (it is assumed that there is no participating medium in space). Additionally, the human eye is directly sensitive to radiance.

We can compute irradiance at a location x, by integrating incoming radiance over all directions Ω if incoming radiance field is known:

$$E(x) = \int_{\Omega} L_i(x, \vec{\omega}_i)(\vec{\omega}_i \cdot \vec{n}) d\vec{\omega}_i \tag{2.4}$$

where $(\vec{\omega}_i \cdot \vec{n}) = \cos\theta$.

2.2 Luminaires

Light is emitted from a light source which might be artificial (an incandescent bulb or a fluorescent lamp) or natural (the sun or fire). In the global illumination algorithms, the intensity of a light source is given in flux. The following light source types are considered within the scope of this thesis:

Point Light
A point light is an idealized light source whose size is infinitely small. Despite of this unrealistic assumption, it is often used in computer graphics to simplify the lighting computation. The power of light is usually uniformly distributed in all spherical directions, but goniometric diagrams can be used to specify more complex angular power distributions [78].

Area Light
An area light source is used when the spatial dimensionality of luminaires cannot be neglected. The luminaire can be of arbitrary shape, but a simple geometry such as a square and disk is commonly preferred because sampling algorithms are also simple. Note that in contrast to point light sources, the probability of sampling directions must be proportional to the cosine of the outgoing angle for area light sources. More details on light source sampling are given in Section 2.6.

Directional Light
A directional light has a fixed direction and is often used to emulate the sunlight. Because the distance to the sun is very large, all directions of the emitted rays from a directional light source can be assumed parallel to each other.

2.3 Bidirectional Reflectance Distribution Function (BRDF)

The Bidirectional Reflectance Distribution Function (BRDF) describes how light reflects on a surface. It is defined as the ratio of reflected radiance and irradiance:

$$f_r(x, \vec{\omega}_i, \vec{\omega}_o) = \frac{dL_r(x, \vec{\omega}_o)}{dE_i(x, \vec{\omega}_i)} = \frac{dL_r(x, \vec{\omega}_o)}{L_i(x, \vec{\omega}_i)(\vec{\omega}_i \cdot \vec{n})d\vec{\omega}_i} \tag{2.5}$$

where $\vec{n}$ is the normal at a point x. Using the BRDF and incoming radiance, reflected radiance, L, is computed as:

$$\begin{aligned} L(x, \vec{\omega}_o) &= \int_\Omega f_r(x, \vec{\omega}_i, \vec{\omega}_o) dE(x, \vec{\omega}_i) && (2.6) \\ &= \int_\Omega f_r(x, \vec{\omega}_i, \vec{\omega}_o) L_i(x, \vec{\omega}_i)(\vec{\omega}_i \cdot \vec{n}) d\vec{\omega}_i. && (2.7) \end{aligned}$$

The physically correct BRDF must satisfy the following two properties:

Helmholtz Reciprocity
This law states that the BRDF is independent from the direction of light flows, i.e., the incident and outgoing directions can be exchanged in the BRDF:

$$f_r(x, \vec{\omega}_i, \vec{\omega}_o) = f_r(x, \vec{\omega}_o, \vec{\omega}_i). \tag{2.8}$$

All two-pass global illumination algorithms utilize this property by tracing light paths both from the light sources and the eye.

Energy conservation
This property ensures that a surface cannot reflect more energy than it received:

$$\int_\Omega f_r(x, \vec{\omega}_i, \vec{\omega}_o)(\vec{\omega}_i \cdot \vec{n}) d\vec{\omega}_i < 1, \forall \vec{\omega}_o. \tag{2.9}$$

2.3.1 Idealized Light Reflection Models

Although the BRDF is a generalized reflection function and can represent any type of distribution functions, it is often decomposed into idealized reflection models to simplify the computation.

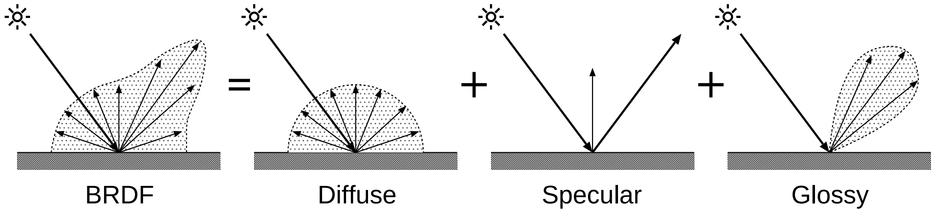


Figure 2.1: The BRDF can be represented as a sum of different reflection types.

Diffuse Reflection
When light arrives at a diffuse surface, it is reflected in all directions. It is caused by multiple scattering of light in the microstructure of a surface. A special case of diffuse reflection is Lambertian, when light is uniformly reflected over all hemispherical directions:

$$f_r(x, \vec{\omega}_i, \vec{\omega}_o) = \rho_d \frac{1}{\pi} \tag{2.10}$$

where ρ_d is the diffuse reflectance.

Specular Reflection
On a mirror-like surface, the incident and reflected light lie in the same plane along

with the surface's normal, and the incident light angle measured in respect to the normal vector is equal to the outgoing light direction. The reflected direction $\vec{\omega}_o$ can be expressed using the incident light direction $\vec{\omega}_i$ and the normal vector $\vec{n}$:

$$\vec{\omega}_o = 2(\vec{\omega}_i \cdot \vec{n})\vec{n} - \vec{\omega}_i. \tag{2.11}$$

Refraction

A transparent surface such as water and glass refracts light at the interface with the air (or another media). The incident angle, θ_1, and the refracted angle, θ_2, measured with respect to the normal are defined by *Snell's law*:

$$\eta_1 \sin\theta_1 = \eta_2 \sin\theta_2 \tag{2.12}$$

where η_1 and η_2 are the corresponding indices of refraction for the media at both sides of the interface surface. The refracted direction $\vec{\omega}_o$ can be computed as

$$\vec{\omega}_o = -\frac{\eta_1}{\eta_2}(\vec{\omega}_i - (\vec{\omega}_i \cdot \vec{n})\vec{n}) - \left(\sqrt{1 - \left(\frac{\eta_1}{\eta_2}\right)^2 (1 - (\vec{\omega}_i \cdot \vec{n})^2)}\right)\vec{n}. \tag{2.13}$$

Glossy Reflection

Most of surfaces have glossiness which can be determined between perfect diffuse and specular reflection. Several functions have been proposed to capture realistic glossiness using physically based or empirical approaches. A simple reflectance function based on the Phong model is shown [39] as

$$f_r(x, \vec{\omega}_i, \vec{\omega}_o) = \rho_s \frac{k+2}{2\pi} \cos^k\theta \tag{2.14}$$

$$\cos\theta = R(\vec{\omega}_i) \cdot \vec{\omega}_o, \cos\theta > 0 \tag{2.15}$$

where ρ_s is the glossy reflectance and k is the exponential coefficient. $R(\vec{\omega}_i)$ is the mirror reflection direction of $\vec{\omega}_i$. The function exhibits all properties required by the physically correct BRDF as discussed in Section 2.3.

2.4 Rendering Equation

Kajiya introduced the rendering equation [35] which is the fundamental formula for all global illumination algorithms. In other words, each global illumination algorithm is designed to compute the rendering equation efficiently and accurately.

The rendering equation represents outgoing radiance, L_o, at any location and is formed as the sum of emitted radiance, L_e, and reflected radiance computed as the integral of incoming radiance convolved with the surface BRDF $f_r(x, \vec{\omega}_i, \vec{\omega}_o)$:

$$L_o(x, \vec{\omega}_o) = L_e(x, \vec{\omega}_o) + \int_\Omega f_r(x, \vec{\omega}_i, \vec{\omega}_o) L_i(x, \vec{\omega}_i)(\vec{\omega}_i \cdot \vec{n}) d\vec{\omega}_i. \tag{2.16}$$

Note that $L_i(x, \vec{\omega}_i)$ depends on L_o for other surfaces. It effectively means that those radiances are coupled and unknown.

2.5 Monte Carlo Integration

In this section, we briefly discuss Monte Carlo integration techniques which are commonly used to solve the rendering equation. The advantage of Monte Carlo integration is that it does not impose any constraints on the integrand in terms of its continuity or smoothness. It is only required that the integrand can be evaluated at a given point. Monte Carlo integration easily extends to high dimensional problems without introducing exponential growth of the computation [76]. In the following parts of this section, we shortly review the terminology of probability theory and introduce a basic Monte Carlo estimator.

Discrete Random Variables
First, let us consider a random variable which can take a finite number of possible values. For a discrete random variable with N possible outcomes, the expected value (or mean) of a random variable can be estimated as

$$E(x) = \sum_{i=1}^{N} p_i x_i \tag{2.17}$$

where p_i is a probability associated with any event with outcome x_i.

Continuous Random Variables
Now we extend our discussion to include continuous random variables. The probability P that a variable x takes a value in the range $[a, b]$ is defined as follows:

$$P(x \in [a, b]) = \int_a^b p(x) dx \tag{2.18}$$

where $p(x)$ is a *probability density function* (PDF) such that

$$\int_{-\infty}^{\infty} p(x) dx = 1, \forall x : p(x) \geq 0. \tag{2.19}$$

Similar to the case of discrete random variables, the expected value of a given function $f(x)$ can be computed as

$$E[f(x)] = \int f(x)p(x)dx. \tag{2.20}$$

We can also calculate the expected value of a function by taking the mean of a large number of random samples from the function. It will converge toward the correct answer as the number of samples approaches infinity (Law of Large Numbers):

$$E[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i). \tag{2.21}$$

Monte Carlo Integration

Now we would like to compute an estimate of the integral of a function $f(x)$. From Equations 2.20 and 2.21, the following equation can be derived:

$$\int f(x)dx = \int \frac{f(x)}{p(x)}p(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}. \tag{2.22}$$

The rightmost term in Equation 2.22 is called the Monte Carlo estimator for an integral $\int f(x)dx$. The accuracy of the estimator depends on the number of samples N, and the error is proportional to $1/\sqrt{N}$.

2.6 Sampling Random Variables

In Monte Carlo integration techniques [36, 19], sampling random variables is an important problem. Random numbers are used to decide the location in a given pixel of a rendered image, a position inside an area light source, a direction of a reflected and transmitted ray, the terminating condition of a ray path, and so on.

2.6.1 Explicit Sampling

We often need to sample directions in spherical coordinates (θ, ϕ). Two uniform random numbers $\xi_1 \in [0, 1]$ and $\xi_2 \in [0, 1]$ can be mapped to spherical coordinates keeping the property of the uniform distribution for some functions. In terms of coordinates (x, y, z), the direction becomes

$$\vec{\omega} = (x, y, z) = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta). \tag{2.23}$$

Spherical Sampling
A point light source distributes the fraction of the power into all possible directions from its position. A sampling direction is given as

$$(\theta, \phi) = (\cos^{-1}(1 - 2\xi_1), 2\pi\xi_2). \tag{2.24}$$

Diffuse Sampling
For sampling reflected directions on a diffuse surface and emitted directions on an area light source with the diffuse light distribution, the uniform sampling should be weighted by the cosine function of the sample direction in respect to the surface normal:

$$(\theta, \phi) = (\sin^{-1}(\sqrt{\xi_1}), 2\pi\xi_2). \tag{2.25}$$

Phong Specular Sampling
The modified Phong reflectance function shown in Equation 2.14 can also be explicitly sampled as

$$(\theta, \phi) = (\cos^{-1}(\xi_1{}^{\frac{1}{n+1}}), 2\pi\xi_2). \tag{2.26}$$

This mapping originally distributes points around the z-axis, so we need to rotate the sampling direction to the actual mirror direction with respect to the incoming direction.

2.6.2 Rejection Sampling

Explicit sampling requires an analytical formula for the inverse of the cumulative distribution function which is often not possible analytically. In such a case, rejection sampling technique can be used. In this technique samples are repeatedly generated and discarded until a specified property is obtained. In case the ratio of sample acceptance is low, this sampling technique may be inefficient. On the other hand, this technique is very easy to implement and works for any probability distribution function.

2.7 Variance Reduction Techniques

The main problem with Monte Carlo integration is the slow convergence rate $1/\sqrt{N}$. It means that in order to halve the error, we must use four times as many samples. To improve the quality of an estimate, several variance reduction techniques are proposed, which we summarize briefly in this section.

Importance Sampling

The main idea of importance sampling is to concentrate more samples in important regions of a function than in remaining regions. The variance can be arbitrarily low by choosing a good PDF used to distribute samples in the function domain. It can be shown that the optimal PDF is $f(x)/\int f(x)dx$. However, its denominator requires the knowledge of the integral value, which is the goal of the computation. In practice, we select PDFs using the knowledge of the BRDF of a surface, the intensity of light sources, and so on.

Stratified Sampling

Stratified sampling is another powerful variance reduction technique. In this technique the domain of the integral is divided into subdomains. The main goal of this domain subdivision is to reduce the integrand variability in each subdomain. Stratified sampling leads to a significant variance reduction, and its performance strongly depends on the choice of subdomains (strata). Even for a very poor strata selection, the variance of stratified sampling cannot be worse than a naive Monte Carlo sampling. It is more efficient to increase the number of strata than to use more samples per stratum. For this reason, only one sample per stratum is usually used. The problem with stratified sampling is that the number of subdomains must be known in advance.

In the following sections, we discuss selected global illumination algorithms based on Monte Carlo methods which lead to the solution of the rendering equation.

2.8 Monte Carlo Ray Tracing

Monte Carlo ray tracing techniques are the extended algorithms of a classic ray tracing [96] to capture global illumination effects. The basic idea is a point sampling by tracing random rays in a scene and the weighted sum of samples to compute the integral of the rendering equation. The advantages of this idea are an easy implementation, memory efficiency, and easy handling of complex geometries and materials. The main disadvantages are the huge computational cost and stochastic noise. In the past 20 years, several algorithms have been proposed as described below.

Distribution Ray Tracing
Distribution ray tracing was introduced by Cook et al. [13] in 1984. It extended traditional ray tracing to capture effects such as soft shadows, motion blur, and depth of field by stochastic sampling in each of those domains.

Path Tracing
Kajiya introduced the path tracing [35] in 1986 as a solution of the rendering equation. The algorithm can simulate all possible light paths in a scene and capture full global illumination effects. In the path tracing, a random ray is shot from the eye and traced a reflected and/or refracted ray until the ray hits a light source or is absorbed on a surface. A number of such random paths through a pixel contributes the power to the outgoing radiance for a given pixel. The main problem of the path tracing is that the solution variance is high. It is seen as the high frequency noise in a rendered image. If a light source is very small compared to a scene, the probability that a path hits the light source becomes very small. The next event estimation [19] helps to overcome this problem by explicit sampling of all light sources in a scene.

Bidirectional Path Tracing
To improve the performance of path tracing, the bidirectional path tracing was proposed by Lafortune and Willems [38] and Veach and Guibas [77]. The idea of the bidirectional path tracing is to trace paths from both the eye and the light sources and connect nodes of each path with proper weights to compute their contribution to a given pixel. The algorithm works significantly better than path tracing for a certain type of paths in case that it is difficult to reach light sources from the eye path. However, it still suffers from high frequency noise, which can be reduced below the perceptibility level by shooting a huge number of rays for each pixel.

2.9 Photon Mapping

The photon mapping was introduced by Jensen [32, 33] in 1996. It is based on Monte Carlo ray tracing algorithm and can efficiently render an image with global illumination in complex models with arbitrary BRDFs. The most important feature of this algorithm is decoupling the illumination representation from geometry. The algorithm stores lighting information in a point-based data structure, *the photon map*. This point-based data structure enables to handle very complex geometry.

The method consists of two passes: (1) building the photon map using photon tracing and (2) rendering an image using the cached information in the photon map. We describe the details of those two passes in the following two sections.

2.9.1 Photon Tracing

In the first stage of the photon mapping method, photons are traced from light sources toward a scene, and the photon-hit points on diffuse surfaces are registered in the photon map. To accelerate the search in the photon map and reduce the memory storage, the kd-tree data structure is used and photons are stored in a heap-like memory layout. This stage is very fast comparing to the second pass.

To improve the rendering efficiency, two photon maps are built: *caustic* and *global* photon maps. The caustic photon map collects photons immediately after they are reflected or refracted by surfaces with specular light reflectance properties. The global photon map stores all photons hitting diffuse surfaces. Since this map is not directly rendered but rather queried in the final gathering procedure, a moderate number of photons is sufficient.

2.9.2 Rendering

In the rendering stage, caustic effects are reconstructed through a direct density estimation performed for photons in the caustic photon map. This enables the reconstruction of quick changes for spatial lighting patterns which are typical for caustic effects. To obtain high quality caustics (to reduce the stochastic noise without excessive blurring of caustic details), a huge number of photons are needed. This can be achieved by reinforcing shooting caustic photons in the direction of specular objects in a scene. This is an easy task for photons propagating directly from light sources to specular surfaces. Recently, it has been shown that selective photon tracing [17] can be used to take into account more complex photon paths involving collisions with diffuse or glossy surfaces before reaching specular surfaces [27].

2.10 Irradiance Caching

Soft indirect lighting is reconstructed from the global photon map through the irradiance cache technique [91, 90, 33]. For each cache location, irradiance is integrated over a scene by sampling the incoming energy for selected directions as illustrated in Figure 2.2. Each sample involves a heavy intersection computation performed by tracing a ray and the estimation of energy incoming from the point

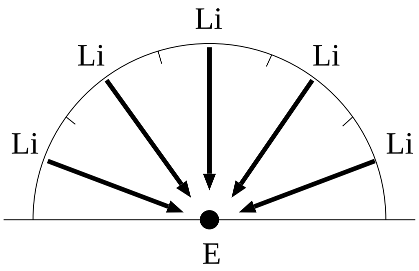


Figure 2.2: Irradiance E and incoming radiance samples L_i (one sample per strata) at an irradiance cache location marked by the black dot.

hit by a ray. The photon map is used for the incoming lighting reconstruction using the nearest neighbor density estimation method [29]. To reduce the density estimation cost, Christensen [8] proposed to precompute irradiance and store the resulting values at sparsely selected photon locations on diffuse surfaces. When the final gathered ray hits an object, the precomputed irradiance value for the nearest photon location is used. To reduce the variance of such sampling, the hemisphere of all possible directions is split into strata, and a small number of sample directions (usually one) are randomly chosen for each stratum (refer to Figure 2.2).

This stratified sampling works well for scenes with low variation of lighting distribution. However, it leads to an enormous number of samples when density of photons in the global photon map significantly changes between scene regions. Ideally, the angular density of samples should correspond to the density of photons stored in the map.

CHAPTER 3

Related Work

The problem of global illumination for dynamically changing environments has attracted significant attention in the research community, and a number of solutions that attempt to exploit temporal coherence in lighting have been proposed. In this survey of previous work, we limit our discussion to off-line methods whose emphasis is high animation quality. In addition, we assume that all animation paths are known in advance.

In the following sections, we describe solutions tailored for a keyframe-based approach to an animation. We also present extensions of radiosity, bi-directional path tracing, and final gathering rendering to handle image sequences. At the end of this chapter, we discuss more recent attempts of exploiting the characteristics of human visual system to improve the performance of animation rendering.

3.1 Keyframe-Based Animation Rendering

Keyframe-based animation rendering relies on exact computation for a fixed number of frames (*keyframes*) and reusing the results obtained for the remaining frames (*inbetween frames*). This idea has been also used in the context of lighting computation which are performed only for keyframes and interpolated for inbetween frames [53, 97]. Usually the number of inbetween frames between a pair of keyframes is the same for the whole animation, and there is no control of the validity of applying the keyframe lighting to the inbetween frames [97]. This approach can result in visually noticeable errors in the lighting distribution, which is affected by changes in the environment that occur in the course of an animation.

Obviously, the errors in lighting that are explicitly caused by the scripted animation of light sources can be compensated by increasing the number of keyframes for the affected animation segments. However, the question arises how many additional keyframes must be placed, so that approximations in the lighting reconstructed for inbetween frames remain unnoticeable. It is even more difficult to predict how the moving objects will affect the lighting distribution based merely on the animation script.

A significant step toward adaptive keyframe selection have been done by Nimeroff et al. [53]. They proposed a powerful range-image-based framework for handling indirect lighting in dynamic environments. The indirect lighting is sparsely sampled in time and then interpolated to reconstruct full global illumination for selected base images. The time steps for recomputing the indirect lighting are found by recursive subdivision. At each time step, the lighting is calculated for a number of vertices using deterministic wavelet radiosity [12, 65], then the differences between the corresponding vertices are computed. If larger differences than an assumed threshold are found for a certain percentage of vertices the time sequence is subdivided. The drawback of this approach is that direct lighting is not considered, therefore it could effectively wash out even significant differences in indirect lighting [23]. Additionally, the tone reproduction [75] is not applied to the resulting lighting. Applying tone reproductions is difficult in the view-independent framework as proposed by Nimeroff et al. [53] because the eye adaptation conditions cannot be established.

The interpolation of indirect lighting between two time steps is an important feature of Nimeroff's framework. The continuity of changes in the lighting distribution between time steps is modelled, and popping effects resulting from switching between two distinct lighting distributions as in [97] can be avoided. However, in all discussed approaches, the accuracy of indirect lighting reconstruction fluctuates between frames, achieves the highest level for the keyframes, and then gradually decreases for the remaining frames usually as a function of their distance to the keyframes along an animation path.

Clearly, even for the simple approach with reusing lighting for inbetween frames, some error metrics are needed to guide the keyframe placements. Some perception-based animation quality metrics are required to enable direct judgement whether the errors introduced by exploiting the temporal coherence are below the sensitivity level of human observers. In addition, by performing some limited computation for all frames (not just keyframes), abrupt changes in lighting can be identified more easily. We propose an algorithm designed along those guidelines in Chapter 4.

3.2 Spatio-Temporal Radiosity Solutions

Deterministic and Monte Carlo radiosity techniques are commonly used in many commercial packages and research rendering systems. A common denominator for those technique is that mesh is used for lighting computation and reconstruction. The extension of radiosity techniques to handle dynamic environments proved to be quite successful. Early solutions [6, 22, 48] were embedded into the progressive radiosity framework and relied on shooting the corrective energy (possibly negative) to scene regions affected by the environment changes. Much better performance was obtained by more recently introduced techniques that are based on hierarchical radiosity [18, 61, 16, 55]. A line-space hierarchy proposed by Drettakis and Sillion [18] enables fast identification of links affected by a scene modification and leads to image updates at interactive rates for moderately complex environments. However, the memory requirements inherent in this technique are extremely high because, apart from storing the active links used for energy gathering, also passive (refined) links and shafts for the entire scene are stored. The problem of storing shafts was recently reduced by Schoeffel and Pomi [61]. They store shafts only locally for regions affected by geometry changes. Damez and Sillion [16] explicitly incorporated time in the hierarchical radiosity framework and showed substantial improvements in the rendering performance of animated sequences. However, it was achieved at the expense of a significant increase of memory requirements which become impractical for complex scenes. Pueyo et al. [55] proposed a radiosity algorithm which focuses on exploiting the temporal coherence of subsequent animation frames for static camera parameters. All discussed radiosity techniques work well only when the lighting changes are well localized in a scene. Otherwise, recomputing the lighting from scratch is a better choice.

In Global Monte Carlo Radiosity [2], the temporal coherence of costly visibility computations is efficiently and conservatively exploited. However, the radiosity solution is performed independently for each frame, and all radiosity solutions are stored simultaneously in the memory. The algorithm generates "*global lines*", which are cast independently from surface positions, with a uniform density all over the scene. They can be generated, e.g., by joining random pairs of points taken in a sphere bounding the whole scene. The temporal coherence is exploited by reusing the visibility information between static surfaces stored in global lines.

The high quality of the lighting reconstruction is not guaranteed in those mesh-based object space approaches. In particular, changes of direct lighting effects such as shadows and highlights resulting from object motion are usually highly visible because they are important visual cues (in general, specular effects are difficult to model for radiosity techniques). To avoid similar problems, a majority of high quality rendering solutions computes direct lighting for every frame using view-dependent techniques such as sophisticated scanline methods [1] or ray tracing [89, 32].

In Chapter 5, we propose our mesh-based solution which is based on Monte Carlo light tracing. It avoids many of discussed problems.

3.3 Spatio-Temporal Bi-Directional Path Tracing

In the previous section, we discussed radiosity algorithms in which spatio-temporal processing was performed in the object space. In this section, we consider a view-dependent algorithm called *bi-directional path tracing* (BPT) [38, 77] which was extended by Havran et al. to handle dynamic environments [28]. In this algorithm, the bookkeeping of global illumination samples is organized in the image space. The BPT algorithm is considered within a more general rendering framework for computing multiple frames at once by exploiting the coherence between image samples (pixels) in the temporal domain. For each sample representing a given point in the scene, its view-dependent components are updated for each frame and their contribution is added to pixels identified through the compensation of camera and object motion.

The global illumination computation in the framework proposed by Havran et al. is based on the BPT algorithm [38, 77] and uses the *multi-frame visibility data structure* (MFVDS) to query visibility for all considered frames at once. Each bi-directional estimate of a given pixel color is reused for several frames before and after the one it was originally computed for. To reuse these estimates, the BRDF values at the first hit point of the eye path need to be recomputed to take into account the new viewpoint. The corresponding estimates are then added to the pixel through which the hit point can be seen for the considered frame. Since it involves only the evaluation of direct visibility from the viewpoint and a few BRDF recomputations, reusing a sample is much faster than recomputing from scratch. Reusing samples for several frames also makes the noise inherent to stochastic methods fixed in an object space and enhances the quality of the resulting animations.

The main advantage of this framework is a significant speedup of animation rendering, which is usually over one order of magnitude in respect to traditional frame-by-frame rendering, while the obtained quality is always much higher due to a significant reduction of flickering. Many standard tasks in rendering such as shading, texturing, and motion-blur can be efficiently performed in this rendering architecture.

3.4 Spatio-Temporal Final Gathering

In the rendering of production quality animation, global illumination computations are usually performed using two-pass methods [69, 9]. In the first (preprocessing) pass, the lighting distribution over scene surfaces is sparsely computed using radiosity [42, 67, 11] or photon mapping [33, 8] methods. In the second (rendering) pass, more exact global illumination computation is performed on a per-pixel basis using the results obtained in the first pass. To improve the spatial resolution of lighting details for a given camera view, the *final gathering* [57, 42, 67, 11] is commonly used. Usually the direct lighting is explicitly computed for each pixel, and the indirect lighting is obtained through the integration of incoming radiances, which is computationally expensive.

The final gathering has been initially introduced in the context of the radiosity algorithm to overcome the problems arising with the mesh-based storage of lighting [57, 42, 67, 11]. More efficient versions of this final gathering have been recently proposed specifically for Hierarchical Radiosity with Clustering [59, 60].

The final gathering costs also can be reduced by using the *irradiance cache* data structure [91, 90] which is more suitable for ray tracing methods. Within this method, irradiance samples are lazily computed and sparsely cached in an object space for a given camera position (a view-dependent process). The indirect illumination is interpolated for each pixel based on those cached irradiance values. It is significantly faster than the final gathering computation for each pixel. The irradiance cache technique efficiently removes shading artifacts which are very difficult to avoid if the indirect lighting is directly reconstructed based on the radiosity mesh or the photon maps. However, this high quality lighting reconstruction needs long computation time mostly because of the irradiance integration that is performed for each cache location in a scene. All those final gathering approaches can be used for a walkthrough animation, however, they are not suitable for the rendering of dynamic environments.

Recently, Martin et al. [45] proposed a final gathering algorithm in the framework of space-time hierarchical radiosity. Martin et al. classified the hierarchical radiosity links into the *good links* if the error of gathered lighting is within given bounds and the *bad links* otherwise. For the good links, the final gathering step is not required, and resulting lighting is accumulated in a texture using linear interpolation within a given patch. For each shooter polygon associated with a bad link, the graphics hardware is used to estimate the visibility of receiver patch texels using the projective shadow technique. In the temporal domain, the bad links are classified into *static* and *dynamic*. The costly visibility computation is performed once for a given time interval for static links and repeated for each frame for dynamic links.

The final gathering method proposed by Martin et al. leads to significant rendering speedup (1.7–7 times in the examples given by the authors). However, the method shares typical drawbacks of hierarchical radiosity solutions such as poor handling of non-Lambertian surfaces and significant storage costs required by the link data structures. Those costs are even more significant in the presented solution because the history of links is also stored, i.e., links are not deleted when refined for possible reuse in different time intervals.

In Chapter 6, we propose our final gathering approach which exploits the temporal coherence and is embedded into the photon mapping rendering algorithm.

3.5 Perception-guided Animation Rendering

The main goal of perception-guided animation rendering techniques is to save computational cost without compromising the resulting animation quality as perceived by human observers. Recently, some successful examples of perception-based rendering of static images have been presented [20, 4, 56]. However, expanding those techniques to handle the temporal domain remains mostly an open problem.

Yee [97] proposed an interesting application of a visual attention model to improve the efficiency of indirect lighting computations in the RADIANCE system [89] for dynamic environments. Yee demonstrated that greater errors can be tolerated for less salient image regions in which the density of indirect lighting samples can be substantially reduced. However, variability in the selection of the region of interests (ROI) for different observers or even for the same observer from session to session can lead to some degradation of the animation quality in regions that were not considered as important attractors of the visual attention.Yee reported that such degradations of quality could be perceived when the same animation sequence was viewed more than once by the same observer.

CHAPTER 4

Exploiting Temporal Coherence in Walkthrough Rendering

In this chapter, we consider accelerated rendering of walkthrough animation sequences using a combination of ray tracing and Image-Based Rendering (IBR) techniques. Our goal is to derive as many pixels as possible using inexpensive IBR techniques without affecting the animation quality. A perception-based spatio-temporal Animation Quality Metric (AQM) is used to automatically guide such a hybrid rendering. The Image Flow (IF) [30] obtained as a by-product of the IBR computation is an integral part of the AQM. The final animation quality is enhanced by an efficient spatio-temporal antialiasing, which utilizes the PF to perform a motion-compensated filtering.

4.1 Introduction

The central part of our approach is the Image Flow (IF) which is computed as a displacement vector field in the image plane due to the motion of the camera along an animation path (refer to Figure 4.1). The displacement distribution is provided for all pixels and all frames in an animation sequence. The IF is computed using IBR techniques, which guarantees very good accuracy and high speed processing for synthetic images. The IF is used in our technique in the following tasks:

- To perform the temporal considerations of our perception-based animation quality metric (Section 4.2).
- To reproject pixels from the ray-traced keyframes to the image-based inbetween frames (Section 4.3).
- To enhance the animation quality by performing antialiasing based on motion-compensated filtering (Section 4.3.3).

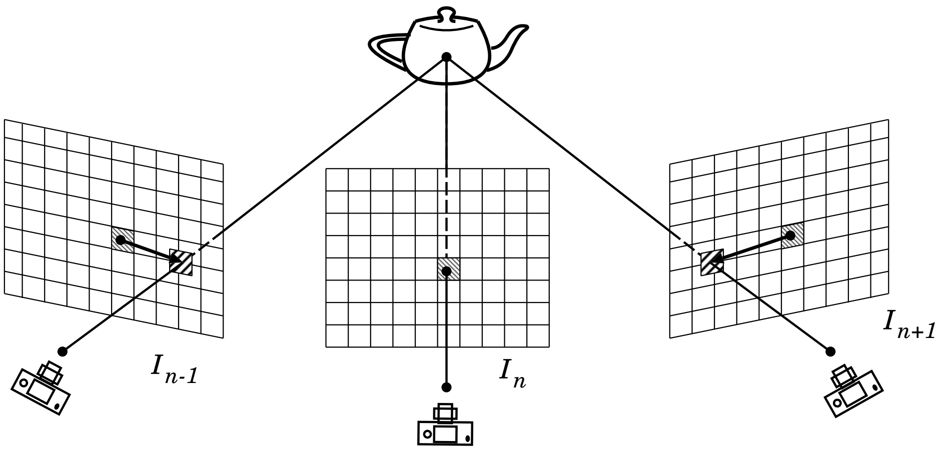


Figure 4.1: Displacement vectors between positions of the corresponding pixels which represent the same scene details in the subsequent animation frames I_{n-1}, I_n, and I_{n+1}. The planar image-warping equation [46] is used to derive displacement vectors that are computed in respect to pixel positions in I_n. The computation requires the range data for I_n, and the camera parameters of all three involved frames. Note that the RGB values are not necessary to compute displacement vectors.

In this chapter, we limit our discussion to the production of high quality walkthrough animations when only camera animation is considered (refer to Chapters 5–7 for the discussion of global illumination in more general animation cases). We assume that a walkthrough animation is of high quality, involving complex geometry and global illumination solutions, thus a single frame rendering incurs significant costs. The other reasonable assumptions are that the animation path and all camera positions are known in advance, that ray tracing (or other high quality rendering methods) for selected pixels is available, that depth (range) data for each pixel is inexpensive to derive for every frame (e.g., using z-buffer), and that the object identifiers for each pixel can be easily accessed for every frame (e.g., using item buffer [95]).

The material in this chapter is organized as follows. In Section 4.2, we briefly describe the animation quality metric used for the keyframe selection. We describe efficient methods of inbetween frames computation in Section 4.3. Section 4.4 shows results obtained using our approach. Section 4.5 concludes this chapter.

4.2 Animation Quality Metric

Automatic assessment of animation quality as perceived by the human observer is becoming very important in various applications dealing with digital video encoding and compression techniques. The most successful error metrics are based on the computational models of human vision [43, 93, 74] and are designed specifically for digital video applications.

In this study, we deal exclusively with synthetic images, and we are looking for a metric well tuned to our application requirements, even at the expense of some loss of its generality. As the framework of our animation quality metric (AQM) we decided to expand the VDP [14] into the temporal domain. The central part of the AQM is a model for the spatiovelocity Contrast Sensitivity Function (CSF), which specifies the detection threshold for a stimulus as a function of its spatial and temporal frequencies. The visual pattern velocity required by this model is estimated based on the average IF magnitude between the currently considered frame and the previous/subsequent frames (refer to Figure 4.1). Also, visual masking is modelled, which affects the detection threshold of a stimulus as a function of the interfering background stimulus which is closely coupled in space. The AQM models temporal and spatial mechanisms (channels) which are used to represent the visual information at various scales and orientations in a similar way as the primary visual cortex does. For more detailed description of the AQM processing refer to [50].

As input to the AQM two comparison animation sequences are provided. For every pair of input frames a map of probability values is generated as output, which characterizes the difference in perceivability. This map is used in our walkthrough animation rendering algorithm to identify pixels for which IBR techniques do not provide sufficient quality and more precise computation must be performed. We apply the AQM to guide inbetween frame computation, which we discuss in Section 4.3.2.

4.3 Inbetween Frames Rendering

For animation techniques relying on keyframing the rendering cost depends heavily upon the efficiency of inbetween frame computation because the inbetween frames usually significantly outnumber the keyframes. In this work, we apply well-known off-the-shelf IBR solutions suitable for inbetween frame computations, which are based on simple data structures and do not require intensive preparatory computations. We use a combination of the following standard techniques:

- To account for proper IF computation and occlusion relations we select 3D warping and warp ordering algorithms developed by McMillan [46], which require just the reference image and the corresponding range data.

- To reduce gaps between stretched samples during image reprojection we use the adaptive “splatting” technique proposed by Shade *et al.* [62].

- To remove holes resulting from occluded objects we composite the two warped keyframes as proposed by Mark *et al.* [44]. Pixels depicting objects occluded in the two warped keyframes are computed using ray tracing.

This choice is the result of extensive analysis of the suitability of existing IBR techniques for walkthrough applications which we presented in [49]. Figure 4.2 summarizes the processing flow for the inbetween frame derivation using the techniques we selected.

The quality of pixels computed using the IBR techniques selected by us can be deteriorated occasionally due to such reasons as occlusions in the keyframes of the scene regions that are visible in the inbetween frames, specular properties of depicted objects, and so on. In the following section, we discuss our solutions to modifying bad pixels which could affect the animation quality as perceived by the human observer. One of important factors toward reducing the number of bad pixels is the selection of keyframes along the walkthrough trajectory. In Section 4.3.2 we propose an efficient method for adaptive keyframe selection which is specifically tuned for deriving inbetween frames using IBR techniques and is guided by the AQM predictions. In Section 4.3.3 we briefly discuss our spatiotemporal antialiasing solution, which is applied as a post-processing step to all animation frames.

4.3.1 Quality Problems with Inbetween Frames

The goal of our animation rendering solution is to maximize the number of pixels computed using the IBR approach without deteriorating the animation quality. However, the quality of pixels derived using IBR techniques is usually lower than ray-traced pixels, e.g., in the regions of inbetween frames which are expanded (zoomed-in) in respect to the keyframes.

Human vision is especially sensitive to distortions in image regions with low IF velocities. As a part of our antialiasing solution we replace IBR-derived pixels in such regions with ray-traced pixels. The replacement is performed when the IF velocity is below a specified threshold value, which we estimated in subjective and objective (using the AQM) experiments (for more details refer to [50]).

Specular effects often attract the viewer’s attention and are of high local contrast, so special care should be taken to process them properly. In existing IBR

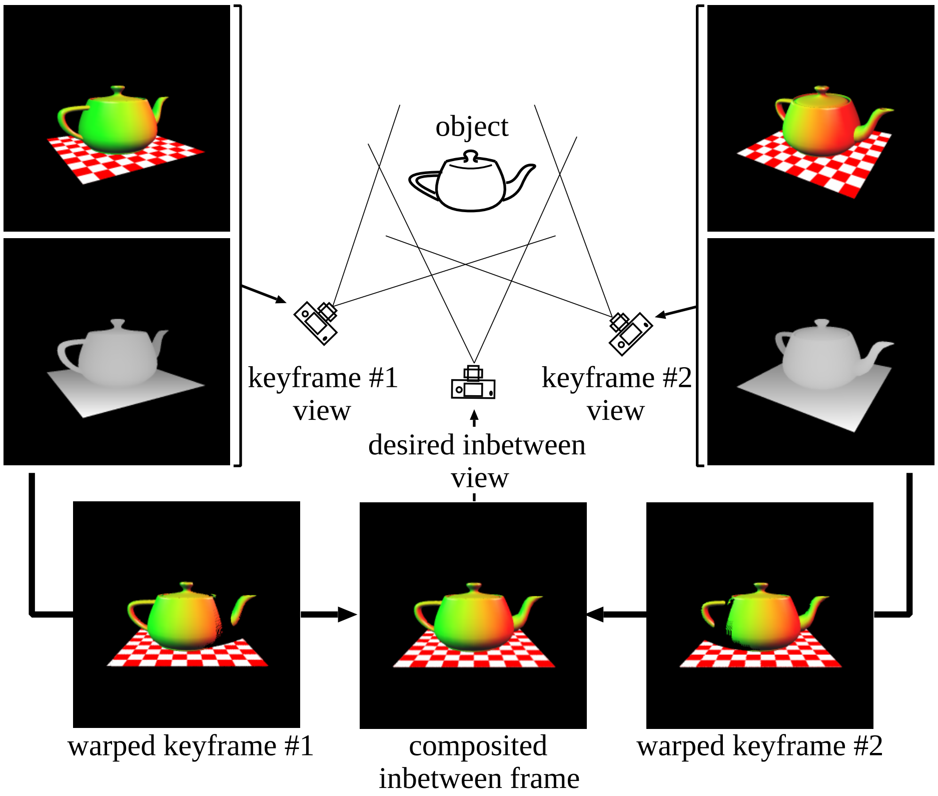


Figure 4.2: Derivation of an inbetween frame based on two keyframes, and the corresponding range data (the distance is shown in greyscale). At first, the keyframes are 3D warped to the inbetween frame viewpoint. Then, composition of the keyframe warps is performed accounting for the proper solution of occlusion problems.

methods, handling of non-diffuse reflectance functions requires very costly preprocessing to produce a huge number of images needed to obtain crisp mirror reflections [47, 41]. Because of these problems we decided to use ray tracing for pixels depicting objects with strong specular or transparent properties. We perform such a computation only for those objects for which well visible artifacts are predicted by the AQM.

Scene fragments that are visible in the inbetween frames which are not present in the keyframes cannot be properly derived using the IBR techniques. We apply ray tracing to compute the missing image fragments. To reduce the number of pixels which must be ray traced we propose a technique for adaptive selection of keyframes, which we discuss in the following section.

4.3.2 Adaptive Selection of Keyframes

The selection of keyframes should be considered in the context of the actual technique used for inbetween frame computation. Our goal is to find an inexpensive and automatic solution for the initial placement of keyframes which improves the IBR rendering performance. We assume a fixed number of initial keyframes and we want to minimize the number of pixels which cannot be properly derived from the keyframes due to visibility problems [46]. Since in our animation rendering solution all such pixels are computed using ray tracing for all inbetween frames, our objective is to reduce their number. In this section, we discuss an adaptive refinement of keyframe placement which is performed taking into account perceptual considerations, and is guided by AQM predictions.

After the initial uniform frame placement, every resulting segment S of length $\delta = N + 1$ is processed separately through application of the following recursive procedure:

1. Generate the first frame I_0 and the last frame I_N in segment S using ray tracing. The keyframes that are shared by two neighboring segments are computed only once.

2. Derive two instances of the central inbetween frame $I'_{[N/2]}$ and $I''_{[N/2]}$ for segment S by 3D warping [46] the keyframes:

   - I_0: $I'_{[N/2]} = 3DWarp(I_0)$, and
   - I_N: $I''_{[N/2]} = 3DWarp(I_N)$.

3. Use the AQM to compute the probability map P_{Map} with perceivable differences between $I'_{[N/2]}$ and $I''_{[N/2]}$.

4. Mask out from P_{Map} all pixels that must be ray traced because of the IBR deficiencies (discussed in Section 4.3.1). The following order for masking out pixels is taken:

   (a) Mask out from P_{Map} pixels with low PF values.

   (b) Mask out from P_{Map} pixels depicting objects with strong specular properties (i.e., mirrors, transparent, and glossy objects). The item buffer [95] of frame $I_{[N/2]}$ is used to identify pixels representing objects with such properties. Only those specular objects are masked out for which the differences between $I'_{[N/2]}$ and $I''_{[N/2]}$ as reported in P_{Map} can be readily perceived by the human observer. In Section 4.4 we provide details on setting the thresholds of AQM response, which are used by us to discriminate between the perceivable and imperceivable differences.

   (c) Mask out from P_{Map} holes composed of pixels that could not be derived from keyframes I_0 and I_N using 3D warping.

5. If masked-out P_{Map} shows the differences between $I'_{[N/2]}$ and $I''_{[N/2]}$ for a bigger percentage of pixels than the assumed threshold value:

   (a) Split S at frame $I_{[N/2]}$ into two subsegments S_1 $(I_0, \ldots, I_{[N/2]})$ and S_2 $(I_{[N/2]}, \ldots, I_N)$.

   (b) Process recursively S_1 and S_2, starting this procedure from the beginning for each of them.

   Else

   (a) Composite $I'_{[N/2]}$ and $I''_{[N/2]}$ with correct processing of object occlusions [44, 62] to derive $I_{[N/2]}$.

   (b) Ray trace all pixels which were masked out in the step 4 of this procedure, and composite these pixels with $I_{[N/2]}$.

   (c) Repeat two latter steps for all remaining inbetween frames, i.e.,
   $I_1, \ldots, I_{[N/2]-1}$ and $I_{[N/2]+1}, \ldots, I_{N-1}$ in S.

To avoid image quality degradation resulting from multiple resamplings, always the fully ray-traced keyframes I_0 and I_N are warped in step 5c to obtain all inbetween frames in S. Pixels to be ray traced, i.e., pixels with low PF values, pixels depicting specular objects with visible differences (such objects are selected once for the whole S in step 4b), and pixels with holes resulting from the IBR processing must be identified for every inbetween frame separately.

We evaluate the AQM response only for frame $I_{[N/2]}$. We assume that derivation of $I_{[N/2]}$ applying the IBR techniques is the most error-prone in the whole segment S because its arclength distance along the animation path to either the I_0 or I_N frames is the longest one. This assumption is a trade off between the time spent for rendering and for the control of its quality (we discuss the costs of AQM in Section 4.4), but in practice, it holds well for typical animation paths.

Figure 4.3 summarizes the computation and composition of an inbetween frame. We used a dotted line to mark those processing stages that are performed only once for segment S. All other processing stages are repeated for all inbetween frames.

4.3.3 Spatiotemporal Antialiasing

As a final step, the animation quality is enhanced by an efficient spatiotemporal antialiasing, which utilizes the IF to perform a motion-compensated filtering [63].

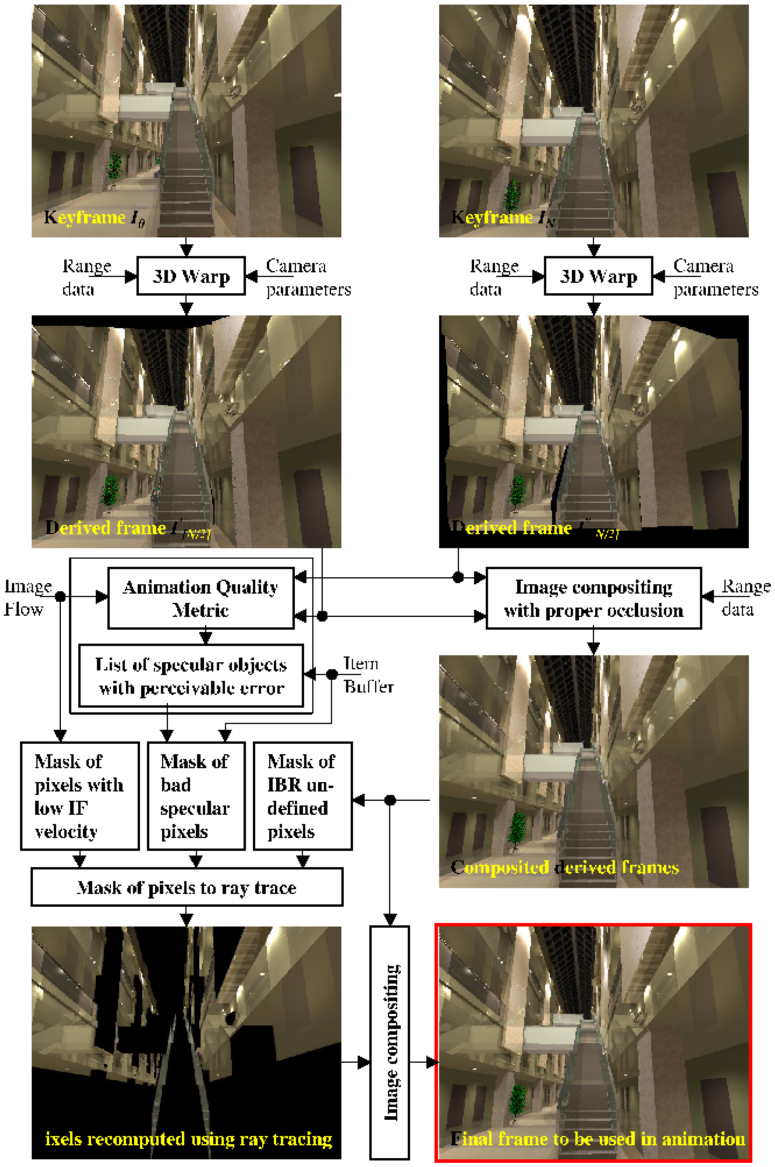
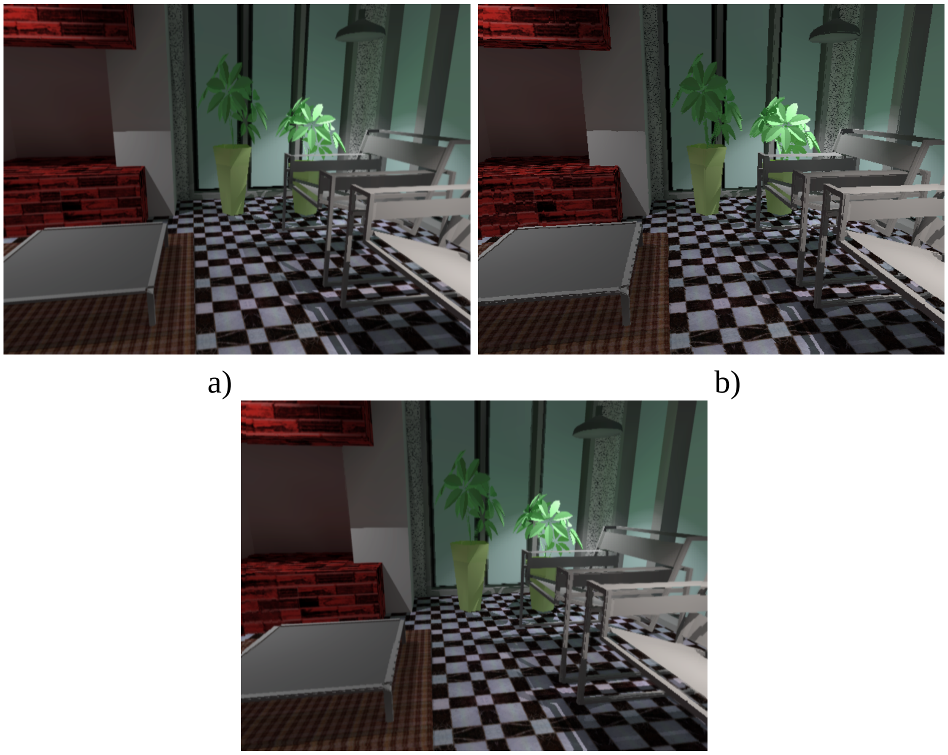

Keyframe I_0
Keyframe I_N
Range data
3D Warp
Camera parameters
Range data
3D Warp
Camera parameters
Derived frame $I'_{N/2}$
erived frame $I''_{N/2}$
Image Flow
Animation Quality Metric
Image compositing with proper occlusion
Range data
List of specular objects with perceivable error
Item Buffer
Mask of pixels with low IF velocity
Mask of bad specular pixels
Mask of IBR undefined pixels
Composited derived frames
Mask of pixels to ray trace
ixels recomputed using ray tracing
Image compositing
Final frame to be used in animation

Figure 4.3: The processing flow for inbetween frames computation.

The filter parameters have been tuned using the AQM predictions of animation quality as perceived by the human observer. These parameters adapt locally to the visual pattern velocity which is estimated based on the IF as shown in Figure 4.1. For more details on our spatiotemporal antialiasing technique refer to [50].

Figures 4.4a and b show a single frame from an animation sequence, that was obtained using ray tracing with antialiasing and that from our technique of inbetween frame computation described in in this section. Figure 4.4c shows the frame depicted in Figure 4.4b, which was processed using our spatiotemporal antialiasing. Although, the frames in Figures 4.4a and c exhibit many perceivable differences when observed as still images, they are visually indistinguishable when observed within animation sequences.



a) b)

c)

Figure 4.4: Selected animation frame computed using: a) ray tracing with antialiasing, b) composition of IBR-derived and ray-traced pixels, and c) as in b), but processed by our spatiotemporal antialiasing solution. Note that frames in a) and c) are visually indistinguishable when observed within the animation context.

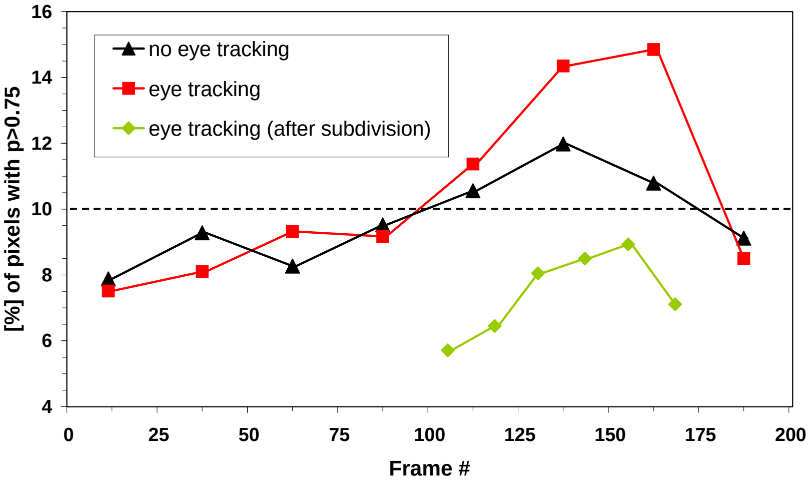
4.4 Results

As the case study in this research we selected a walkthrough animation for two different scenes: the ATRIUM of the University of Aizu and a ROOM (refer to Figures 4.3 and 4.4). The main motivation for this choice were the interesting occlusion relationships between objects which are challenging for IBR. In the case of the ATRIUM scene, a vast majority of the surfaces exhibit some view-dependent reflection properties, including the mirror-like and transparent surfaces, which made inbetween frames calculation more difficult. Under such conditions, the AQM guided selection of keyframes and glossy objects within inbetween frames to be recomputed was more critical, and wrong decisions concerning these issues could be easy to perceive. For the ROOM scene we disabled specular properties, and we designed an animation path which causes great variation in the IF velocity. Our goal was to investigate the performance of our animation rendering solution for the conditions in which eye sensitivity changes dramatically.

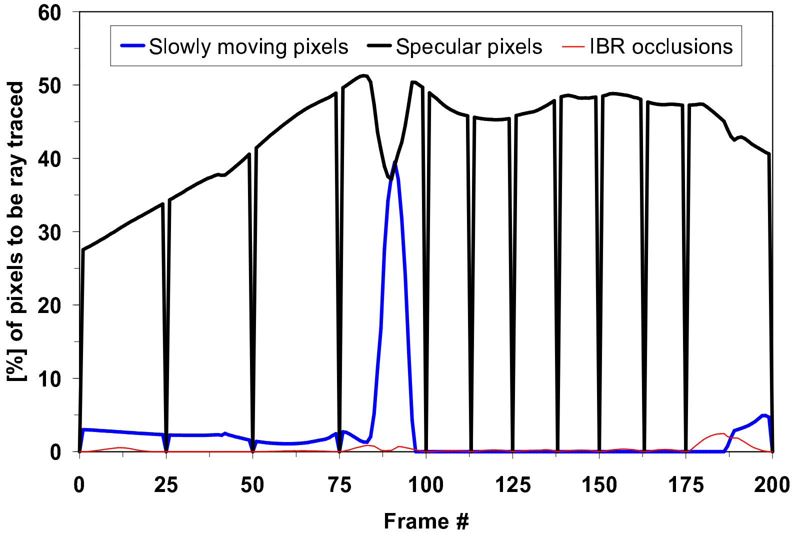
For our experiments we selected a walkthrough sequence of 200 frames for the ATRIUM, and 448 frames for the ROOM. The resolution of each frame was 640×480 (to accommodate for the NTSC standard). At the initial keyframe selection step, we assumed an animation segment length of $\Delta = 25$ frames. For the ATRIUM walkthrough we kept the length of every segment fixed, i.e., $\delta_i = \Delta$, because changes of the average IF velocity computed for every frame are relatively small. For the ROOM scene the average IF velocity variates significantly, so we adjusted the length of every segment δ_i using an algorithm presented in [50]. The goal of such adjustment was the reduction of the percentage of pixels with occlusion problems which arise in IBR techniques.

As described in Section 4.3.2, for every segment S we run the AQM once to decide upon the specular objects which require recomputation. The AQM is calibrated in such a way that 1 JND unit corresponds to a 75% probability that an observer can perceive the difference between the corresponding image regions (such a probability value is the standard threshold value for discrimination tasks [14]). If a group of connected pixels representing an object (or a part of an object) exhibits differences greater than 2 JND (93.75% probability of discrimination) we select such an object for recalculation. If for an object the differences below 2 JND are reported by the AQM then we estimate the ratio of pixels exhibiting such differences to all pixels depicting this object. If the ratio is bigger than 25%, we select such an object for recomputation - 25% is an experimentally selected trade-off value, which makes possible a reduction in the number of specular objects requiring recomputation, at the expense of some potentially perceivable image artifacts. These artifacts are usually hard to notice unless the observer's attention is specifically directed to the given image region.

After masking out the pixels to be recomputed, the decision on further split-

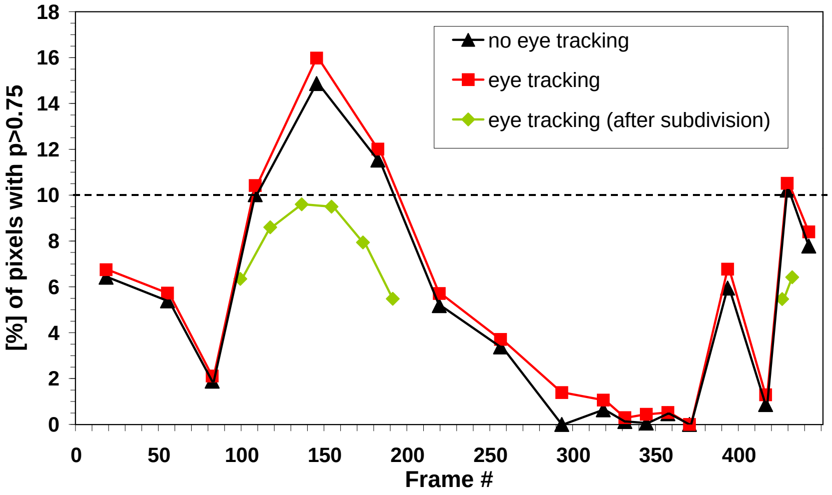


a)

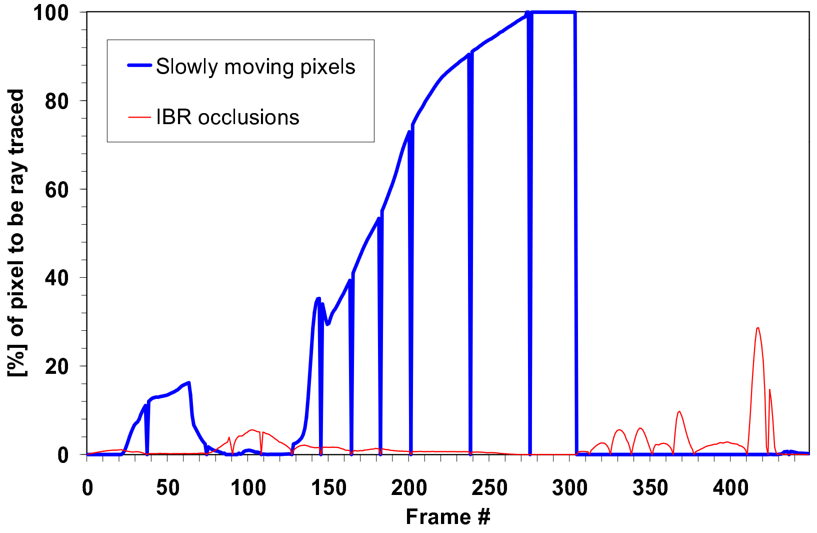


b)

Figure 4.5: ATRIUM walkthrough statistics: a) the AQM prediction of the perceived differences between the warped images of two neighboring keyframes (taking into account various retinal image velocity), and b) the percentage of pixels to be recalculated by ray tracing. In a) lines connecting the symbols were added for readability and they do not have any meaning for unmarked frames.



a)



b)

Figure 4.6: ROOM walkthrough statistics: a) the AQM prediction of the perceived differences between the warped images of two neighboring keyframes (taking into account various retinal image velocity), and b) the percentage of pixels to be recalculated by ray tracing. In a) lines connecting the symbols were added for readability and they do not have any meaning for unmarked frames.

ting S is made using AQM predictions for the remaining pixels. The predictions are expressed by the percentage of unmasked pixels for which the probability p of detecting the differences is greater than 0.75. Based on experiments that we conducted, we decided to split every segment S when the percentage of such pixels is bigger than 10%. When computing the AQM predictions that we used to decide upon segment splitting, we assumed good tracking of moving visual patterns with smooth-pursuit eye movements. The filled squares in Figures 4.5a and 4.6a show such predictions for the inbetween frames located in the middle of every initial segment S. Segments with AQM predictions over 10% were split and the filled diamonds show the corresponding reduction of the predicted perceivable differences. We also performed experiments assuming higher levels of retinal velocity for our walkthrough animation. The filled triangles in Figures 4.5a and 4.6a show the AQM predictions when the retinal velocity is equal to the IF (eye movements are ignored). For all segments selected for splitting based on smooth-pursuit eye movement assumption, the AQM predictions also exceeded the threshold of 10% when the eye movements were ignored.

The overall costs of inbetween frame computations are strongly affected by the average number of pixels that must be ray traced. The graph in Figure 4.5b shows the percentage of pixels depicting specular objects that are replaced by ray traced pixels in the ATRIUM walkthrough sequence. This graph also shows the percentage of replaced pixels due to IBR occlusion problems, and the high sensitivity of the visual system for image patterns moving with low velocity (the velocity threshold of 0.5 degree/second was assumed). Obviously, a given pixel was replaced only once, and we assumed the following processing order of pixels replacement: 1) pixels depicting slowly moving patterns, 2) pixels with possible reflection/refraction artifacts, and 3) pixels with occlusion problems. Figure 4.6b shows the equivalent results for the scene ROOM.

To evaluate the efficiency of our animation rendering system we compared the average time required for a single frame of our test walkthroughs using the following rendering methods: ART - fully ray traced frames with antialiasing (using adaptive supersampling) which are commonly applied in the traditional rendering animation approach, RT - fully ray traced frames (one sample per pixel), and IBR+RT frames generated using our approach with mixed ray traced and IBR-derived pixels. Table 4.4 summarizes the obtained results for the ATRIUM and ROOM walkthroughs. In the case IBR+RT, we included the computation involved in IBR rendering (which requires about 12 seconds to warp and composite two keyframes), motion-compensated 3D filtering which added an overhead of 10 seconds per frame (refer to Section 4.3.3), and AQM processing which takes 243 seconds to process a pair of frames (refer to Section 4.2). The AQM computations are so costly mainly because of the software implemented Fast Fourier Transform (FFT). Since our frames are of resolution 640×480 we had to consider images

Scene	ART [minutes]	RT [minutes]	IBR+RT [minutes]
ATRIUM	170.0	40.0	20.5
ROOM	6.9	1.5	1.1

Table 4.1: Average computation time per frame for various animation rendering solutions. All timings were measured on the MIPS 195 MHz processor.

of resolution 1024×512 for the FFT processing. The AQM processing can be shortened by about 50% considering just 512×480 fragments of frames. This is acceptable in many cases because near the frame boundaries many pixels cannot be properly derived using the IBR approach (refer to the warped keyframes shown in Figure 4.3), so they do not contribute to the AQM response.

The most significant speedup was achieved by using our spatiotemporal antialiasing technique and avoiding the traditional adaptive supersampling. Our inbetween frames rendering technique added a further 25–50% of speedup with respect to the RT approach. The tested scenes were hard for our algorithm because of the strong specular reflectance properties exhibited by many of the surfaces (ATRIUM), and the slow motion of the camera, in which case eye sensitivity is high (ROOM). Also, the chessboard-like pattern of textures in the ROOM scene made it quite challenging in terms of proper antialiasing. Even better performance can be expected for environments in which specular objects are depicted by a moderate percentage of pixels, and camera motion is faster.

4.5 Conclusions

In this chapter, we proposed an efficient approach for rendering of high quality walkthrough animation sequences. Our contribution is in developing a fully automatic, perception-based guidance of inbetween frame computation, which minimizes the number of pixels computed using costly ray tracing, and seamlessly (in terms of the perception of animated sequences) replaces them by pixels derived using inexpensive IBR techniques. Also, we have shown two important applications of the image flow obtained as a by-product of IBR processing. It was applied to: 1) estimate the spatio-velocity Contrast Sensitivity Function which made it possible to incorporate temporal factors into our perceptually-based image quality metric, and 2) perform the spatiotemporal antialiasing with motion-compensated filtering based on image processing principles. We integrated all these techniques into a balanced animation rendering system.

CHAPTER 5

Exploiting Temporal Coherence in Photon Density Estimation

In this chapter, we introduce a novel framework for efficient global illumination computation in dynamic environments. We propose a combination of energy- and perception-based error metrics which efficiently guide lighting computation. Using these metrics, the spatiotemporal coherence in lighting distribution can be better exploited and accurate lighting computation can be obtained without degrading the animation quality as perceived by human observers. As a result, a perceptually-homogeneous quality of indirect lighting reconstruction across the spatial and temporal domains is obtained.

5.1 Introduction

The goal of this work is to improve the performance of global illumination computations for high quality animation sequences by exploiting the temporal coherence in indirect lighting distribution. The mesh-based view-independent Density Estimation Particle Tracing (DEPT) algorithm [79], which we extend in this work to handle animated objects, is used as a global illumination framework, but the proposed solutions could be easily applied to other stochastic algorithms such as the photon map [32]. Initially, the lighting function is sparsely sampled in space for all frames (not just for fixed keyframes as in [53, 97], refer to Section 3.1 for a discussion of those techniques) within a given animation segment. Then, based on the obtained results, the decision is made whether the segment can be expanded or contracted in the temporal domain. Since the validity of samples may depend on the particular region in a scene for which indirect lighting conditions change more rapidly, different segment lengths are chosen locally for each mesh element (used to store particle hits), based on the variation of the lighting function. Energy-based statistical measures of such local variations are used to calculate the number of preceding and following frames for which samples can be safely used for a given region. More samples are generated if the quality of the frames obtained for a given segment length is not sufficient. The perception-based Animation Quality Metric (AQM) [49] is used to choose the average number of photons per frame for each segment to prevent perceivable degradation of animation quality. Spatial filtering is performed for those scene regions in which a sufficient number of samples cannot be collected in the temporal domain. For the final rendering, the indirect lighting is reconstructed using the outlined techniques while specular effects and direct lighting are computed for every frame separately by ray tracing.

The material in this chapter is organized as follows. In Section 5.2, we describe our indirect lighting algorithm. In Section 5.3, we present extensions of this algorithm to process photons in temporal domain. In Section 5.4, we describe our algorithm of perception-based lighting simulation for all frames within an animation segment. In Section 5.5, we analyze the accuracy of lighting simulation in our approach. In Section 5.6, we present the results obtained using our approach. In Section 5.7, we conclude this Chapter and investigate future work directions.

5.2 Indirect Lighting Solution

As a framework for global illumination computation, we chose the Density Estimation Photon Tracing (DEPT) algorithm [79]. The DEPT is similar to other stochastic solutions in which photons are traced from light sources towards surfaces in the scene, and the lighting energy carried by every photon is deposited at the hit point locations on those surfaces [29, 64, 86]. A simple photon bucketing on a dense triangular mesh is performed, and every photon is discarded immediately after its energy is distributed to the mesh vertices. Efficient object space filtering substantially reduces visible noise, while the excessive smoothing of the lighting function can be avoided by adaptively controlling the local filter support which is based on stochastically-derived estimates of the local illumination error [79, 86].

An important feature of the DEPT technique is that the bucket-based lighting reconstruction and filtering are very efficient, and the quality of the reconstructed lighting is quite good. Thus, the resulting illumination maps can be displayed immediately, and many variants of lighting reconstruction in the context of temporal photon processing for animations can be inexpensively analyzed to choose the best solution. This is in contrast with other photon tracing techniques which involve costly density estimation techniques such as the kernel methods [64, 86], or the final gather step [32] in order to obtain images of good quality.

Another advantage of the DEPT computation is that a reasonable mesh-based approximation of the direct lighting is available (although in the final frames the direct lighting is recomputed on a per pixel basis), which is required to model the local eye adaptation conditions. The eye adaptation characteristics are needed for proper tone reproduction [75] during the illumination map display, and for the prediction of the eye sensitivity to the errors in indirect lighting. The reduction of these errors is the main objective of the perception-based guidance of temporal photon processing.

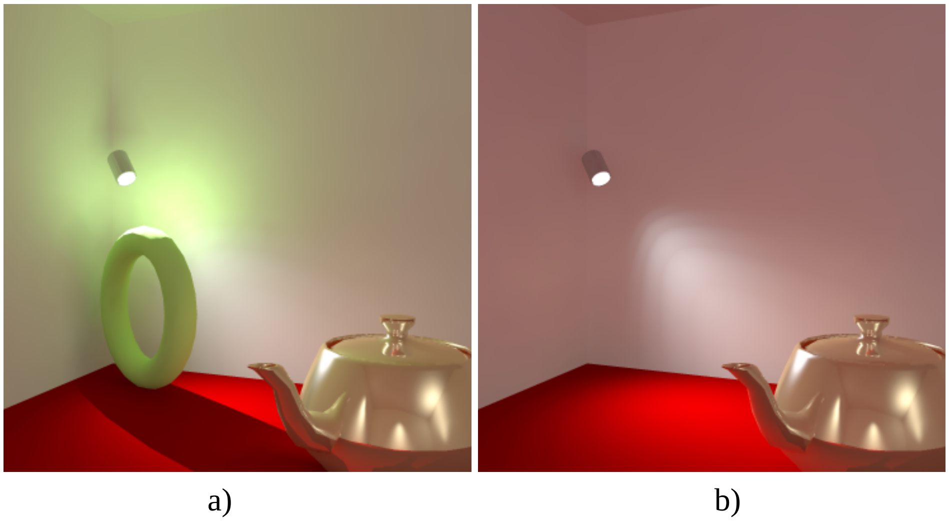


Figure 5.1: Indirect lighting changes can be significant in dynamic environments. Note the color bleeding effect in the room corner caused by the strongly illuminated torus in a). The effect disappears as the torus moves away in b).

The extension of the DEPT algorithm to handle animated sequences of dynamically changing environments requires the storage of photons that are reused in neighbouring frames. The photons are stored on a per frame basis. For each photon, information on its spectral energy distribution is stored to account for reflected light and allow effects such as color bleeding (refer to Figure 5.1). Also, the hit point coordinates in the form of two barycentric coordinates are stored to facilitate distributing the photon energy to the mesh vertices, and to keep the photon position within the mesh elements in relative rather than in absolute terms. Finally, the mesh element identifier is stored, which forms a basis together with the photon barycentric coordinates for re-using a given photon for neighbouring frames. We assume that the motion of objects between the subsequent frames is small enough that even the photons which are “attached” to moving mesh elements approximate the indirect lighting within these elements well. In Section 5.3.1 we present our solutions that prevent re-using of photons for neighbouring frames when the variations in indirect lighting are significant enough to be perceived by the observer.

Since mesh elements are the framework for our temporal processing, the photons are sorted according to the elements which they hit. Apart from storing photon records, an additional table which summarizes the number of photons per element is also kept for every frame. Those tables are later used to efficiently derive some statistics on lighting variations in the temporal domain for a given element (refer to Section 5.3.1). With the help of these statistics we can detect significant changes of lighting. In this case reusing photons for reconstructing the lighting in adjacent frames might be restricted. Using the tables for the purpose of such statistics means that effectively the exact positions of photon hit points within an element are ignored (only the photon number counts), however, the processing of complete photon records can be avoided. In the following section we describe those statistics in more detail. We also explain the spatiotemporal processing of photons for dynamic environments.

5.3 Spatiotemporal Photon Processing

In our technique we assume that photons are traced sparsely for all animation frames and our goal is to minimize the number of those photons without compromising the animation quality. To achieve this goal we exploit the temporal coherence of indirect lighting and for a given frame we also consider photons that were traced for neighboring frames. Ideally, as many frames should be processed as it is required to reduce the stochastic noise below the sensitivity level of the human observer. However, the expansion of the photon collection in the temporal domain might be limited due to changes in dynamic environments that affect the lighting distribution. A contradictory requirement arises between maximizing the number of collected photons and minimizing the number of neighbouring frames (the time span) for which these photons were traced. A trade-off solution to this problem relies on balancing the stochastic noise (resulting from collecting too few photons) and the errors in reconstructed illumination (caused by collecting too many invalid photons in the temporal domain) to make those artifacts as little objectionable as possible for the human observer. The perception-based AQM is used to find the minimal number of photons per frame which is required to make the noise undetectable. An energy-based stochastic error metric, which is applied to each mesh element and to every frame, is used to guide the photon collection in the temporal domain. We found this mesh-element level of applying the energy-based metric to be very efficient, and therefore abandoned the use of perception-based guidance of photon collection at this low level which would be far more expensive.

We describe our energy-based error metric which controls the temporal photon processing and reduces the probability of using invalid photons in the scene regions in which lighting changes rapidly in Section 5.3.1. In Section 5.4 we discuss our techniques of spatiotemporal photon processing, which guarantee that the quality of the indirect lighting reconstruction is consistent through the whole animation.

5.3.1 Error Metric for Temporal Processing

As we argued in the previous section, collecting photons in the temporal domain makes sense only if the lighting distribution does not change too rapidly for subsequent frames. We attempt to detect such changes locally on the level of single mesh elements. The practical question how to distinguish the actual changes in lighting from the stochastic error arises. This problem is especially difficult in our technique because we compute a very small number of photons for every frame, which results in high levels of noise. In practice, this means that only lighting changes that are significantly higher than the noise level can be detected, which requires estimating the noise.

If we assume for a moment that the lighting does not change between subsequent frames, then hitting mesh elements by photons can be modelled well by the Poisson distribution [3]. Since the mesh elements are small the probability p of hitting a given mesh element by a photon is also small, i.e., $p \ll 1$ as required by the Poisson process. Also, different photons hitting a mesh element are mutually independent, i.e., the probability of the same photon hitting a mesh element again as a result of its multiple reflections is small. The Poisson distribution only has a single parameter, the mean μ, which can be estimated as the mean number of photons hitting a mesh element. The standard deviation σ can simply be derived as $\sigma = \sqrt{\mu}$. Thus, the noise level can be estimated as $\mu \pm k\sigma$, where e.g., $k = 2$ (for $\mu = 0$ we assign $\sigma = 1$). Based on this estimate we assume that if the number of photons x hitting a mesh element does not satisfy the condition

$$\mu - k\sigma \leq x \leq \mu + k\sigma \tag{5.1}$$

a change of lighting can be expected and the photon collection for this mesh element is disabled. For a given mesh element the mean μ is estimated for the currently considered frame and the values of x are obtained for the corresponding mesh element in the preceding and following frames. The temporal collection of photons is initiated from the current frame and proceeds for subsequent frames as long as Condition (5.1) is satisfied. The photon processing is performed independently in the directions of the preceding and following frames. Thus, the photon collection for a given mesh element may be asymmetric with respect to the current frame when Condition (5.1) is violated earlier in one of those directions than in another one.

There are many possible sources of error, which may prevent the application of Condition (5.1) from working robustly for all mesh elements. First of all the estimate of μ might be inexact since we use a Monte Carlo solution at very early stages of convergence. To reduce the influence of outliers on the estimate of μ we consider the mean number of photons hitting a given mesh element for three subsequent frames. We apply this procedure to derive x for the same reason. More than three subsequent frames might be considered but then the μ estimate might be affected by changes in lighting. Clearly, the procedure works the better the more photons per frame are computed.

If the average number of photons per mesh element is small, assigning $k > 2$ might be considered to compensate for inaccuracies in the μ estimates. Also, small lighting changes on the level $\pm k\sigma$ are usually less perceptible than the temporal aliasing which becomes visible when the temporal photon collection is prematurely disabled. In practice, we assume the minimal extent of the temporal processing to be at least 3 frames. Even if the lighting conditions change drastically for a pair of subsequent frames, the visibility thresholds are usually elevated due to the temporal masking for up to 100 milliseconds [24]. Since new lighting details cannot be seen well for 2–3 animation frames the lighting reconstruction accuracy can be relaxed for those frames, and tracing additional photons can be avoided.

In the following section we describe the algorithm of animation rendering, which extensively uses the procedure of adaptive photon collection in the temporal domain described in this section.

5.4 Algorithm

The animation sequence is split into animation segments, which are then processed sequentially one by one. The organization of a single segment is depicted in Figure 5.2.

The frame K divides the segment into two halves of length $F_{\max}/2$ each. The goal of the segment processing is to reconstruct indirect lighting for all $F_{\max} + 1$ frames. To enable temporal processing of the whole segment the photons collected for neighbouring segments for up to $F_{\max}$ frames preceding K and up to $F_{\max}$ frames following K are also considered. For the very first and last animation segments the lack of those extra frames is compensated by tracing more photons.

The processing of every animation segment consists of five major steps:

1. Initialization: determination of the number of initial photons per frame, and processing of those photons for all frames in the current segment.

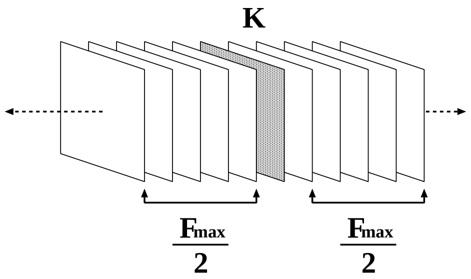


Figure 5.2: Organization of an animation segment.

2. Adjustment of the segment length depending on the temporal variations of indirect lighting which are measured using energy-based criteria (refer to Section 5.3.1).

3. Adjustment of the number of photons per frame based on the AQM response in order to limit the perceivable stochastic noise to an acceptable level.

4. Spatiotemporal reconstruction of indirect lighting for all mesh elements guided by the same energy-based criteria as in Step 2.

5. Spatial filtering step for those mesh elements that did not meet the perceptual and energy criteria in the previous two steps.

In practice, Steps 1–3 are performed not only for the current segment but for the subsequent segment as well, which results in the processing of all photons that are used for indirect lighting reconstruction in the current segment at Step 4. Obviously, photons for the preceding segment that has already been processed are also available for the temporal processing of frames in the current segment.

In the following sections we describe all steps in detail.

5.4.1 Initialization

In the initialization step we decide how many photons N_{frame} per frame have to be shot for a given animation segment. Also, the initial segment length $F_{\max}$ needs to be chosen based on the anticipated complexity of the lighting changes during an animation. The initial settings of $F_{\max}$ and N_{frame} are adjusted later for every animation segment using the energy-based and perception-based error metrics. However, the better the initial guess is, the smaller number of iterations involving those metrics is needed. In our approach $F_{\max}$ is set manually by the user for the first processed segment. The adjusted $F_{\max}$ for a previous segment is assigned as the initial length of a subsequent segment. Based on F_{max} chosen for a given segment, N_{frame} is decided automatically as follows.

An initial value for N_{frame} should take into account basic lighting characteristics of the considered animation segment. To derive such an initial guess we applied the energy-based measure of the error of indirect lighting simulation[1] E. We observed that by setting the maximum error value E_{max} to lie inside the range of 1–5% we usually obtained images of good quality. Further computation basically did not introduce any improvement as perceived by the human observer [79]. In practice, we assume a less conservative $E_{\mathrm{max}} = 5\%$. Now, if we would run the DEPT computation until $E \approx E_{\mathrm{max}}$ we would find the number of photons N_{max} that is required to achieve this accuracy. Since we want to reconstruct indirect lighting for a given frame using photons computed for up to $F_{\mathrm{max}}/2$ preceding and $F_{\mathrm{max}}/2$ following frames, we could estimate $N_{\mathrm{frame}} = N_{\mathrm{max}}/(F_{\mathrm{max}} + 1)$. In practice, we do not want to run the computation for a single frame long enough to reach the error level E_{max}. However, we can directly get a good estimate of N_{frame} much faster by using the basic property of stochastic solution convergence stating that the error is proportional to the inverse square root of the number of traced photons [36]:

$$\frac{E}{E_{\mathrm{max}}} \approx \sqrt{\frac{N_{\mathrm{max}}}{N_{\mathrm{frame}}}}$$

This means that N_{frame} can be estimated by running the pilot DEPT computation until $E \approx E_{\mathrm{max}}\sqrt{F_{\mathrm{max}} + 1}$.

Finally, for all frames in the current animation segment N_{frame} photons are traced, and their hit point records are sorted for every mesh element.

5.4.2 Choosing the Animation Segment Length

The main goal of adjusting the maximum segment length F_{max} is to reduce the animation artifacts caused by collecting photons in the temporal domain over too many frames, as this may lead to an invalid lighting reconstruction. For the scene regions in which the temporal changes of the lighting function are fast, the collection of photons can be performed only for a small number of subsequent frames. Conversely, for the scene regions in which the temporal changes are slow the photons should be collected for as many frames as possible to reduce the stochastic noise.

In Section 5.3.1 we described our procedure of adaptive photon collection in the temporal domain. The maximum number of frames for which photons are collected using this procedure is limited by the segment length F_{max}. Increasing

[1] The detailed description of the stochastic method used to estimate the lighting simulation error in the framework of the DEPT algorithm can be found in [80].

$F_{\max}$ incurs additional costs of processing a larger number of frames, and is justified only when collecting photons from $F_{\max}$ frames is possible for a significant percentage p_{me} of the mesh elements. In practice, the user sets a certain threshold value p_{thr}, e.g., 40%, and $F_{\max}$ is increased until the condition $p_{\mathrm{me}} < p_{\mathrm{thr}}$ is met. Such a test is performed only once per segment, when the lighting is reconstructed for the central keyframe K. Note that increasing $F_{\max}$ causes moving K apart $F_{\max}/2$ frames from the end of the previously processed segment (or from the beginning of animation sequence for the very first processed segment). The $F_{\max}$ value obtained for K is then used for all frames in the current segment.

5.4.3 Choosing the Number of Photons

The main goal of adjusting the number of photons per frame N_{frame} is to keep the noise below a perceivable level. Increasing N_{frame} for each frame in a given segment is an expensive operation and should be performed if the adjustment of $F_{\max}$ performed in the previous step did not provide the required animation quality as perceived by the human observer.

The AQM is used to measure the errors of the indirect lighting reconstruction which enables the perception-guided selection of N_{frame} to minimize the computational costs without degrading the animation quality. The AQM requires two animation frames I_1 and I_2 as input, and will then predict the perceivable differences between them. Ideally, a frame resulting from the temporal photon processing should be compared to some exact reference solution. Since such a reference solution is not available in practice, we decided to measure the differences in indirect lighting reconstructed for the central frame K by splitting the photons collected for all frames in a given segment into two halves (the even and odd photons). The indirect lighting in $I_1(K)$ and $I_2(K)$ is reconstructed using these halved sets of photons. In order to measure the level of noise in the conditions in which the actual temporal photon processing is performed for all animation frames, the procedure of adaptive photon collection in the temporal domain is used for the $I_1(K)$ and $I_2(K)$ computation as well (refer to Section 5.3.1).

The approach of halving sets of photons is quite conservative because according to the Monte Carlo methods theory [36] the stochastic error of indirect lighting reconstruction in the frame $I(K)$ that is obtained for 100% of photons is smaller by the factor $\sqrt{2}$ with respect to $I_1(K)$ and $I_2(K)$. The perceivable differences as predicted by the AQM usually are reduced by an even larger factor if the number of photons is doubled.

As the result of AQM processing a map M_{AQM} is generated which shows the prediction of perceivable differences in indirect lighting between all corresponding pixels in $I_1(K)$ and $I_2(K)$. As a scalar metric of the frame quality the percentage d_{AQM} of M_{AQM} pixels with differences over one unit Just Noticeable

Difference (JND) is assumed [49]. The user chooses a certain threshold value d_{thr} of the AQM predicted differences, and when $d_{\mathrm{AQM}} > d_{\mathrm{thr}}$, N_{frame} is doubled and the whole procedure is repeated until $d_{\mathrm{AQM}} < d_{\mathrm{thr}}$.

To reduce the costs of Human Visual System (HVS) modelling the AQM processing is performed only once per segment for the central frame K. Thus, the N_{frame} value obtained for K is assumed to be valid for all frames in a given segment. In practice, this trade-off approach works well because the differences in indirect lighting are usually small for a given animation segment whose length $F_{\max}$ was adjusted to reduce such differences (refer to Section 5.4.2).

5.4.4 Indirect Lighting Reconstruction

After establishing $F_{\max}$ and N_{frame} further processing of all frames in a given segment becomes straightforward. At first, the valid photons are collected for each mesh element, using the procedure of adaptive photon collection in the temporal domain described in Section 5.3.1. Then the standard procedure for deriving illumination at mesh vertices is applied (refer to Section 5.2).

5.4.5 Repairing Noisy Pixels

The procedure described in the previous section may potentially result in locally noisy images[2] for the scene regions (e.g., moving objects) in which the indirect lighting changes much faster than for the remaining parts of environment. For such regions collecting photons in the temporal domain is usually limited to a few subsequent frames. Obviously, the noise level could be reduced by increasing N_{frame} which is costly. Note that in the procedure of selecting N_{frame} using the AQM we allow perceivable differences for up to d_{thr} pixels (refer to Section 5.4.3). When the perceivable problems concern only a small fraction of mesh elements, then for efficiency reasons, increasing N_{frame} for all frames in a given segment should be avoided.

Our solution relies on using the spatial filtering performed in the object space selectively for those mesh elements for which the expansion in the temporal domain was not possible, resulting in a small number of collected photons. We apply the filtering algorithm which was originally proposed in [79]. To achieve the required level of accuracy of reconstructed lighting at a given vertex, photons hitting a region h centered at this vertex are considered. Stochastic variance estimates of the local illumination are used to decide upon the size of h. This effectively reduces noise, however some bias is introduced to the reconstructed lighting (refer

[2] As a matter of fact for a vast majority of tests that we performed, we were not able to notice such problems.

to [79] for a formal derivation of a mathematically-sound measure of illumination accuracy and a detailed description of the filtering algorithm).

Using spatial filtering is equivalent to trading in the spatial details of indirect lighting in order to remove excessive noise. If this approach is not acceptable, a final gathering step [32] could be performed. However, we did not apply this solution because of its significant cost. We found that the spatial filtering approach as applied in the indirect lighting reconstruction produces good results in terms of the animation quality as perceived by the human observer.

5.5 Accuracy Considerations

The accuracy of the indirect lighting reconstruction using our algorithm described in Section 5.4 is limited by the spatial resolution of the mesh used for collecting photons. Consequently the solution is biased. On the other hand the mesh resolution can be set arbitrarily fine and more photons can be traced. For those mesh elements that still collect too few photons in the temporal domain, the spatial filtering discussed in Section 5.4.5 can be used to remove visible noise. Another source of the bias is the temporal blur resulting from the collection of invalid photons in the temporal domain. The level of blurring can be controlled in the energy terms and traded for the stochastic noise by decreasing the value of parameter k in Condition (5.1).

The AQM is used to measure the perceivable differences between two equally biased indirect lighting solutions, which means that all measured differences between frames $I_1(K)$ and $I_2(K)$ result from the stochastic noise (refer to Section 5.4.3). Effectively the AQM provides a conservative stopping condition for photon tracing when the noise falls below the sensitivity level of the human observer. Tracing more photons cannot improve the perceived quality of the indirect lighting reconstruction due to limitations in the spatial mesh resolution.

5.6 Results

We present results that we obtained for the ROOM scene (about 5,300 mesh elements). Also, we briefly summarize the results obtained for another scene ATRIUM (about 45,000 mesh elements), which are qualitatively very similar and therefore do not need to be discussed in full length. Both scenes were designed in such a way that moving objects significantly affected the lighting distribution. Also, some scene regions are illuminated exclusively by indirect lighting which imposes higher requirements on its reconstruction. We begin with discussing the experimental results for the adaptive algorithm of temporal photon processing discussed in Section 5.3.1. Then we discuss the step by step results obtained for the spatiotemporal photon processing discussed in Section 5.4.

The simplest scenario of temporal photon processing is to consider the fixed number of preceding and following frames. However, this approach may lead to significant errors as illustrated in Figures 5.3. Figure 5.3a shows the correct reference frame obtained for the converged DEPT solution. In this scene a spot light illuminates the bottom of the aircraft, and the highlight on the floor is caused by the light reflected from the aircraft. Note that as the result of non-adaptive temporal processing for all $F_{\max} = 30$ frames the highlight is significantly washed out (Figure 5.3b). When applying our adaptive photon collection technique (Section 5.3.1) the highlight shown in Figure 5.3c is similar to the reference frame in Figure 5.3a. Figure 5.3d shows the AQM produced map M_{AQM}, in which red color marks pixels for which visible differences are predicted.

Figures 5.4a and b summarize the AQM predicted percentage of pixels d_{AQM} with perceivable differences derived from M_{AQM} for various settings of N_{frame} and $F_{\max}$ for non-adaptive and adaptive photon collection approaches. As can be seen in Figure 5.4a for the non-adaptive approach, expanding $F_{\max}$ initially leads to reducing d_{AQM}, but then the collection of invalid photons results in increasing d_{AQM} for large $F_{\max}$. The corresponding characteristics for the adaptive approach shown in Figure 5.4b are extremely favorable because the expansion of $F_{\max}$ always leads to the reduction of d_{AQM}, which means that collecting invalid photons is mostly avoided.

Following the subsequent processing steps described in Section 5.4 we obtained the following animation settings for the ROOM scene. As the result of the initialization procedure $N_{\mathrm{frame}} = 10,000$ and $F_{\max} = 15$ were chosen. The animation was split into three segments and the final settings computed for the central segment frames K_i are summarized in Table 5.1. In segment K_3 N_{frame} is smaller and $F_{\max}$ more expanded because strong direct lighting washes out some imperfections of the indirect lighting reconstruction. Also, changes of lighting are less dynamic in this animation segment.

	N_{frame}	$F_{\max}$
K_1	40,000	30
K_2	40,000	30
K_3	10,000	44

Table 5.1: Final settings for the ROOM scene animation.

Figure 5.5 summarizes the results obtained for K_2 using the energy-based procedure of photon validity estimation for subsequent frames described in Section 5.4.2. We assumed $p_{\mathrm{thr}} = 40\%$. The maximum segment length $F_{\max} = 30$

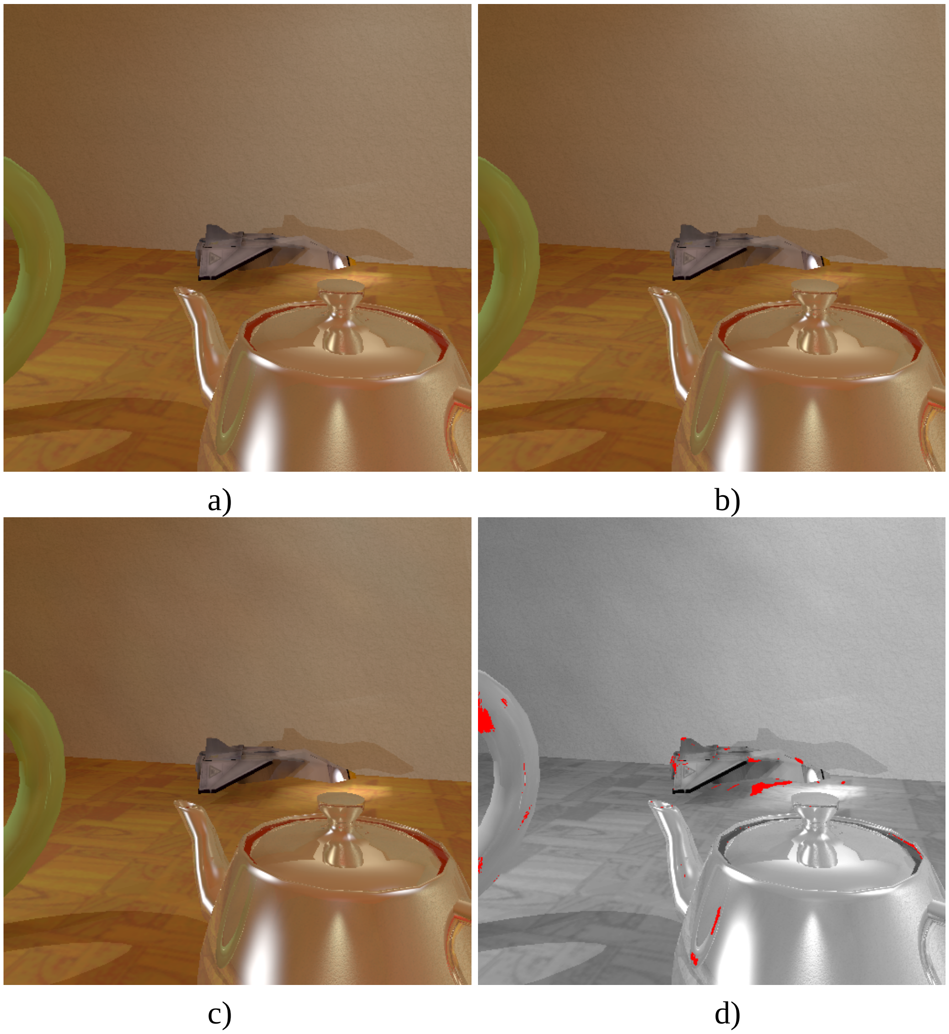
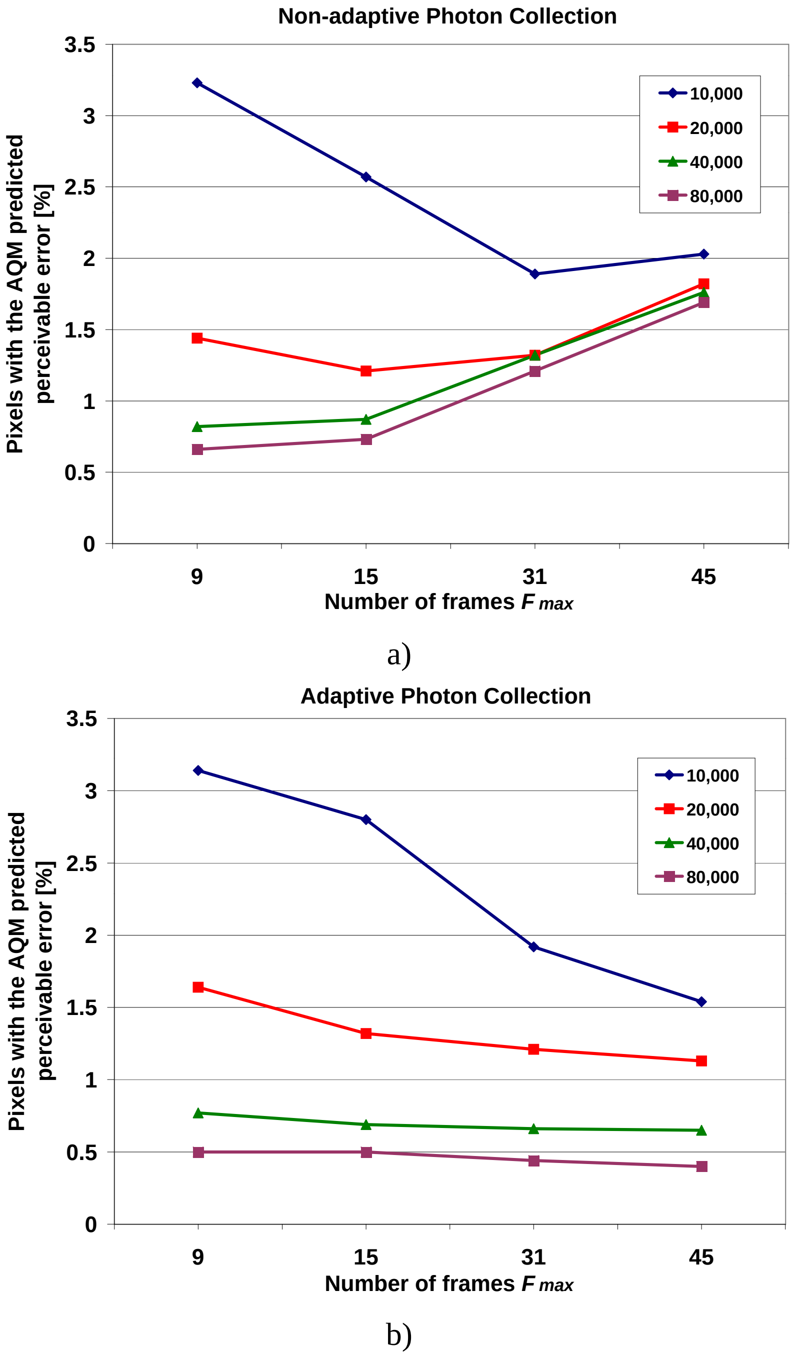


Figure 5.3: Example frame from the ROOM sequence a) reference solution for 2,000,000 photons without temporal processing, b) non-adaptive (note the washed-out highlight under the plane) and c) adaptive photon collection in the temporal domain for $F_{\max} = 30$ frames and $N_{\text{frame}} = 40,000$ photons, and d) the map of AQM predicted perceivable differences (marked in red) between a) and c).



a)

b)

Figure 5.4: The AQM predicted percentage of pixels d_{AQM} with perceivable differences for a) non-adaptive and b) adaptive temporal photon collection approaches for increasing $F_{\max}$ and various settings of N_{frame} (as specified in the legend).

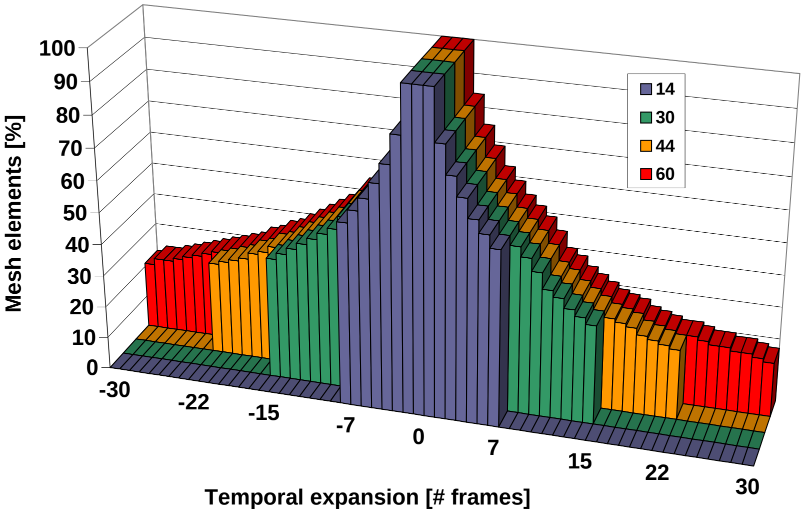


Figure 5.5: Distribution of mesh elements for frame K_2 as a function of the number of preceding (negative values) and following frames for which temporal photon processing was possible. Various $F_{\max}$ were considered as specified in the legend.

was chosen (refer to Table 5.1), in which case $p_{\mathrm{me}} = 36\%$ and $p_{\mathrm{me}} = 30\%$ were obtained for the preceding and following directions, respectively.

Figure 5.6 summarizes the results obtained using the perception-based procedure of noise level estimation as described in Section 5.4.3. It was assumed that $d_{\mathrm{thr}} = 3\%$, which means in practice that the perceivable differences $d_{\mathrm{AQM}} < 1\%$ with respect to the reference solution are usually obtained. Table 5.1 summarizes the number of photons N_{frame} chosen for every segment based on the graphs in Figure 5.6. For such animation settings the spatial filtering described in Section 5.4.5 was not necessary.

Figure 5.7a shows an animation frame obtained using the settings presented in Table 5.1. Figure 5.7b depicts the corresponding frame obtained using the traditional approach without any temporal photon processing. Strong temporal aliasing was observed when the animation composed of such quality frames was viewed. We also tried the traditional approach with $N_{\mathrm{frame}} = 845,000$ which corresponds to the average number of photons collected in the temporal domain using our approach. While the static image is of a quality comparable to the frame in Figure 5.7a some temporal aliasing can be seen clearly when the resulting

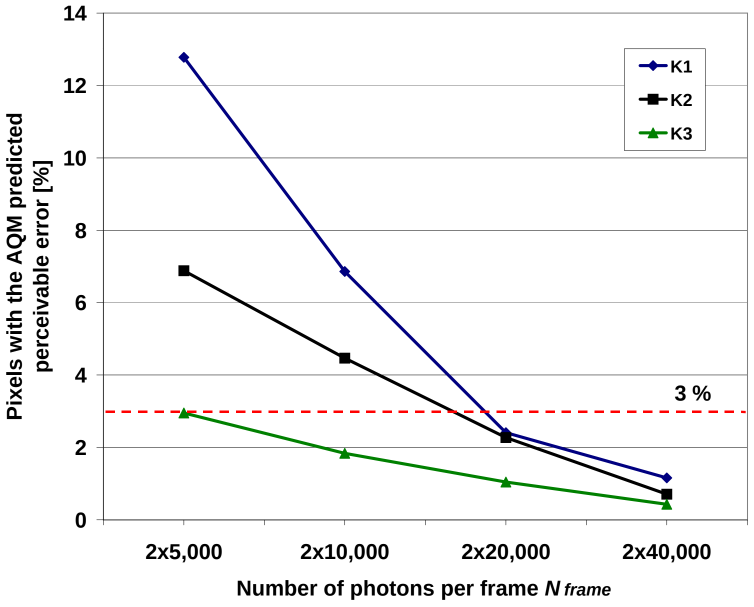


Figure 5.6: The AQM predicted percentage of pixels d_{AQM} with perceivable differences as a function of N_{frame} for the central segment frames K_i.

animation is viewed.

The results obtained for the ATRIUM scene are very similar to the ones for ROOM. For a majority of segments $N_{\mathrm{frame}} = 20,000$ and $F_{\max} = 44$ were chosen using our automatic procedures described in Section 5.4. In general, N_{frame} fell into the range of 10,000–40,000 photons while $F_{\max}$ lay between 30 and 44 frames. The activation of spatial filtering (Section 5.4.5) led to some minor improvement of the animation quality. Figure 5.8 shows an example of a frame with spatial filtering and the only differences with respect to the corresponding frame without filtering can be seen on the stairs which feature small mesh elements and little visual masking.

A summary of timings of indirect lighting computation is given in Table 5.2. As can be seen, tracing photons and their temporal processing is rather inexpensive. The higher cost of temporal processing for the ATRIUM scene is due to the larger number of processed mesh elements. The I/O costs related to disk access are given for two extreme scenarios denoted in Table 5.2 as MIN and MAX (provided in brackets). In the former case photons are accessed from the disk only once and then cached until all frames using those photons have been processed. This means that all photons processed for a given frame must reside in memory, which is a common requirement for many photon-based techniques such as the

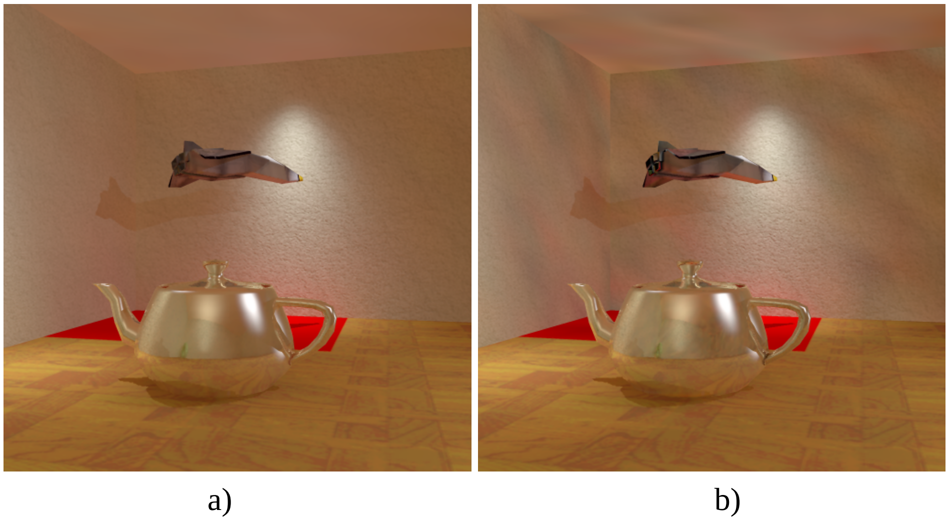


Figure 5.7: Example frame from the ROOM sequence a) with temporal processing for $N_{\text{frame}} = 10,000$ and $F_{\max} = 44$ and b) without temporal processing for $N_{\text{frame}} = 24,550$. The same computation time was spent to generate those frames, but in b) the average overhead time of 1.31 seconds, which was required in a) for the temporal, AQM, and I/O processing (refer to Table 5.2), was used to trace more photons.

photon map [32]. This is usually a reasonable assumption for our technique as well. For example, the maximum memory storage per frame for the ATRIUM sequence was about 120 MB (one photon hit point requires 20 bytes of storage). In the case denoted by (MAX) in Table 5.2 it was assumed that all photons are always loaded from disk for each frame. Such a situation may arise for complex scenes when a high accuracy of lighting reconstruction is required, in which case a large number of photons that are bucketed into a fine mesh must be considered (refer to Section 5.5). Note that even in such a case our timings are at least three times better than shooting a similar number of photons that we collected in the temporal domain for every frame, which requires 87 and 133 seconds for the ROOM and ATRIUM scenes, respectively. Ray tracing of a single frame requires 9.4 seconds for ROOM and 158 seconds for ATRIUM.

5.7 Conclusions

We proposed a novel global illumination technique for dynamic environments which is suitable for high-quality animation rendering. A combination of efficient energy- and perception-based error metrics was used to guide the computation as

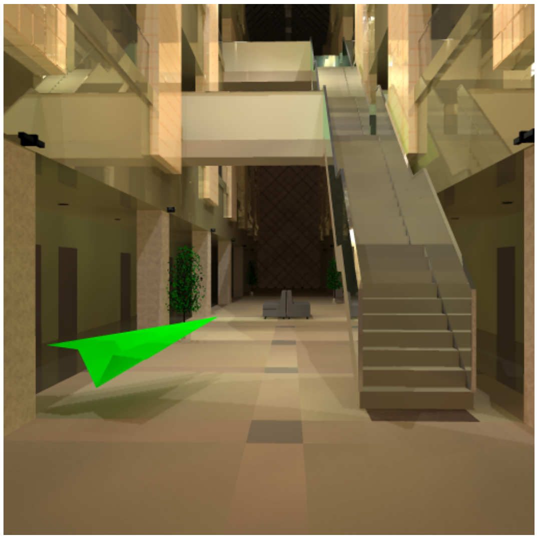


Figure 5.8: Example frame from the ATRIUM sequence with temporal processing and spatial filtering for $N_{frame} = 20,000$ and $F_{max} = 44$.

Scene	Photon tracing	AQM	Temp. proc.	I/O MIN (MAX)	Total MIN
ROOM	2.57	0.27	0.32	0.72 (21.56)	3.88
ATRIUM	2.95	0.21	1.85	0.88 (26.93)	5.89

Table 5.2: Timings of the indirect lighting computation for a single frame obtained as the average cost per frame for the whole animation. All timings are given in seconds and were measured on a 800 MHz Pentium III processor.

a function of local spatiotemporal variations of the lighting distribution. As a result the animation quality as perceived by a human observer is consistent across all frames both in spatial and temporal dimensions. Also, the efficiency of computation is improved and the temporal aliasing is reduced with respect to traditional approaches which ignore temporal processing.

As future work we want to investigate our technique in the context of MPEG coding. The accuracy of the lighting reconstruction can be adjusted in order to obtain a degradation of the animation quality that is perceived as being as homogeneous as possible for an assumed animation compression level. Also, by removing non-visible lighting details from animations the compression performance can be improved.

CHAPTER 6

Local Update of Global Illumination in Final Gathering

Rendering of high quality animations with global illumination effects is computationally expensive using traditional techniques designed for static scenes. In this chapter, we present an extension of the photon mapping algorithm to handle dynamic environments. First, for each animation segment, the static irradiance cache is computed only once for a scene with all dynamic objects removed. Then, for each frame, the dynamic objects are inserted and the irradiance cache is updated locally in the scene regions whose lighting is strongly affected by the objects. In the remaining scene regions, the photon map is used to correct the irradiance values in the static cache. As a result, the overall animation rendering efficiency is significantly improved and the temporal aliasing is reduced.

6.1 Introduction

The algorithm of choosing the final rendering for high quality images is called *final gathering* [57, 42, 67, 11] (also refer to Section 3.4 for more details). Usually, the direct lighting is computed for a surface region represented by a given pixel, and the indirect lighting is obtained through the integration of incoming radiances. The cost of those computation is very high. Those computational cost can be reduced by using the *irradiance cache* data structures [91, 90] to store irradiance samples sparsely in the object space. The cached values are used to interpolate the indirect lighting for each pixel and are computed lazily. The coarse distribution of lighting, which is used for the irradiance integration, can be computed in the preprocessing stage using a deterministic or stochastic radiosity [65], photon maps [33], and so on. The irradiance cache technique efficiently removes shading artifacts which are very difficult to avoid if the indirect lighting is directly reconstructed based on the radiosity mesh or the photon maps. However, this high quality lighting reconstruction needs long computation time mostly because of the irradiance integration that is repeated for many sample points in a scene.

In this work, we extend the concept of irradiance cache for dynamic environments to improve the rendering performance and reduce the temporal aliasing. In addition, we extend the photon mapping algorithm to obtain more efficient global illumination computation for such dynamic environments.

The material in this chapter is organized as follows. In Section 6.2, we present our extensions to the photon mapping and irradiance cache algorithms to handle animation sequences efficiently. In Section 6.3, we discuss our solutions to combat temporal aliasing. In Section 6.4, we present the results obtained using our techniques, and we conclude this chapter in Section 6.5.

6.2 Algorithm

In our approach we use a two pass photon mapping algorithm [33]. In the first pass, photons are traced from light sources and stored in the photon map. In the rendering pass, the lighting computation is performed separately for direct illumination and glossy/specular reflection, diffuse indirect illumination, and caustics. The two first lighting components are computed for each frame from scratch using ray tracing. The diffuse indirect illumination is computed using the irradiance cache [91, 90]. Illumination values stored in the cache are computed through integration of the scene illumination, which is reconstructed from the photon map using the nearest neighbor density estimation technique. Such an integration is not performed to render direct caustics, which are directly reconstructed through density estimation of the caustic photon map.

In this chapter we focus on exploiting the temporal coherence of photons to speedup the costly irradiance cache computation and to improve the quality of indirect lighting reconstruction. We introduce the notion of *static irradiance cache*, which is computed once for an animation segment. For the static irradiance cache computation we remove all dynamic objects (i.e., objects changing their position, shape or light reflectance properties as a function of time) from the scene and we trace the so-called *static photons*.

The illumination component reconstructed from the static irradiance cache is perfectly coherent in the temporal domain and results in the flicker-free animations. However, the dynamic illumination component caused by dynamic objects must be also considered. For this purpose the *dynamic photons* which interact with dynamic objects are computed for each frame and are stored in a separate photon map. The map may store photons with negative energy [34], which are needed to compensate for occlusions of the static parts of the scene by dynamic objects. For example, the negative photons are stored in the regions of indirect

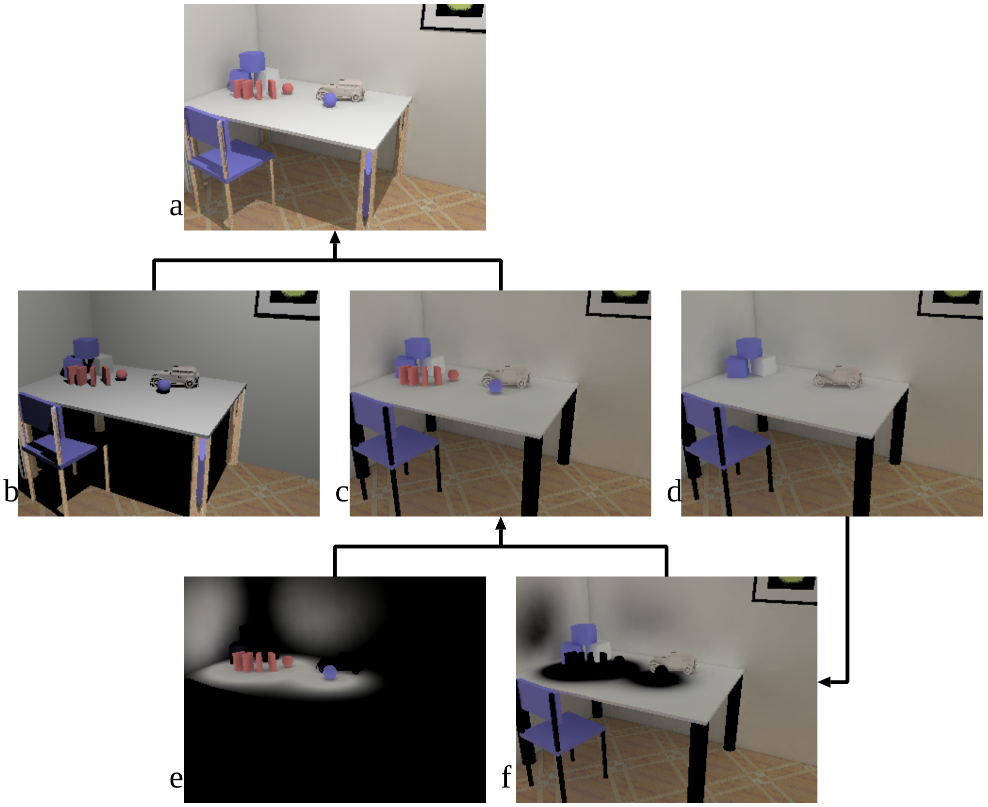


Figure 6.1: Processing flow in the computation of global illumination for an animation frame: a) the final frame, b) direct lighting, c) indirect lighting, d) static indirect lighting computed using the static irradiance cache, e) dynamic indirect lighting computed using the dynamic irradiance cache, and f) the dynamic indirect lighting computed through the photon density estimation and summed with the static indirect lighting which is shown in d).

shadows cast by dynamic objects. We describe the dynamic photons approach in more detail in Section 6.2.1.

Figures 6.1 illustrate the illumination reconstruction using our technique. Figure 6.1a shows the final animation frame whose lighting was composed from the direct illumination and specular reflection (Figure 6.1b) and the diffuse indirect illumination (Figure 6.1c). The dynamic component of the indirect lighting is reconstructed at two levels of accuracy depending on the influence of dynamic objects on local scene illumination. In the regions with the higher influence (we discuss the problem of detecting such regions in Section 6.2.3) the dynamic irradiance cache is recomputed, which leads to better accuracy of reconstructed dynamic lighting (refer to Figure 6.1e). In the remaining scene regions as shown in Figure 6.1f the illumination stored in the static irradiance cache (shown in Figure 6.1d) is corrected by adding its dynamic component reconstructed from the dynamic photon map using density estimation. A direct visualization of the dynamic illumination component is not shown because its values are rather small for the most parts of the scene and are negative in the regions occluded by dynamic objects. We blend lighting reconstructed using those two different methods (Figures 6.1e and 6.1f) to assure smooth transition of the resulting lighting.

While the static photons processing is performed in the same way as tracing ordinary photons for the static scenes [33], the dynamic photons are treated differently which we describe in the following section. In Section 6.2.2 we discuss our extensions of the irradiance cache to handle dynamic and static illumination components. We describe our animation rendering algorithm in Section 6.2.3.

6.2.1 Tracing Dynamic Photons

We introduce the dynamic photon map which is an extension to the photon maps to handle dynamic environments. It is leveraged to estimate the indirect illumination contributed only by dynamic objects (i.e., objects changing the position, orientation, shape, and material properties). The dynamic photon map stores dynamic photons which intersect with dynamic objects at least once.

Figure 6.2 illustrates the paths of two dynamic photons in the room with two dynamic objects, which are shown as an ellipsoid and a circle. The dynamic photons are traced from light sources toward the scene as in the traditional photon tracing approach [33]. When a photon hits a dynamic object, the ray is marked as a dynamic ray, and then it is reflected or transmitted with the positive energy, or simply absorbed. In the first intersection with the dynamic object, a negative ray is spawned at the intersection point. This ray pierces the dynamic object and it is traced further as an usual ray with the only difference that it carries negative energy (in Figure 6.2 all negative rays are depicted using dashed lines). When a dynamic photon hits on a diffuse surface, it is stored in the dynamic photon map.

Special care is required in handling the direct photon paths from light sources to the scene. Since we separately compute the direct illumination, a dynamic photon directly hitting an object is not stored in the dynamic photon map to avoid doubling the computation of the direct illumination (refer to **points a**, **b** and **g** in Figure 6.2). Also, for the negative rays their intersection with dynamic objects must be ignored because their purpose is to subtract the energy from the static irradiance cache, which is computed for the scene without dynamic objects. For example, note that the ellipsoid is simply ignored on the way of the negative ray which travels between a pair of **points h** and **i**. Finally, note that the negative ray is not spawned at **point e** when the dynamic ray hits on the another dynamic object. This can be interpreted as the redirection of energy which in the case of static scene traveled along the path between **points b**, **c**, and **d** into the new path between **points a**, **e**, and **f** in the complete scene with dynamic objects.

The key point of construction of the dynamic photon map is to store only those photons which indirectly intersect with a dynamic object at least once.

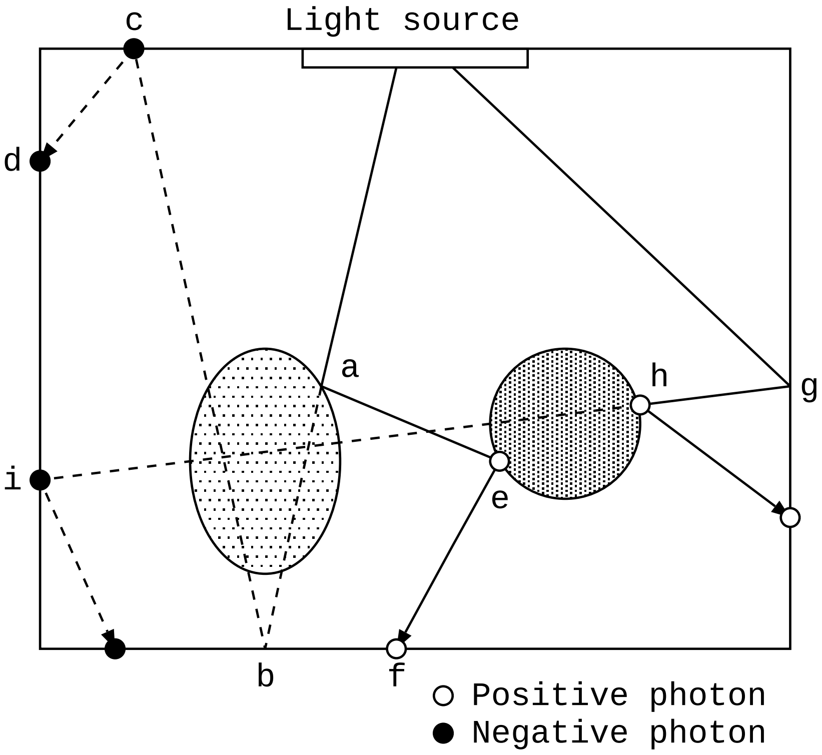


Figure 6.2: Tracing the dynamic photons: Example photon paths.

6.2.2 Static and Dynamic Irradiance Caches

We also extend the irradiance cache to handle dynamic environments. We use two (static and dynamic) irradiance caches for the efficient rendering of indirect illumination in dynamic environments. The static irradiance cache is computed for the scene with static objects only, i.e., all dynamic objects are removed. This cache can be computed only once for an animation sequence when lighting conditions do not change significantly. Otherwise, the animation must be split into shorter segments with coherent illumination. For example, a new static irradiance cache must be computed when a light is turned on in the dark room.

When the camera path is known in advance, which is usually the case for the off-line animation rendering, all values in the irradiance cache can be pre-computed. It is also possible to lazily reconstruct the static irradiance cache in a frame-by-frame manner. However, in such a case all photons for the static scene must be kept in memory.

On the other hand, the dynamic irradiance cache is always lazily recomputed for each frame from scratch for those scene regions whose illumination changes significantly (e.g. for dynamic objects themselves and their neighborhood). The irradiance in the dynamic irradiance cache is computed using the global photon map of the current frame for the complete scene with static and dynamic objects. This approach is similar to the traditional irradiance cache technique with the only difference that in our approach the computation is not performed for the whole scene but rather for its selected regions. The problem of identifying such regions is discussed in detail in the following section.

Note that the sample locations in the dynamic irradiance cache are different for each frame, and they depend on the dynamic changes of illumination. By contrast, the sample locations in the static irradiance cache are the same for each considered animation segment.

Figure 6.3 shows the locations of sample points in the static and dynamic irradiance caches for the scene depicted in Figure 6.4.

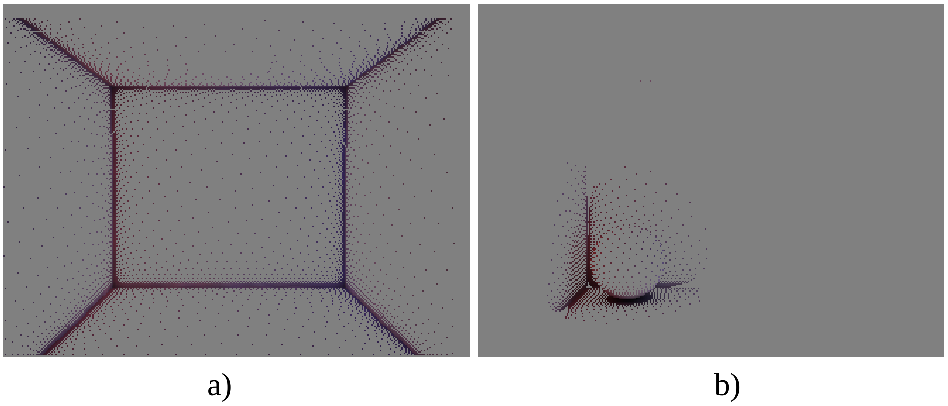


Figure 6.3: Locations of sample points for the a) static and b) dynamic irradiance caches.

6.2.3 Rendering

A practical two-pass rendering algorithm using photon maps is presented in [33]. In this algorithm the outgoing radiance L_r is computed as a sum of four terms: direct illumination L_d, soft indirect illumination L_i, caustics L_c, and specular component L_s:

$$L_r = L_d + L_i + L_c + L_s$$

In this chapter we focus on the computation of indirect illumination L_i, which is usually quite expensive to compute. We further split L_i into two components: L_y which is strongly affected by dynamic objects and L_t which is less affected by them. The choice between L_y and L_t depends on the influence I of dynamic objects on changes of illumination in the scene as follows:

$$L_i = \begin{cases} L_y & I > \tau_u, \\ f(g(I)) * L_y + & \tau_u > I > \tau_l, \\ \quad (1 - f(g(I))) * L_t & \\ L_t & \tau_l > I. \end{cases}$$

where τ_l and τ_u are the lower and upper threshold values for I, and $f(g(I))$ is a blending function between L_y and L_t in the transition scene regions. The blending function must be introduced to avoid discontinuities in lighting distribution due to inaccuracies in L_y and L_t estimates. In particular, the dynamic component of L_t which is computed using density estimation is prone to such inaccuracies. We use a cubic equation as the blending function $f(x)$ and the scaling function $g(x)$ to map the influence values I into the range from 0 to 1:

$$\begin{aligned} f(x) &= -2x^3 + 3x^2 \\ g(x) &= \frac{x - \tau_l}{\tau_u - \tau_l} \end{aligned}$$

The influence I is computed using the following density estimation equation based on the dynamic photon map:

$$I = \frac{1}{\pi r^2} \sum_{p=1}^{N} max(|\Delta\Phi_{p,r}|, |\Delta\Phi_{p,g}|, |\Delta\Phi_{p,b}|)$$

where $\Delta\Phi_p$ is a power of the photon p, and r, g and b denote the red, green and blue components in the power of p. The absolute value of photon energy is used because the dynamic photon map contains photons with both positive and negative energy.

L_y is computed using the dynamic irradiance cache as described in the previous section for the scene regions with $I > \tau_l$ and the dynamic objects themselves. Because such regions are strongly affected by dynamic objects, we use an accurate method for the lighting computation.

To compute L_t, first all irradiances of the static irradiance cache are updated by adding the differential power estimated by the density estimation of the dynamic photon map. We apply the density estimation only in the positions of the

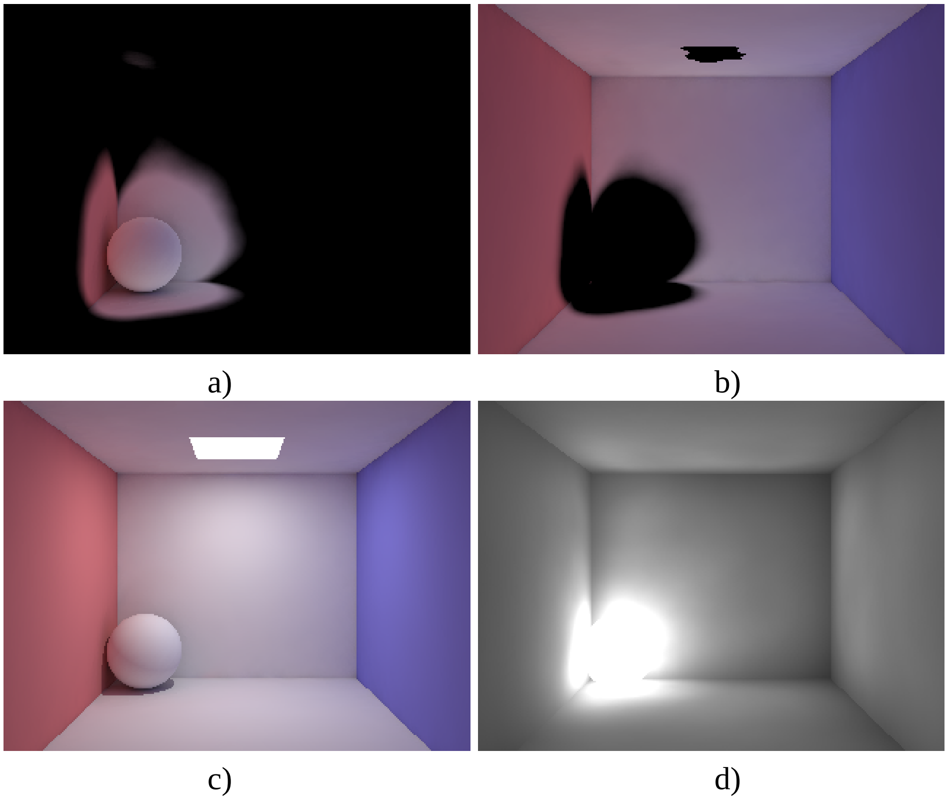


Figure 6.4: Decomposition of indirect illumination into a) L_y and b) L_t components, and c) the resulting image. The corresponding influence function I is shown in d).

static irradiance cache because we assume that in the scene regions less affected by dynamic objects we do not need to recompute cache positions. Then L_t is computed using interpolation of sample values stored in the updated static irradiance cache.

Figures 6.4 illustrate the concept of splitting the indirect illumination into two localized in the scene components L_y (Figure 6.4a) and L_t (Figure 6.4b). The final image (Figure 6.4c) is obtained by combining L_y and L_t with the direct illumination. The influence function I is shown in Figure 6.4c, where brighter regions correspond to higher values of I.

6.3 Temporal Consideration

It is very important to reduce temporal aliasing in rendered animation frames because the human observer is very sensitive to such artifacts. The stochastic noise, which is inherent in Monte Carlo rendering, may result in intolerable degradation of animation quality if proper temporal antialiasing techniques are not used. The photon mapping approach usually successfully eliminates the high spatial frequency noise in the reconstructed lighting distribution. However, the remaining low frequency noise can be still quite objectionable in the context of animation.

We reduce the aliasing problem by introducing the concept of static irradiance cache, which means that the static part of illumination in the scene is perfectly coherent for the subsequent animation frames. For the dynamic illumination component we use two simple techniques applied to the L_y and L_t computation.

The dynamic irradiance cache used to derive L_y is computed by gathering incoming radiances, i.e., shooting a number of random rays into the scene. The noise in L_y can be substantially reduced by fixing the directions of gathering rays. This simple solution works well for the pseudo-random and stratified sampling.

Our second antialiasing technique deals with shooting dynamic photons. Because illumination changes in L_t are computed through density estimation of the dynamic photon map, ideally those photons should be coherent in the temporal domain as much as possible. One way to achieve this goal is to use quasi random walk for photon tracing. Then, the index of the quasi random sequence should be reset for each frame. Of course, the resulting photon paths are not exactly the same because the presence of dynamic objects in the scene but at least should be quite similar. Another solution is to process photons in the time domain as in [51].

6.4 Results

We experimented with animation rendering for scenes BALL and TABLE (refer to Figures 6.4 and 6.5, respectively).

Since the most time consuming part of the animation rendering is the irradiance cache computation we estimated the number of recomputed irradiance samples per frame (refer to Table 6.1). The reference solution recomputes the irradiance samples for all visible scene regions of each frame from scratch. On the other hand, our method recomputes them only for the region strongly affected by dynamic objects. Our method usually requires 3–4 times less irradiance samples per frame than the reference solution.

	Reference	Our method
Scene1	6,719	1,543
Scene2	6,081	1,635

Table 6.1: The number of recomputed irradiance samples per frame.

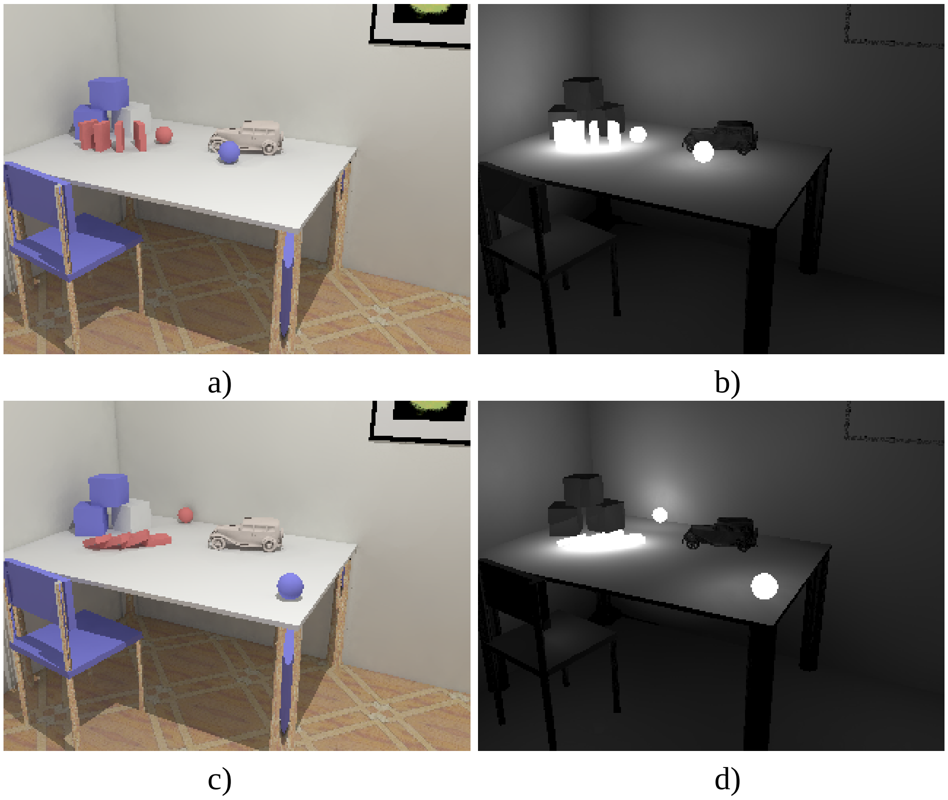


Figure 6.5: Example animation frames a) and c), and the corresponding influence I b) and d).

Table 6.2 compares the rendering time of indirect illumination per frame for our method and the reference solution for scene BALL. The frame resolution is 320×240 pixels. Our method needs extra 1.4 seconds for tracing dynamic photons as well as 1.4 seconds for tracing global photons (shown in the column PT). The computation time for the influence I is rather significant in our approach because we used a large number (300) of nearest photons for its estimation, and a large searching area for every pixel. The timing of L_y is much smaller than in the reference solution. Before computing L_t, we must update the static irradiance cache using density estimation but this takes only 1 second. Table 6.3 compares

	PT	L_i (sec.)		
	(sec.)	I	L_y	L_t
Our method	1.4 + 1.4	16	40	5.5
Reference	1.4	-	205	

Table 6.2: Scene BALL: Timings per frame.

	PT	L_i (sec.)		
	(sec.)	I	L_y	L_t
Our method	6.6 + 33	12	59	54
Reference	6.6	-	224	

Table 6.3: Scene TABLE: Timings per frame.

the timings for scene TABLE. In the experiment performed our method is 1.4 to 3.2 times faster than traditional approach for computation of indirect illumination.

6.5 Conclusions

We presented an efficient technique for high-quality animation rendering. For this purpose we extended the photon mapping approach to dynamic environments. Significant speedup of the computation was achieved by localizing in the scene space the costly recomputation of the irradiance cache. Also, temporal aliasing was reduced by introducing the concept of static irradiance cache which can be reused across many subsequent frames until the scene lighting conditions do not change significantly.

CHAPTER 7

Exploiting Temporal Coherence in Final Gathering

In high quality animation rendering, computationally expensive *final gathering* technique is commonly used (refer to Section 3.4). We extend this technique to the temporal domain by exploiting coherence between the subsequent frames. We store previously computed incoming radiance samples and refresh them evenly in space and time using some aging criteria. The approach is based upon a two-pass photon mapping algorithm with irradiance cache (refer to Sections 2.9 and 2.10), but it can be also applied in other gathering methods. The algorithm significantly reduces the cost of expensive indirect lighting computation and suppresses temporal aliasing with respect to the state of the art frame-by-frame rendering techniques.

7.1 Introduction

The irradiance caches can be reused for the efficient rendering of walkthrough animations in static environments [91]. However, for dynamic environments, such a simple reusing technique may lead to invalid lighting. In this chapter, we address this problem and propose some extensions of irradiance cache management into the temporal domain, specifically in the context of photon mapping technique. Not only do we try to reuse the irradiance cache locations, but also we update the cache values required in dynamic environments. For each frame and each cache location, a certain percentage of stored incoming radiance samples is updated potentially for those scene regions in which lighting changes are the most significant. Additionally, in response to changing lighting and camera positions, new cache locations are lazily inserted and unnecessary cache locations are removed. It reduces computational time significantly compared to frame-by-frame rendering. In addition, better animation quality can be obtained due to the temporal coherence between cache locations and cached incoming radiance samples.

Our space-time approach presented in this chapter focuses only upon efficient computation of soft indirect lighting. Direct illumination and caustics as well as all directional effects such as specular reflections and refractions are computed from scratch for each frame.

In the first stage of the photon mapping algorithm, photons are traced from light sources toward the scene, and photon hit points are registered in a kd-tree structure, called photon map. Because this stage is done very quickly (compared to the second pass), photons can be easily recomputed for each frame. The only attempt at reusing photons for several frames was done to perform motion blur for caustics [5]. In our approach, the photon map is computed from scratch for each frame, which guarantees that selectively updated incoming radiance samples in the final gathering computation are correct.

In the following section, we extend the concept of irradiance cache into the temporal domain, in which case cached irradiance values may change from frame to frame due to dynamic changes in a scene.

7.2 Temporally Coherent Gathering

In this section we describe our approach to updating irradiance cache values for the subsequent animation frames. Our goal is to exploit temporal coherence of incoming radiance samples contributing to each cache value. This requires sharing information on the samples between neighboring frames and selectively replacing those samples that become invalid due to changes in the dynamic environment.

In Section 7.2.1 we describe data structures used for the incoming radiance samples. Then in Section 7.2.2 we present our strategies to update those samples selectively. In Section 7.2.3 we discuss the problem of choosing the ratio of samples to be updated and we propose an algorithm for adaptive selection of such a ratio for each cache.

7.2.1 Cache Data Structures

In the traditional irradiance cache algorithm [91, 90] positioning each cache is a view-dependent process. Then the numerical integration of incoming radiance is performed by tracing rays towards the environment and gathering lighting information. To produce high quality images a huge number of rays is traced, and the incoming radiance samples are immediately discarded after each cache value is computed. In papers [66, 37] those samples are stored and reprojected to improve interpolation between caches.

In this research we show that storing incoming radiance samples is not only feasible, but also offers many advantages for efficient animation rendering. For each sampling direction, we store incoming radiance packed in four bytes using the common exponent method [88] which reduces the storage to one third of that when using the standard floating-point format. The error of the irradiance estimation caused by the inaccuracy of this format is negligible because an irradiance value is the sum of a large number of incoming radiance samples. The distance to the nearest intersection point is stored to reevaluate the harmonic mean distance [91] (the reciprocal of the sum of reciprocal distances) in upper hemispherical directions, which is required to compute optimal cache locations. The age of each sample, which is a function of the number of frames since the sample was computed, is stored in a flag. This value is used to create an approximated probability density function which is used to decide in which order directions should be replaced (refer to Section 7.2.2). To save storage, we use the same flag to indicate whether the sample hits a moving object. This information is used to adaptively estimate the number of rays to be updated (refer to Section 7.2.3). The data structure for the incoming radiance sample is as follows:

```
struct RadianceSample {
  RGBE    Li;   // incoming radiance
                // packed in 4 bytes
  float16 Di;   // distance to the nearest
                // intersection point
                // packed in 2 bytes
  ushort  flag; // number of frames and
                // the flag of hitting on
                // dynamic objects
};
```

Each incoming radiance sample occupies 8 bytes. Usually, 200–1,000 samples per cache are required for a still image to remove artifacts and the number of irradiance values varies in each scene. We store the incoming radiance samples on a hard disk. The overhead of the disk IO is negligible because the samples are accessed only once per frame when they are updated. In Section 7.4 we provide information on storage requirements for our test scenes.

We use the kd-tree data structure to store cache locations in the object space.

7.2.2 Age Driven Cache Update

In this section we describe our solution for updating incoming radiance samples based on their age. For the purpose of illustration we show in Figure 7.1 our

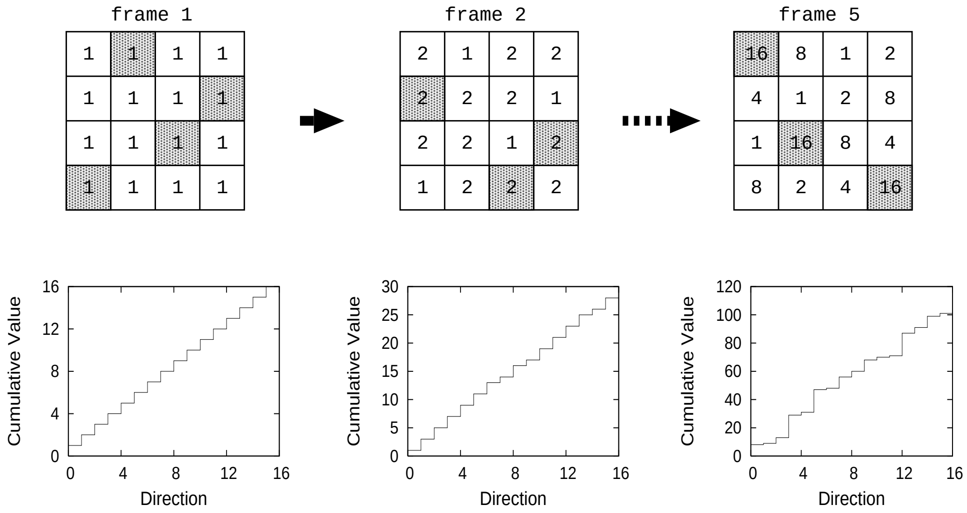


Figure 7.1: Sampling scheme for the irradiance cache. Upper row: Internal cache structure with 16 stratified sampling directions in the upper hemisphere for frames 1, 2, and 5. The values in the grid cells are computed as 2^{age}. Lower row: The corresponding cumulative distribution functions (*c.d.f.*).

sampling scheme for a simple cache, which is composed of 16 samples. The grid depicts the stratified sampling directions over the upper hemisphere. The number in each cell shows the corresponding sample age which is measured using the exponential function 2^{age}, where the age value is stored in the structure `RadianceSample`. The graphs in the bottom row show the corresponding cumulative distribution functions (*c.d.f.*) of the sampling direction based on the value of 2^{age}. This value for the updated cells is reset to zero for the current frame. Because all directions are computed from scratch for *frame0* (the first frame in each animation) the age values are set to 0 for all cells. The age of all cells is increased when the subsequent frame is processed. Shaded cells in the grid in Figure 7.1 indicate the selected cells for which incoming radiance value as well as other records in the structure `RadianceSample` (refer to Section 7.2.1) are updated by shooting a new random ray within the cell for that frame.

Our purpose is to pick a number of sampling directions and replace the old samples by re-shooting the ray for each selected direction. The random permutation algorithm that randomly changes the order of elements is well suited to our goals because we want to randomly pick directions with the constraint that they should not be the same for each frame. When we have m elements $v_0, v_1, \ldots, v_{m-1}$, a random integer value X in $0 \leq X \leq m-1$ is chosen and the last element v_{m-1} is swapped with v_X. This process is repeated for the remaining elements $v_0, v_1, \ldots, v_{m-2}$ until the number of the remaining elements becomes one. Figure 7.2 shows the pseudo-code of the random permutation algorithm.

```
randomperm ()
  elements v[size]
  m = size
  while (m > 1)
    X = uniform random integer number
        in the range of 0 and m-1
    swap v[m-1] and v[X]
    m--
```

Figure 7.2: Pseudo-code of the random permutation algorithm

It is straightforward to apply this algorithm to the elements which have non-uniform probability. Each time a random cell is selected, the *c.d.f.* is rebuilt and the uniform random value X in $0 \leq X \leq total\_cumulative\_value$ is mapped to the cell in the grid.

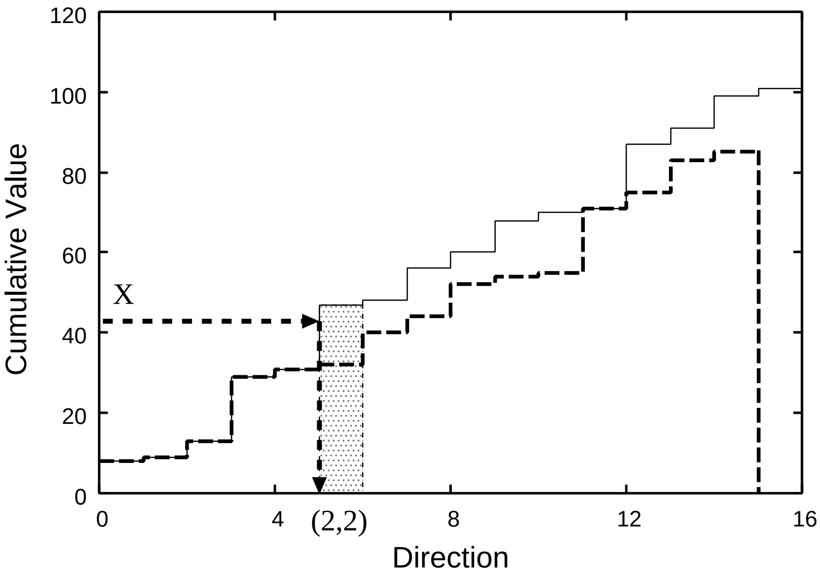


Figure 7.3: A random number X is mapped to the index of a cell (2, 2) and a new *c.d.f.* (the bold dashed line) is rebuilt after the selected cell (the shaded area) is removed.

Figure 7.3 illustrates how the random value is mapped to the index of a cell for frame 5 in Figure 7.1. Because the *c.d.f.* is already sorted, the binary search can be efficiently exploited for this complete balanced data. After one cell is selected, this cell is excluded from the *c.d.f.* and a new *c.d.f.* is built for the remaining cells as shown in the bold dashed line in Figure 7.3. This process is repeated until a sufficient number of directions are chosen. The pseudo-code in Figure 7.4 summarizes the overall algorithm.

This scheme makes sample directions uniformly distributed in space and time. The recently selected cells are less likely to be selected than others in the subsequent frames.

```
render_animation ()
  render the first frame using traditional
  irradiance cache
  for all remaining frames
    photon tracing
    update_irradiance_cache()
    render the current frame

update_irradiance_cache ()
  for every irradiance cache E
    create a cdf
    n = the number of updated samples
        (refer to Equation (2))
    update_incoming_samples(E, cdf, n)

update_incoming_samples (E, cdf, n)
  while (n > 0)
    pick a cell based on the cdf
    shoot a gathering ray to the cell
    delete the entry of the selected cell
    from the cdf and rebuild it
    n--
  evaluate irradiance values
```

Figure 7.4: Pseudo-code of the overall algorithm

7.2.3 Adaptive Cache Update

So far we described in which order directions should be replaced. Another question is how many rays should be replaced for each cache. Intuitively, this should depend on the magnitude of changes in lighting: More rays should be replaced when and where lighting changes quickly and less rays may be enough for slowly changing environments. We tried both uniform and adaptive number of replacement rays. The uniform number approach is very intuitive and leads to refreshing the same number of rays for every cache. For example, when we set the number of replacement rays to 10% of the total number of strata, the rendering speed is roughly ten times faster. Moreover, it ensures that every direction is likely to be refreshed after about 10 frames (the reciprocal of 10%). This means that temporal error propagation by reusing invalid samples for more than 10 frames is not very likely.

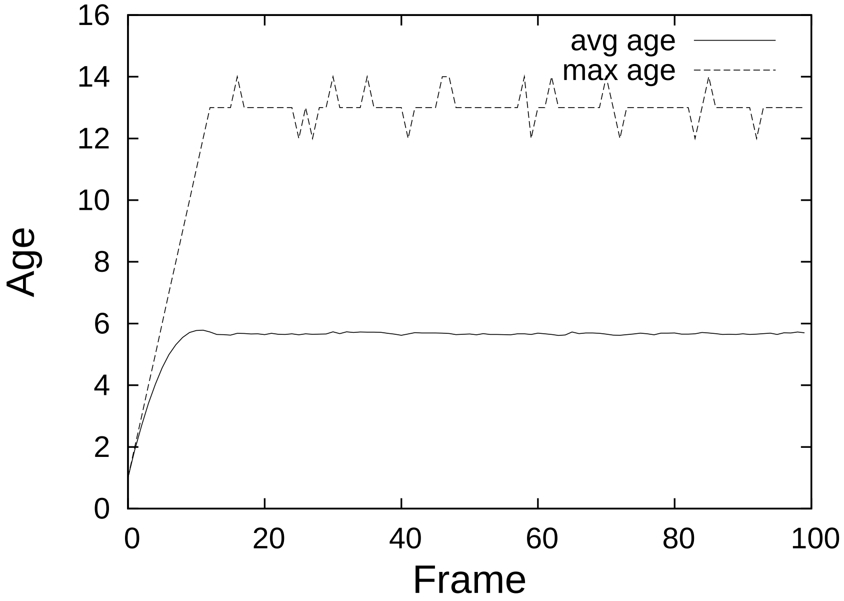


Figure 7.5: The average and maximum age of samples measured for an animation composed of 100 frames. The total number of considered sample directions for each cache was 500 and 10% of samples were replaced for each frame. The average age is about 5.6 frames and the maximum age is 14 frames.

Figure 7.5 shows the graph of the average and maximum age of all directions when 10% of samples is replaced. We assumed that 500 directions are stored for each cache and 50 directions are updated per frame. The average age of about 5.6 frames is obtained experimentally for an animation composed of 100 frames and it differs only slightly from the theoretical average age 5.5 frames computed as:

$$\frac{50\sum_{x=1}^{10} x}{500} = 5.5 \tag{7.1}$$

The graph of the maximum age indicates that every ray is refreshed at most after 14 frames compared with the optimal 10 frames case. These experimental data show that our algorithm works well in practice.

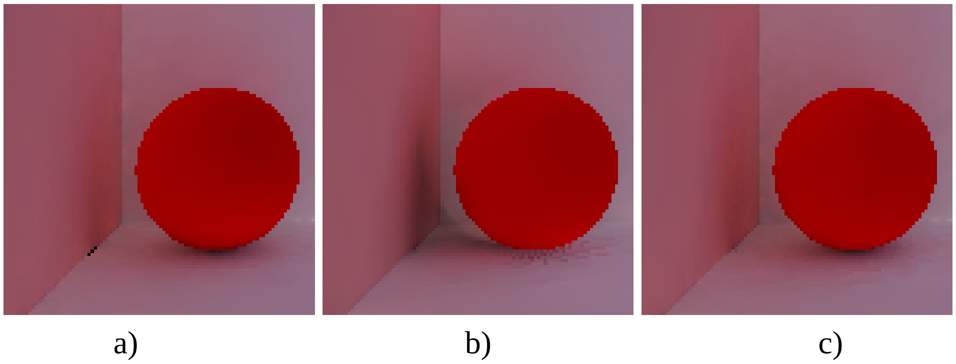


Figure 7.6: Adaptive cache update for scene regions in which indirect illumination changes rapidly: a) reference frame-by-frame solution, and temporal update solutions for b) uniform and c) adaptive number of refreshing rays. Notice the incorrect shadow which is cast by the ball in the corner in b). This shadow mostly disappears in c).

The reason of the low average and maximum age at the leftmost part of the graph is that all directions at *frame0* are computed from scratch.

We also tried an empirical adaptive approach which adjusts the number of the replacement rays based on the number of rays which hit dynamic objects. We estimate the number of directions to be updated as:

$$N_{update}(x) = (\tau_{\max} - \tau_{\min}) \cdot x + \tau_{\min} \cdot N \qquad (7.2)$$

where N is the total number of gathering rays, the $\tau_{\min}$ and $\tau_{\max}$ are user specified percentages of the minimum and the maximum of updated rays and x is the number of rays which hit on the dynamic objects. This simple empirical strategy works very well, especially in the scene regions near moving objects. Figure 7.6 shows an example where a red ball leaves the corner and is rolling on the floor. Figure 7.6a represents the reference solution of the soft indirect illumination. Figure 7.6b is rendered using a uniform number of refreshing rays for every cache. This solution leads to incorrect results at the corner near the moving red ball due to a too small number of refreshing rays in this region. Figure 7.6c shows the result of the adaptive scheme which is very similar to the reference solution.

7.3 Handling Irradiance Caches

In the previous section we discussed the issues of single cache update and the goal of this section is the problem of cache locations. We focus on two issues specific to our approach that arise when the irradiance cache data structures are reused between frames. In Section 7.3.1 we discuss how to handle caches located on moving objects. Then in Section 7.3.2 we present our solution to remove redundant caches as camera and lighting change.

7.3.1 Transforming Caches on Animating Objects

Separating the irradiance cache data structure from the geometric one is advantageous because optimal cache locations can be selected independently from the geometry. However, it is cumbersome in dynamic scenes because caches on moving objects may no longer lay in the same position on a given surface for subsequent frames. To solve this problem, we store the object ID for each cache. This allows us to move caches to different locations in the next frame. This requires transformation of a local coordinate system for each cache to preserve the directional distribution of incoming radiance samples. Although the sampling coordinates on the moving objects are not precisely the same, artifacts were not visible in our test scenes if a reasonable number of refreshing rays is chosen (e.g., 10% of total gathering rays) even for the scene in which the ball is rolling and the normals on the ball change quickly (refer to Figure 7.6). For fast moving objects a complete update of all incoming radiance samples can be also considered.

7.3.2 Removing Redundant Caches

When objects are moving, the optimal distribution of cache locations is changing. If an object approaches another surface the value of a harmonic mean distance becomes small and the valid domain of the irradiance cache is decreased, which leads to denser cache locations in such regions. Because irradiance cache algorithm has a unique lazy construction mechanism, new caches are simply inserted when valid caches are not found near the query location. The problem occurs when the object moves away from a surface. In this case, the harmonic mean distance of each cache becomes large and the valid cache domain increases. In such regions, some caches become redundant. This problem is illustrated in Figure 7.7a where the rolling red ball leaves many redundant caches along its motion trajectory.

It is difficult to find an optimal layout of caches which completely covers all visible surfaces and leads to a minimal number of caches without affecting image quality. Our solution is inspired by neighborhood-based stratification approach [87] developed in point-based rendering to remove superfluous points. We simply remove caches when too many valid caches are found at some locations. For each cache location, a nearest neighbor search is done. If the number of found neighbors is bigger than some threshold number (for example 10), this cache is removed. Figure 7.7b shows the result of applying this procedure and as can be seen many redundant caches are successfully removed. Updating the data structure for each cache to be removed is not efficient, so at first we mark all redundant caches and hide them for the subsequent cache density queries. After all caches are examined, marked caches are removed and the kd-tree is balanced in a packed heap data structure.

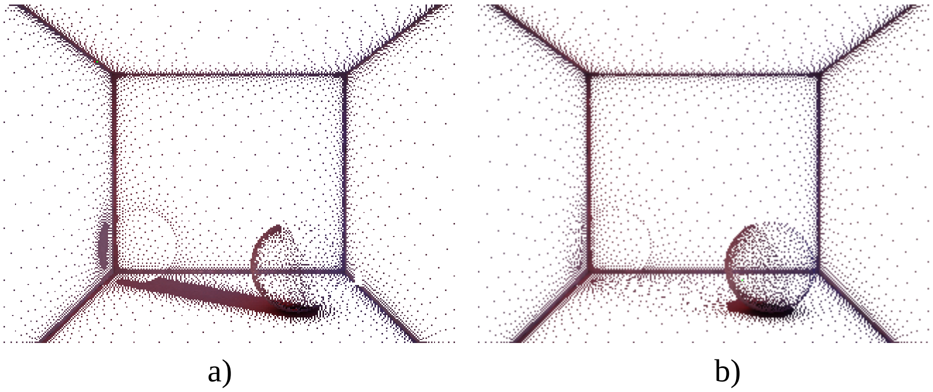


Figure 7.7: Irradiance cache locations: a) redundant irradiance caches resulting from the ball motion on the floor, b) result of removing the redundant caches.

7.4 Results

We tested our algorithm for three different scenes BOX, LIGHT and ROOM (refer to Figures 7.8, 7.9 and 7.10, respectively). For the BOX scene indirect lighting changes significantly in the proximity of regions traversed by the object. As a result of the motion of the red box towards the light source, strong color bleeding effects can be seen on the ceiling. For the LIGHT scene the light source turns toward the left red wall, which results in strong color bleeding from that wall. The ROOM scene is substantially more complex and we consider both the motion of light (sun position) and of objects (rotating fan) simultaneously. In all our test scenes indirect lighting changes quickly and those changes affect either a large portion of the scene, so these test scenes are very difficult cases for our algorithm. We expect that in many practical applications the indirect lighting changes will be more moderate.

The animation sequences were rendered by our experimental renderer on a Pentium 4 Xeon 1.7 GHz, 1 GB memory, Debian GNU/Linux. In both BOX and LIGHT scenes 192 incoming radiance samples are considered for each cache, and 768 samples are used in the ROOM scene. Incoming radiances are refreshed using

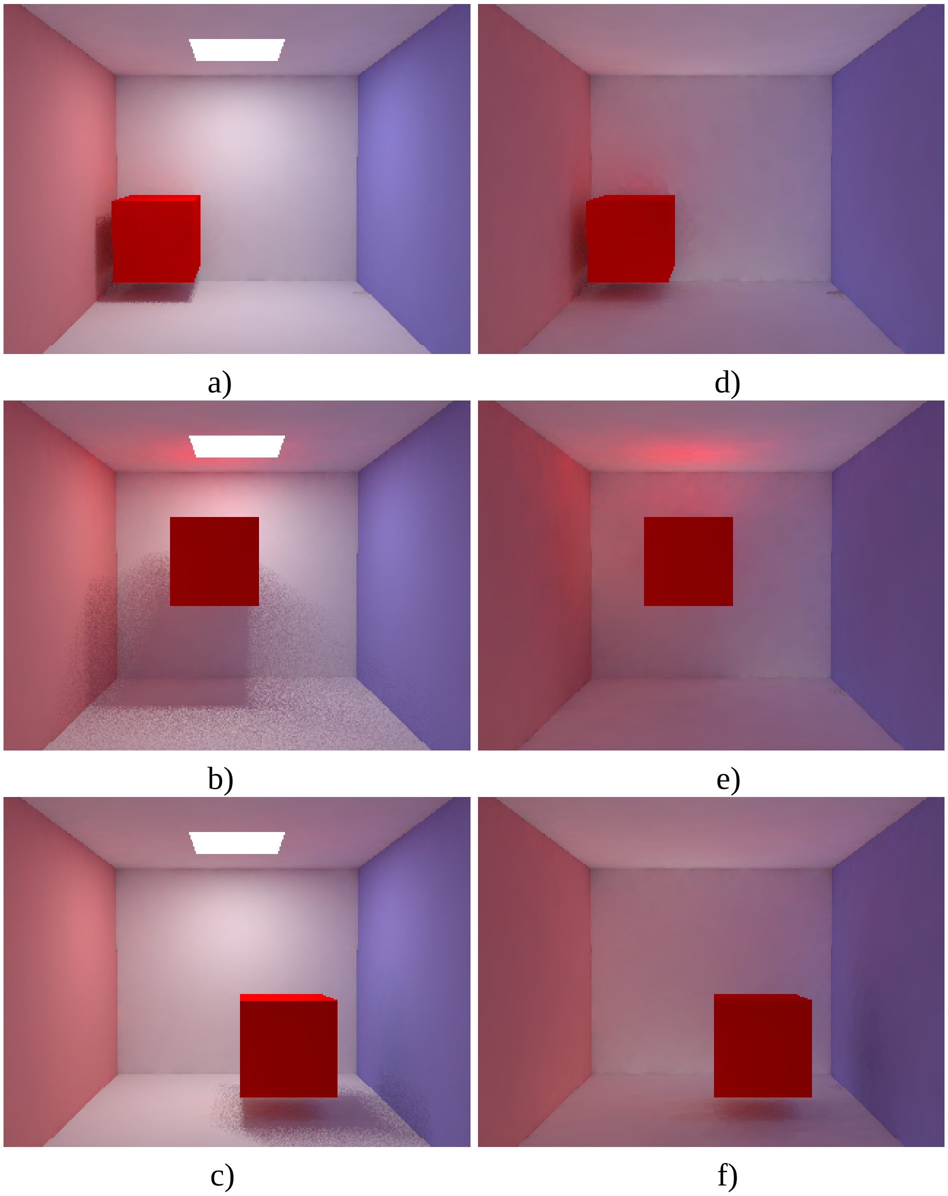


Figure 7.8: Selected frames for the BOX animation sequence. In the left column the final frames are shown while the right column shows the corresponding indirect lighting solutions computed using temporally coherent gathering.

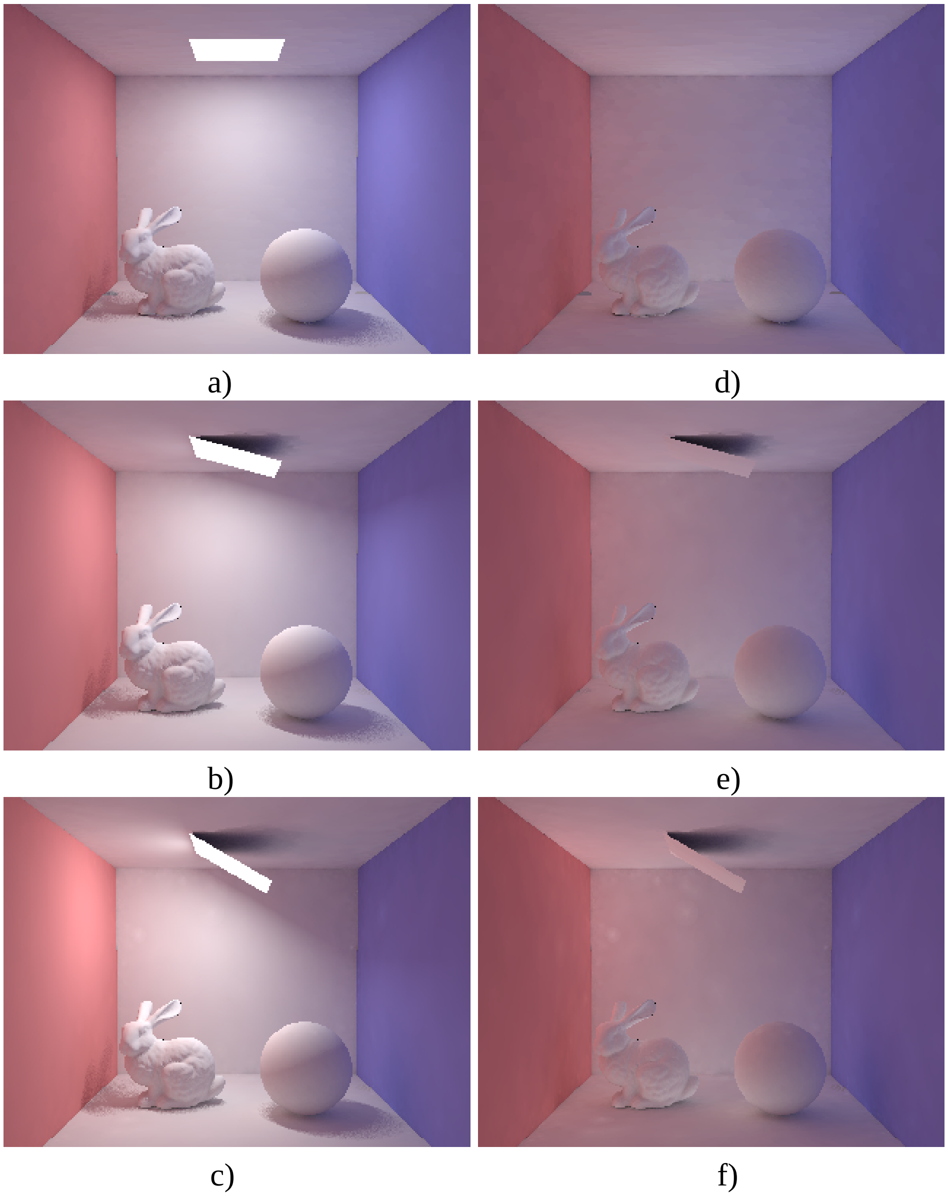
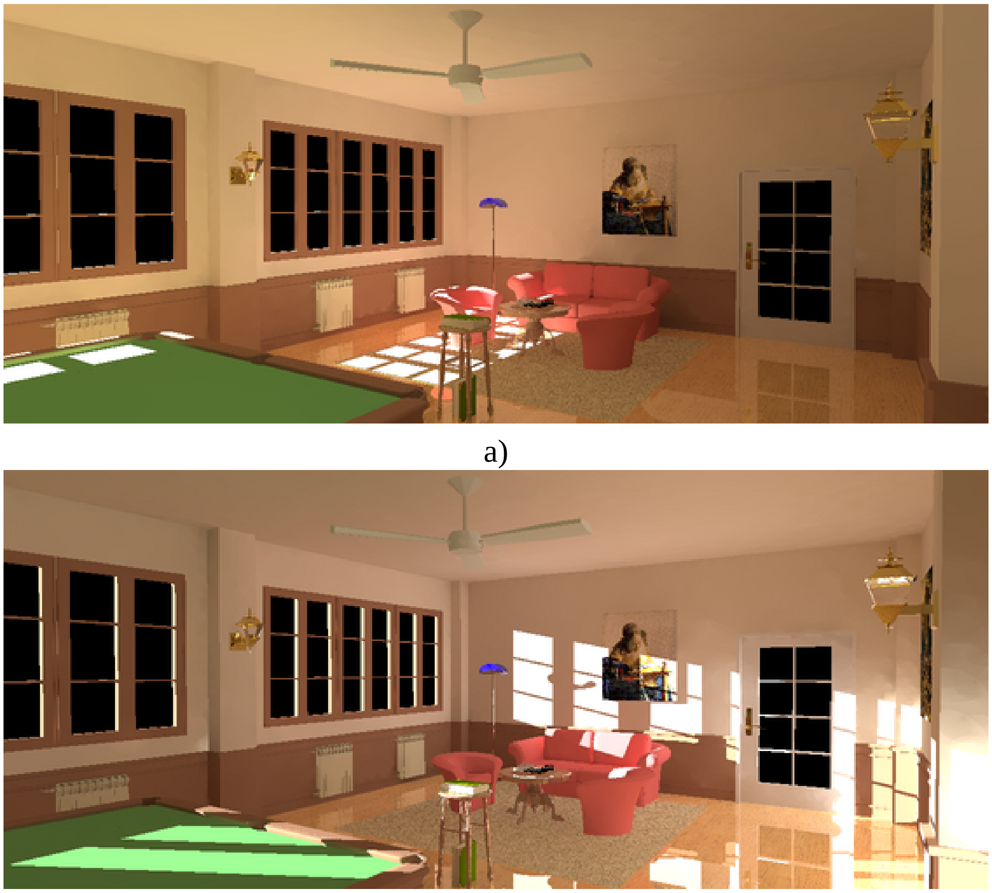


Figure 7.9: Selected frames for the LIGHT animation sequence. In the left column the final frames are shown while the right column shows the corresponding indirect lighting solutions computed using temporally coherent gathering.



a)

b)

Figure 7.10: Selected frames for the ROOM animation sequence.

Scene		T_{pt}	T_d	T_i	T	total
BOX	Ref	21.6	1.0	19.3	41.9	1 h 3 min
	Our	21.6	1.0	6.7 (x2.9)	29.3 (x1.4)	45 min
LIGHT	Ref	24.3	1.1	34.3	59.8	1 h 31 min
	Our	24.4	1.1	6.6 (x5.2)	32.2 (x1.9)	49 min
ROOM	Ref	34.6	11.6	674.6	720.9	18 h 14 min
	Our	34.6	11.6	74.3 (x9.1)	120.8 (x6.0)	3 h 14 min

Table 7.1: Timings for scenes BOX, LIGHT and ROOM shown in Figures 7.8, 7.9 and 7.10, respectively. All timings except the last column are given in seconds. The row "Ref" presents timings of the reference solution, in which all frames are computed independently. The row "Our" presents timings of our solution. The columns show the average time per frame for each component: T_{pt} - photon tracing and precomputation of irradiance [7], T_d - direct illumination, T_i - indirect illumination. The column T is the average time per frame for all components, i.e. $T = T_{pt} + T_d + T_i$. The values in the parentheses show the acceleration factors compared to the corresponding reference solution. The last column shows the total timings for rendering all frames.

the aging method. The $\tau_{\min}$ and $\tau_{\max}$ parameters for adaptive control in Equation (7.2) are set to 0.05 and 1.0, respectively. The image resolution for both animations BOX and LIGHT is 320×240 and the one for ROOM is 564×240.

Detailed timings are shown in Table 7.1. The photon tracing and the precomputation of irradiance [7] (refer to the timings in column T_{pt}) are repeated for each frame. The precomputation of irradiance is time consuming (about 40–60 % of T_{pt}), but it vastly accelerates the computation of the indirect illumination in both the reference solution and ours. So we used this technique to render all animations. Column T_d shows timings for the direct lighting computation which is repeated for each frame from scratch. T_{pt} and T_d are usually a small fraction of the total computation time for complex scenes. We did not optimize our experimental code to speed up those computations. Column T_i shows timings for indirect lighting reconstruction including the overhead of temporal processing. Significant speedup of 3–9 times with respect to the reference frame-by-frame solution was achieved. Column T summarizes the overall processing time per frame. Note that our algorithm performs much better for the more complex ROOM scene.

The number of irradiance samples and the resulting storage requirements are shown in Table 7.2. Since the irradiance cache data are stored in the object space the storage requirements weakly depend on the frame resolution. For example, in the ROOM scene the number of irradiances becomes 1.7 times bigger size when the image resolution is quadrupled. Recently, Gautron et al. [21] demonstrated

Scene	N	#E	Size
BOX	192	4,009	6.2 MB
LIGHT	192	4,377	6.7 MB
ROOM	768	11,599	71.3 MB

Table 7.2: The size of the irradiance cache: N - the number of incoming radiance samples per an irradiance E (refer to Equation 7.2), #E - the number of irradiance values, Size - the storage requirement of incoming radiance samples.

τ_{min}	RMS Error	T_i
10 %	0.5 %	7.3
5 %	0.6 %	6.7
2.5 %	0.8 %	6.1
1.25 %	1.1 %	5.9

Table 7.3: The RMS Error measured in respect to the reference frame-by-frame animation for the BOX scene for various settings of τ_{min} which decides upon the minimal number of updated rays (refer to Equation 7.2). $\tau_{\text{max}} = 100\%$ was assumed. As in Table 7.1 T_i denotes the average time per frame (measured in seconds) for the indirect illumination computation during rendering.

that spherical harmonic bases can be used to store compactly an irradiance field over the hemisphere. We could also consider this approach in our application to reduce the storage costs.

The visual quality of an animation produced by our technique is better than for the reference solution (obtained with traditional irradiance cache) because temporal flickering is significantly reduced. When single frames are compared they look almost perfectly the same as the reference solution (the RMS errors are 0.6%, 0.5% and 2.7% for the BOX, LIGHT and ROOM scenes respectively). All frames are of similar quality because our algorithm does not accumulate error and refreshes all gathering rays. Table 7.3 summarizes changes of the RMS error as a function of τ_{min} for the BOX scene. Similar results have been also obtained for the other test scenes.

Figure 7.11 depicts the values of incident radiance samples over the hemisphere for a selected cache location. The samples are captured in the middle of an animation sequence in order to check whether errors in their value do not accumulate as a function of time. As can be seen the directional distribution of samples is very similar for the frame-by-frame computation and our method. The graphs in Figure 7.12 show changes of the irradiance value as a function of time for both approaches. The irradiance is measured for 90 subsequent frames. The same cache location as in Figure 7.11 is considered. Again, the correspondence between the two graphs is good. A small time lag between the two graphs can be observed because samples are reused only in the “chronological” order.

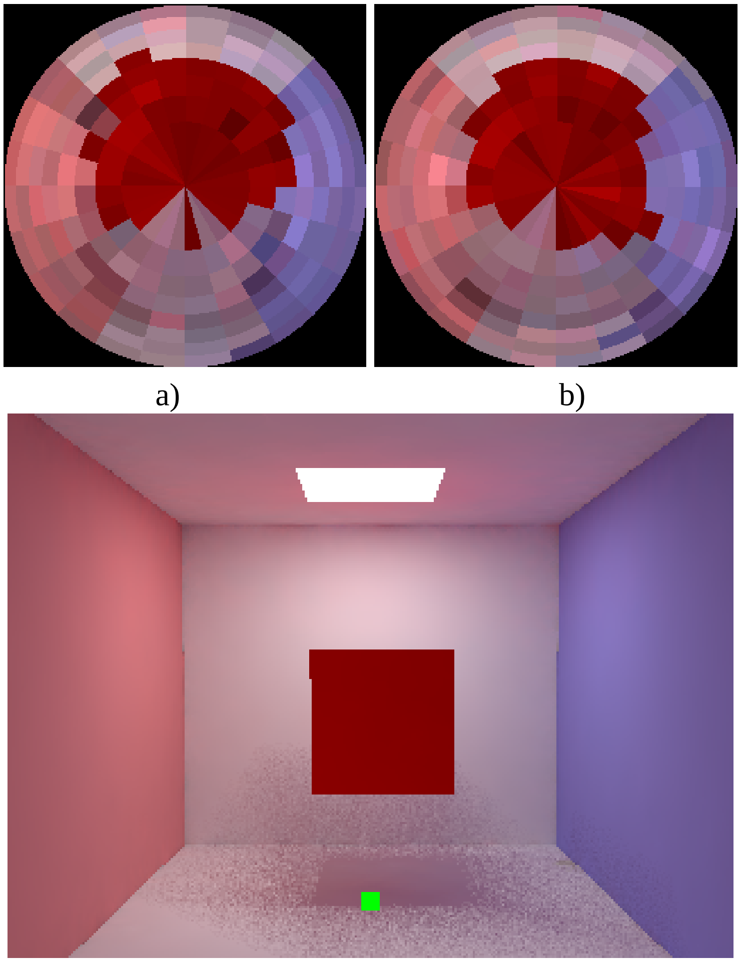


Figure 7.11: Distribution of incoming radiance samples over the hemisphere for a selected cache location at the floor, which is shown as the green dot in the bottom image: a) the frame-by-frame computation and b) our method (10% of samples is refreshed for each frame according to the aging criterion).

7.5 Conclusions

We presented a simple, general, and effective method for computing soft indirect illumination in dynamic scenes. Our algorithm requires to build a cumulative distribution function (*c.d.f.*) for each irradiance cache to decide the order of directional incoming radiance samples updates based on their age. The gathering

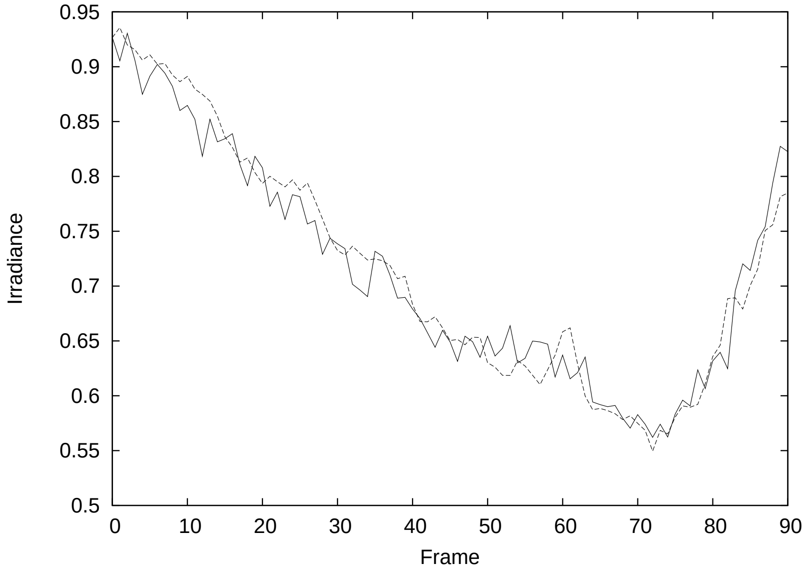


Figure 7.12: Changes of irradiance value as a function of time for the BOX animation at the cache location shown in Figure 7.11. The graph for the frame-by-frame approach is drawn as the solid line, while the dashed line is used for our approach.

rays are updated taking into account the probability in the *c.d.f.* within the limits of user-set minimum and maximum refreshing sampling ratios ($\tau_{\min}$ and $\tau_{\max}$). By contrast to traditional approaches which compute global illumination for every frame from scratch, our algorithm updates a rather small number of incoming radiances. It speeds up expensive indirect illumination computation 3–9 times compared to the currently fastest rendering algorithm. Also, temporal flickering of indirect lighting component is substantially reduced due to the use of temporal coherence. This is achieved by sharing the same cache locations and incoming radiance samples between subsequent frames. Our algorithm can handle general animations including light source motion. Camera animation does not affect cached indirect illumination and new irradiance caches are inserted only when occluded parts become visible for subsequent frames. Our algorithm fits well to the photon mapping algorithm [33] in which direct illumination, caustics, specular reflections and refractions are computed on a frame-by-frame basis while diffuse interreflections can be accelerated using our technique.

As future work we plan to experiment with more compact representations for incoming radiance using wavelets or spherical harmonics [21]. Also, we plan to investigate the applicability of our technique to render moderately glossy surfaces using directional information which we store in our cache data structure. This will require the development of new criteria controlling the density of cache locations, which will be sensitive to surface glossiness. To reduce the number of caches required for high quality rendering of moderately glossy surfaces we plan to use *the final gather reprojection* technique [37].

CHAPTER 8

Importance Sampling in Final Gathering

In this chapter, we propose an efficient algorithm for handling strong secondary light sources within the photon mapping framework. We introduce an additional photon map as an implicit representation of such light sources. At the rendering stage, this map is used for the explicit sampling of strong indirect lighting in a similar way as it is usually performed for primary light sources. Our technique is fully automatic, improves the computational performance, and leads to better image quality than traditional rendering approaches.

8.1 Introduction

A naive sampling with a uniform distribution of sampling directions in a space may lead to very poor convergence of the irradiance integration which manifests in noisy pixels for scenes with significant variations of lighting distribution. An efficient way to improve the performance of the irradiance integration is to sample more densely those scene regions which significantly contribute to illumination at a given cache location. For glossy surfaces, an easy importance sampling scheme can be considered by grouping sample directions around the reflection direction in respect to the eye position. However, for diffuse surfaces, this importance criterion fails. In this chapter, we propose an efficient importance sampling scheme which handles this difficult case. Our solution is embedded into the photon mapping algorithm framework [33].

Slowly changing (soft) indirect lighting is reconstructed from the global photon map through the irradiance cache technique [91, 90, 33] (refer to Section 2.10 for more details). For each cache location, irradiance is integrated over a scene by sampling the incoming energy for selected directions. To reduce the variance of such sampling, the hemisphere of all possible directions is split into strata, and a small number of sample directions (usually one) are randomly chosen for each stratum (refer to Figure 2.2).

This approach works well for scenes with low variation of lighting distribution but leads to a huge number of samples when density of photons in the global photon map significantly changes among scene regions. The angular density of samples should correspond to the density of photons stored in the map. However, the estimation of photon density would require projecting all those photons on the hemisphere centered at a given cache location and it is too costly. We propose a simpler solution which involves splitting photons into two maps in the global photon map. In the first map called the *high-energy* photon map, photons in very bright scene regions are stored. The second map called the *low-energy* photon map stores the remaining photons, which effectively leads to small spatial variations of their density. In terms of irradiance integration, this map can be properly sampled by a small number of uniformly distributed sampling directions. The high-energy map involves explicit directional sampling towards regions of high photon concentration with controllable angular density of irradiance samples. In the following section, we introduce our algorithm for splitting the global photons into those two maps.

8.2 Algorithm

In the first step of our algorithm a voxel grid which contains the whole rendered scene is built. Each voxel has a counter which is the number of photons hitting on surfaces in this voxel. During the photon tracing stage, all photon hit points are initially stored in the low-energy photon map. However, when the number of photons stored in a given voxel is equal to a specified threshold value $c_{\max}$, all subsequent photons are stored in the high-energy photon map. $c_{\max}$ should be chosen so that photons are captured in this map only for strong secondary lights. In practice, the user decides on the threshold irradiance value $E_{\max}$ which is then used to compute the corresponding $c_{\max}$ using the following relation:

$$c_{\max} = \frac{E_{\max} A}{\Delta\Phi} \tag{8.1}$$

where $\Delta\Phi$ is the radiant power carried by each photon and $A = \pi r^2$ is an estimate of the surface area in the voxel of the edge length $2r$. $E_{\max}$ can be automatically selected as a function of the average scene irradiance E_{avg}, which is estimated for a certain percentage $p_{\%}$ of the pilot photons as:

$$E_{avg} = \frac{n\Delta\Phi}{A_{total}} \tag{8.2}$$

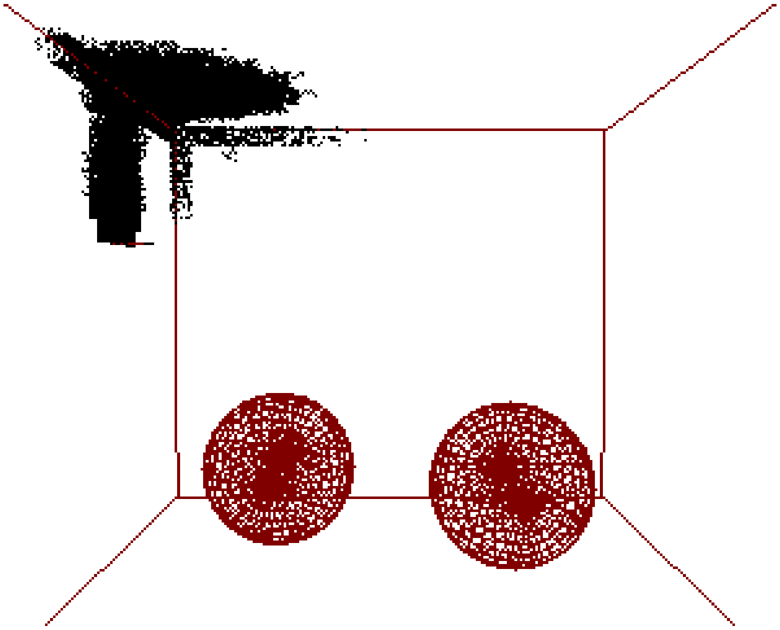


Figure 8.1: Distribution of photon hit points stored in the high-energy photon map for the scene shown in Figure 8.3. The black dots in the upper left region around the primary light source represent photons from this map.

where n is the number of photon hit points for $p_{\%}$ of all traced photons (usually we assume $p_{\%} = 10\%$) and A_{total} is an estimate of the surface area for all objects in the scene.

Figure 8.1 shows the distribution of photons stored in the high-energy photon map, which was obtained using this procedure. Although this scheme is fast and sufficiently accurate for our purposes, more elaborate methods to examine the density of photons are presented in [68].

During the rendering stage, the high-energy photon map is used to explicitly sample irradiance at a given point x in the scene by shooting rays toward the random locations inside the corresponding voxels. The number of rays per voxel can be either proportional to the photon count in each voxel or it can be just a fixed number. The reflected radiance at the location x for strong secondary light sources is represented as the integral of the differential irradiance dE for each high energy voxel:

$$L(x,\omega) = \int_{\Omega} f(x,\omega,\omega')V(x,\omega')dE(x,\omega') \tag{8.3}$$

Here Ω is the set of high energy voxels and $V(x,\omega')$ is the visibility function ($V(x,\omega') = 1$ when the ray traced from x in the direction ω' arrives to the selected voxel, otherwise $V(x,\omega') = 0$).

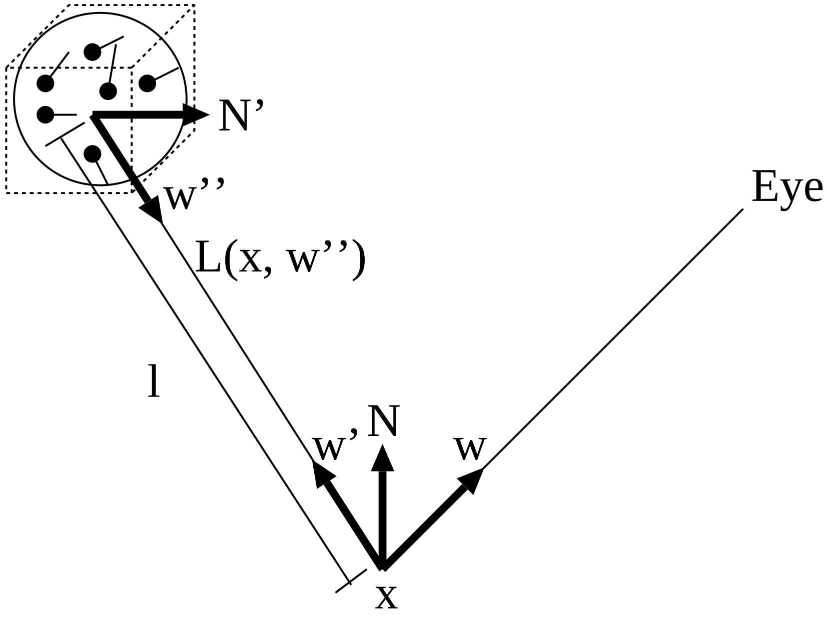


Figure 8.2: The schematic view of computing the differential irradiance dE for the selected voxel. The black dots inside the voxel represent photon hit locations with marked incoming directions.

Figure 8.2 illustrates the schematic view of the differential irradiance dE coming from a selected voxel. The differential irradiance dE is computed as:

$$dE(x,\omega') = L_i(x,\omega'')(\omega' \cdot N)d\vec{\omega}' \tag{8.4}$$

$$d\vec{\omega}' = \frac{(\omega'' \cdot N')A}{l^2} \tag{8.5}$$

The reflected radiance $L_i(x,\omega'')$ from a secondary light source (a selected voxel) is approximated by the density estimation of the high-energy photon map. N and N' are the normal vectors at the location x and the selected location inside the voxel, respectively. Here, $\omega'' = -\omega'$ and l is the distance between the location x and the location inside the voxel. The meaning of A is the same as in equation (8.1).

8.3 Results

We tested our algorithm for scenes shown in Figures 8.3 and 8.4. All images are rendered on a Pentium 4 Xeon 1.7 GHz, Debian GNU/Linux PC.

For images in Figure 8.3 the irradiance integration is performed for each pixel (the final gathering procedure). Figure 8.3a presents the result of stratified sam-

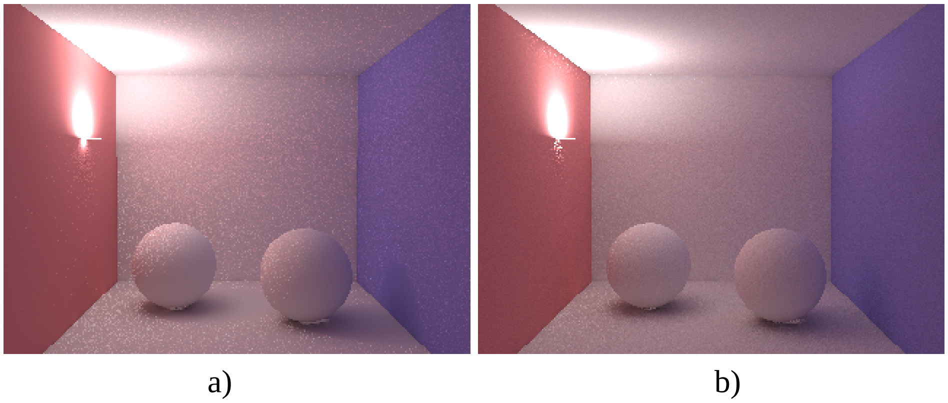


Figure 8.3: Final gathering procedure. Size: 320 × 240 pixels, a) 768 stratified samples/pixel, rendering time 21 min and b) 278 (48 stratified samples + 230 explicit samples) samples/pixel, rendering time 9 min.

pling for 768 samples per pixel using the traditional global photon map. Although rendering requires 21 minutes, stochastic noise is still perceivable. Figure 8.3b shows the results obtained using our technique. The image is rendered using 48 stratified samples towards the low-energy photon map and 230 explicit samples towards high-energy secondary light sources. The rendering time is 9 minutes. Although in our approach nearly three times less samples have been computed the overall image quality is significantly better than in Figure 8.3a.

In Figure 8.4, we demonstrate our algorithm in a more complex scene. The image is rendered using the irradiance caching. Because the irradiance integration is performed only on the irradiance cache locations, significant speedup of the computation is achieved. 300 stratified samples towards the low-energy photon map and 98 explicit samples towards the high-energy photon map are used for each irradiance integration. The total number of cache locations is 13,666 and the rendering time is 10 minutes for the image resolution of 1,128 × 480 pixels.

8.4 Discussion

Our algorithm separates the global photon map into the low- and high-energy photon maps. The former map is used for the irradiance integration as in traditional photon mapping algorithm. Because of the lower variation of photon distribution in this map a high accuracy of such integration can be easily achieved using only a small number of sampling directions. The high-energy photon map associated with the voxel grid identifies strong secondary light sources which are explicitly sampled by shooting rays toward the corresponding directions.

A similar idea is implemented in the *Radiance* rendering system [92], in which

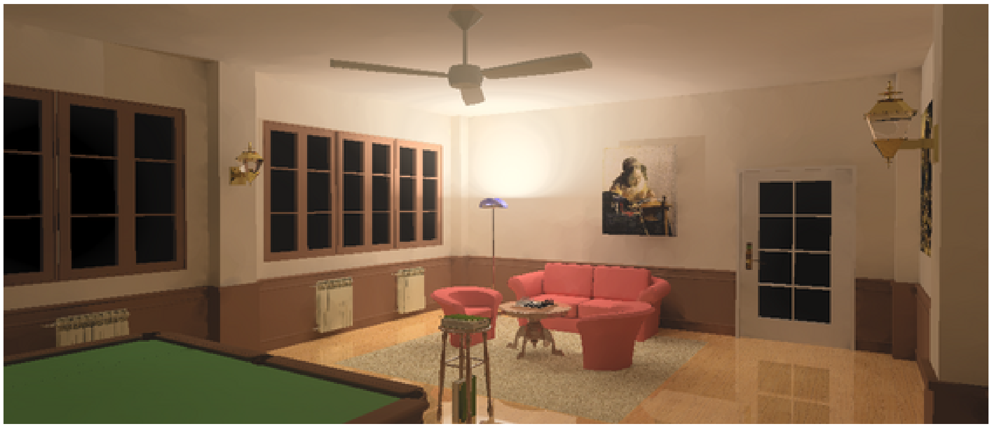


Figure 8.4: The image is rendered by our algorithm using the irradiance caching. Size: 1,128 × 480 pixels, 398 (300 stratified samples + 98 explicit samples) samples/cache, 13,666 caches, rendering time 10 min.

bright scene objects can be manually chosen as the secondary light sources. This scheme works only in the case when the bright objects are known in advance and can be manually selected as virtual lights to be explicitly sampled at the rendering stage. The advantage of our algorithm is that it can be applied automatically for any scene. Since we use the photon map and voxel grid data structures, our secondary light source representation is independent of any type of scene geometry. In our technique strong secondary light sources are automatically selected and the user must specify just one parameter to control this choice.

Another related work is the importance sampling using the photon map algorithm [31] in which the importance directional function is built based on the photon map at each rendered pixel. Our algorithm requires to build the importance positional function only once as the photon count in the voxel grid. In our method, the important directions are immediately found for each pixel by picking the selected voxels.

8.5 Conclusions

We presented an efficient algorithm for rendering scenes with strong secondary light sources. The performance of our algorithm is the better when a scene has the bigger variations in secondary lighting across the whole scene. Our technique can be considered as a variant of importance sampling embedded into the framework of photon mapping algorithm.

As future work we plan to extend our technique to handle dynamic scenes. We intend to exploit the temporal coherence in indirect lighting distribution to improve the computation efficiency.

CHAPTER 9

Conclusions and Future Work

Global illumination is an important visual cue in photo realistic rendering of animations. Producing high quality animations featuring rich object appearance and compelling lighting effects is very time consuming using traditional frame-by-frame rendering systems. In this thesis, we presented a number of global illumination and rendering solutions that exploit temporal coherence in lighting distribution for subsequent frames to improve the computation performance and overall animation quality. Our strategy relied on extending into temporal domain global illumination and rendering techniques such as density estimation path tracing, photon mapping, ray tracing, and irradiance caching, which were originally designed to handle static scenes only. This strategy of extending various algorithms is justified since the best technique which is suitable for all types of scenes and applications simply does not exist. Our solutions led to significant improvements of the computation performance and animation quality through the suppression of temporal aliasing. A practical question arises: *Which of the presented techniques should be chosen for a given application?*

In Chapter 4, we proposed an application of image-based techniques to derive inbetween frames based on a small number of high quality ray-traced keyframes. The key aspect of our algorithm was the perception-driven keyframe placement, which led to the significant reduction of image warping artifacts. The technique works well only for static scenes and walkthrough scenarios. The performance of this technique can be affected for scenes with many specular objects for which the computation must be performed in principle for each frame. Using our perception-based quality metric made it possible to determine for which pixels and frames such computations must be performed to avoid perceivable artifacts.

In Chapter 5, we extended the density estimation path tracing algorithm into the temporal domain by sharing photon-hit points between the subsequent frames and using advanced spatio-temporal filters for lighting reconstruction. The resulting spatio-temporal photon density estimation technique seems to be the most suitable for practical animation systems, where the rendering speed is the key factor even at the expense of lower accuracy in the lighting computation. This method does not require to store photon-hit positions between the subsequent frames. Photon counters for each mesh element are sufficient to significantly reduce the memory requirements. Recently, the technique was implemented as a plug-in for the 3D StudioMax system [94], and experiences with the exploitation of this technique in the animation production are very good.

In Chapters 6–8, we proposed a number of extensions to the photon mapping technique and focused on the problem of efficient rendering using the irradiance cache approach. We extended the irradiance cache into the temporal domain by reusing cache locations (Chapter 6) and selectively updating directional samples that contribute to the cached irradiance whenever required due to changes in a scene (Chapter 7). Both techniques are suitable for all applications in which the quality requirements are very high. The photon mapping technique based on static and dynamic photons (Chapter 6) is suitable in particular for those applications in which scene changes do not significantly affect the lighting distribution (a vast majority of photons can be classified as static). Selective update of irradiance cache (Chapter 7) is suitable for any two-pass global illumination solution including radiosity, density estimation, and photon mapping. It leads to significant reduction of the computational cost. Efficient handling of strong secondary emitters in the irradiance cache approach as proposed in Chapter 8 improves the computational performance for scenes featuring unbalanced lighting distributions. The extension of this technique into the temporal domain was relegated as future work.

The common denominator for all presented algorithms is that they lead to high quality animations. However, each algorithm has its own characteristics and constraints which may affect the quality of frames. Table 9.1 summarizes those characteristics and constraints for all of the algorithms proposed in this thesis. The interpretation of columns in this table is as follows. The leftmost column refers to the chapter number in which a given algorithm was presented. The column ANIMATION TYPE specifies which component describing rendered scene can be changed from frame to frame. The column LIGHTING describes for which lighting component the temporal coherence between frames is used. Our main interest is the computation of indirect illumination which is the most time consuming part of global illumination. The column TEMPORAL COHERENCE describes our approach to the lighting computation for each frame (i.e., in the temporal domain). The column RENDERING METHOD specifies which global illumination and rendering algorithms are used. The column PRECISION determines at which level global illumination is solved and temporal coherence is exploited.

The algorithms proposed in this thesis have also a number of limitations. The problem of light transport in participating media has been completely ignored. It means that we consider only the case in which rays travel between surfaces without any interaction with media. However, an extension of our algorithms such as spatio-temporal density estimation and irradiance caching seems to be feasible for the participating media as well. It can be an interesting direction of future work especially given that the global illumination computation is even more costly in this case, and therefore exploiting temporal coherence may lead to more spectacular speedup.

Another promising direction of future work is an efficient handling of glossy surfaces. Our spatio-temporal density estimation and irradiance caching algorithms are basically limited to diffuse surfaces. In the former case, glossy surfaces could be handled, but then photons should be stored along with their incoming directions and significantly more photons should be considered. Additionally, the granularity of mesh should be reduced, which may lead to an explosion of mesh elements. A better chance of handling glossy surfaces has our spatio-temporal irradiance caching algorithm. This approach would require an extension of data structures to similar as presented in [40].

	ANIMATION TYPE	LIGHTING	TEMPORAL COHERENCE	RENDERING METHOD	PRECISION
Chapter 4	camera	direct + indirect	exact lighting computed only for keyframes	ray tracing + image-based rendering	outgoing radiance $L_o[W/m^2/sr]$ per pixel (image space)
Chapter 5	camera + object	indirect	rough indirect lighting computation for each frame	ray tracing + spatio-temporal density estimation	irradiance $E[W/m^2]$ per mesh element (object space)
Chapter 6	camera + object	indirect	separate computation of static and dynamic lighting components	photon mapping + irradiance caching	irradiance $E[W/m^2]$ per cache location (object space)
Chapter 7	camera + object + light	indirect	re-using and updating information from the previous frames	photon mapping + irradiance caching	incoming radiance samples $L_i[W/m^2/sr]$ at cache location (object space)
Chapter 8	-	strong indirect	-	photon mapping + irradiance caching	-

Table 9.1: Synthetic summary: Basic characteristics of algorithms developed within the scope of this thesis.

Bibliography

[1] A.A. Apodaca and L. Gritz. *Advanced RenderMan*. Morgan Kaufmann, San Francisco, 1999.

[2] G. Besuievsky and X. Pueyo. Animating Radiosity Environments through the Multi-Frame Lighting Method. *The Journal of Visualization and Computer Animation*, 12(2):93–106, 2001.

[3] P.R. Bevington and D.K. Robinson. *Data Reduction and Error Analysis for the Physical Sciences*. McGraw-Hill, Inc., New York, 1992.

[4] M.R. Bolin and G.W. Meyer. A Perceptually Based Adaptive Sampling Algorithm. In *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series, pages 299–310, 1998.

[5] M. Cammarano and H.W. Jensen. Time Dependent Photon Mapping. In *Proceedings of the 13th Eurographics Workshop on Rendering*, pages 135–144, 2002.

[6] S.E. Chen. Incremental Radiosity: An Extension of Progressive Radiosity to an Interactive Image Synthesis System. In *Proc. of Siggraph'90*, pages 135–144, 1990.

[7] P.H. Christensen. Faster Photon Map Global Illumination. *Journal of Graphics Tools*, 4(3):1–10, 1999.

[8] P.H. Christensen. Photon Mapping Tricks. In *Siggraph 2002, Course Notes No. 43, A Practical Guide to Global Illumination using Photon Mapping organized by Jensen, H.W.*, pages 93–121, 2002.

[9] P.H. Christensen and D. Batali. An Irradiance Atlas for Global Illumination in Complex Production Scenes. In *Rendering Techniques 2004 (Proceedings of the Eurographics Symposium on Rendering 2004)*, pages 133–141, 2004.

[10] P.H. Christensen, D.M. Laur, J. Fong, W.L. Wooten, and D. Batali. Ray Differentials and Multiresolution Geometry Caching for Distribution Ray Tracing in Complex Scenes. In *Computer Graphics Forum (Eurographics 2003 Conference Proceedings)*, pages 543–552. Blackwell Publishers, 2003.

[11] P.H. Christensen, D. Lischinski, E.J. Stollnitz, and D.H. Salesin. Clustering for Glossy Global Illumination. *ACM Transactions on Graphics*, 16(1):3–33, 1997.

[12] M.F. Cohen and J.R. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press Professional, 1993.

[13] R.L. Cook, T. Porter, and L. Carpenter. Distributed Ray Tracing. In *Computer Graphics (Proceedings of SIGGRAPH 84)*, volume 18, pages 137–145, July 1984.

[14] S. Daly. The Visible Differences Predictor: An Algorithm for the Assessment of Image Fidelity. In A.B. Watson, editor, *Digital Image and Human Vision*, pages 179–206. Cambridge, MA: MIT Press, 1993.

[15] C. Damez, K. Dmitriev, and K. Myszkowski. State of the Art in Global Illumination for Interactive Applications and High-Quality Animations. *Computer Graphics Forum*, 22(1):55–78, 2003.

[16] C. Damez and F.X. Sillion. Space-Time Hierarchical Radiosity for High-Quality Animations. In *Eurographics Rendering Workshop 1999*, pages 235–246, 1999.

[17] K. Dmitriev, S. Brabec, K. Myszkowski, and H.-P. Seidel. Interactive Global Illumination Using Selective Photon Tracing. In *Rendering Techniques 2002: 13th Eurographics Workshop on Rendering*, pages 25–36, June 2002.

[18] G. Drettakis and F.X. Sillion. Interactive Update of Global Illumination Using a Line-Space Hierarchy. In *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 57–64, 1997.

[19] P. Dutre, P. Bekaert, and K. Bala. *Advanced Global Illumination*. A. K. Peters, Natick, MA, 2003.

[20] J.A. Ferwerda, S. Pattanaik, P. Shirley, and D.P. Greenberg. A Model of Visual Masking for Computer Graphics. In *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 143–152, 1997.

[21] P. Gautron, J. Křivánek, S.N. Pattanaik, and K. Bouatouch. A Novel Hemispherical Basis for Accurate and Efficient Rendering. In *Rendering Techniques 2004, Eurographics Symposium on Rendering*, pages 321–330, June 2004.

[22] D.W. George, F.X. Sillion, and D.P. Greenberg. Radiosity Redistribution for Dynamic Environments. *IEEE Comp. Graphics and Applications*, 10(4):26–34, 1990.

[23] S. Gibson and R.J. Hubbold. Efficient Hierarchical Refinement and Clustering for Radiosity in Complex Environments. *Computer Graphics Forum*, 15(5):297–310, 1996.

[24] B. Girod. The Information Theoretical Significance of Spatial and Temporal Masking in Video Signals. In *Proc. of SPIE Vol. 1077*, pages 178–187, 1989.

[25] A.S. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann Publishers, Inc., 1995.

[26] D.P. Greenberg. A Framework for Realistic Image Synthesis. *Communications of the ACM*, 42(8):43–53, August 1999.

[27] J. Guenther, I. Wald, and P. Slusallek. Realtime Caustics using Distributed Photon Mapping. In *Proceedings of the Eurographics Symposium on Rendering*, 2004.

[28] V. Havran, C. Damez, K. Myszkowski, and H.-P. Seidel. An efficient spatio-temporal architecture for animation rendering. In *Rendering Techniques 2003 : 14th Eurographics Workshop on Rendering*, pages 106–117, Leuven, Belgium, June 2003. Association of Computing Machinery (ACM), ACM.

[29] P. Heckbert. Adaptive Radiosity Textures for Bidirectional Ray Tracing. In *Computer Graphics (ACM SIGGRAPH '90 Proceedings)*, pages 145–154, August 1990.

[30] R. Jain, Kasturi R., and B.G. Schunck. *Machine Vision*. McGraw-Hill, New York, 1995.

[31] H.W. Jensen. Importance Driven Path Tracing using the Photon Map. In *Eurographics Workshop on Rendering*, pages 326–335, 1995.

[32] H.W. Jensen. Global Illumination Using Photon Maps. In *Eurographics Workshop on Rendering*, pages 21–30, 1996.

[33] H.W. Jensen. *Realistic Image Synthesis Using Photon Mapping*. A. K. Peters, Natick, MA, 2001.

[34] H.W. Jensen and N.J. Christensen. Efficiently Rendering Shadows Using the Photon Map. In *Compugraphics*, 1995.

[35] J.T. Kajiya. The Rendering Equation. In *Proc. of Siggraph'86*, pages 143–150, 1986.

[36] M.M. Kalos and P.A. Whitlock. *Monte Carlo Methods.* Wiley International, 1986.

[37] T. Kato. Photon Mapping in Kilauea. In *Siggraph 2002, Course Notes No. 43, A Practical Guide to Global Illumination using Photon Mapping organized by Jensen, H.W.*, pages 122–191, 2002.

[38] E.P. Lafortune and Y.D. Willems. Bidirectional Path Tracing. In *Compugraphics '93*, pages 95–104, 1993.

[39] E.P. Lafortune and Y.D. Willems. Using the Modified Phong Reflectance Model for Physically Based Rendering. Report cw 197, Department of Computer Science, K.U.Leuven, 1994.

[40] E.P. Lafortune and Y.D. Willems. A 5D Tree to Reduce the Variance of Monte Carlo Ray Tracing. In *Eurographics Rendering Workshop 1995*, pages 11–20, June 1995.

[41] D. Lischinski and A. Rappoport. Image-Based Rendering for Non-Diffuse Synthetic Scenes. In G. Drettakis and N. Max, editors, *Rendering Techniques '98*, pages 301–314. Springer, 1998.

[42] D. Lischinski, F. Tampieri, and D.P. Greenberg. Combining Hierarchical Radiosity and Discontinuity Meshing. In *Proc. of Siggraph'93*, pages 199–208, 1993.

[43] J. Lubin. A Human Vision Model for Objective Picture Quality Measurements. In *Conference Publication No. 447*, pages 498–503. IEE International Broadcasting Convention, 1997.

[44] W.R. Mark, L. McMillan, and G. Bishop. Post-Rendering 3D Warping. In *1997 Symposium on Interactive 3D Graphics*, pages 7–16. ACM Siggraph, 1997.

[45] I. Martin, X. Pueyo, and D. Tost. Frame-to-Frame Coherent Animation with Two-Pass Radiosity. *IEEE Transactions on Visualization and Computer Graphics*, 9(1):70–84, 2003.

[46] L. McMillan. *An Image-Based Approach to 3D Computer Graphics*. Ph.D. thesis, North Carolina University, Chapel Hill, 1997.

[47] G. Miller, S. Rubin, and D. Poncelen. Lazy Decompression of Surface Light Fields for Precomputed Global Illumination. In G. Drettakis and N. Max, editors, *Rendering Techniques ’98*, pages 281–292. Springer, 1998.

[48] S. Müller, W. Kresse, and F. Schoeffel. A Radiosity Approach for the Simulation of Daylight. In *Eurographics Rendering Workshop 1995*, pages 137–146, 1995.

[49] K. Myszkowski, P. Rokita, and T. Tawara. Perceptually-Informed Accelerated Rendering of High Quality Walkthrough Sequences. In *Eurographics Rendering Workshop 1999*, pages 5–18, 1999.

[50] K. Myszkowski, P. Rokita, and T. Tawara. Perception-Based Fast Rendering and Antialiasing of Walkthrough Sequences. *IEEE Transactions on Visualization and Computer Graphics*, 6(4):360–379, October 2000.

[51] K. Myszkowski, T. Tawara, H. Akamine, and H.-P. Seidel. Perception-Guided Global Illumination Solution for Animation Rendering. In Eugene Fiume, editor, *Computer Graphics (SIGGRAPH-2001): Conference Proceedings*, Annual Conference Series, pages 221–230, Los Angeles, USA, 2001. Association of Computing Machinery (ACM), ACM.

[52] K. Myszkowski, T. Tawara, and H.-P. Seidel. Using Animation Quality Metric to Improve Efficiency of Global Illumination Computation for Dynamic Environments. In Bernice Rogowitz and Thrasyvoulos Pappas, editors, *Proceedings of 7th SPIE Conference Human Vision and Electronic Imaging*, volume 4662 of *SPIE Proceedings Series*, pages 187–196, San Jose, USA, 2002. SPIE - The International Society for Optical Engineering.

[53] J. Nimeroff, J. Dorsey, and H. Rushmeier. Implementation and Analysis of an Image-Based Global Illumination Framework for Animated Environments. *IEEE Transactions on Visualization and Computer Graphics*, 2(4):283–298, 1996.

[54] M. Pharr and G. Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, San Francisco, 2004.

[55] X. Pueyo, D. Tost, I. Martin, and B. Garcia. Radiosity for Dynamic Environments. *The Journal of Visualization and Comp. Animation*, 8(4):221–231, 1997.

[56] M. Ramasubramanian, S.N. Pattanaik, and D.P. Greenberg. A Perceptually Based Physical Error Metric for Realistic Image Synthesis. In *SIGGRAPH 99 Conference Proceedings*, Annual Conference Series, pages 73–82, 1999.

[57] M.C. Reichert. *A Two-Pass Radiosity Method to Transmitting and Specularly Reflecting Surfaces*. M.Sc. thesis, Cornell University, 1992.

[58] H. Rushmeier, G. Ward, C. Piatko, P. Sanders, and B. Rust. Comparing Real and Synthetic Images: Some Ideas About Metrics. In *Eurographics Workshop on Rendering*, pages 82–91, 1995.

[59] A. Scheel, M. Stamminger, and H.-P. Seidel. Thrifty Final Gather for Radiosity. In *Rendering Techniques 2001: 12th Eurographics Workshop on Rendering*, pages 1–12. Eurographics, June 2001.

[60] A. Scheel, M. Stamminger, and H.-P. Seidel. Grid based Final Gather for Radiosity on Complex Clustered Scenes. In *Eurographics*, 2002.

[61] F. Schoeffel and P. Pomi. Reducing Memory Requirements for Interactive Radiosity Using Movement Prediction. In *Eurographics Rendering Workshop 1999*, pages 225–234, 1999.

[62] J.W. Shade, S.J. Gortler, L. He, and R Szeliski. Layered Depth Images. In *SIGGRAPH 98 Conference Proceedings*, pages 231–242, 1998.

[63] M. Shinya. Spatial Anti-aliasing for Animation Sequences with Spatio-temporal Filtering. In *SIGGRAPH ’93 Proceedings*, volume 27, pages 289–296, 1993.

[64] P. Shirley, B. Wade, P.M. Hubbard, D. Zareski, B. Walter, and D.P. Greenberg. Global Illumination via Density Estimation. In *Eurographics Rendering Workshop 1995*, pages 219–230, 1995.

[65] F.X. Sillion and C. Puech. *Radiosity and Global Illumination*. Morgan Kaufmann, San Francisco, 1994.

[66] P. Slusallek, W. Heidrich, and H.-P. Seidel. Radiance Maps: An Image-Based Approach to Global Illumination. In *SIGGRAPH ’98, Technical Sketch*, 1998.

[67] B. Smits. *Efficient Hierarchical Radiosity in Complex Environments*. Ph.D. thesis, Cornell University, 1994.

[68] F. Suykens and Y. Willens. Density Control for Photon Maps. In *Rendering Techniques 2000: 11th Eurographics Workshop on Rendering*, pages 23–34, June 2000.

[69] E. Tabellion and A. Lamorlette. An Approximate Global Illumination System for Computer-Generated Films. In *SIGGRAPH 2004*, pages 467–474, 2004.

[70] T. Tawara, K. Myszkowski, K. Dmitriev, V. Havran, C. Damez, and H.-P. Seidel. Exploiting Temporal Coherence in Global Illumination. In *Spring conference on Computer Graphics (SCCG 2004)*, pages 16–26, Budmerice, Slovakia, 2004. ACM.

[71] T. Tawara, K. Myszkowski, and H.-P. Seidel. Localizing the Final Gathering for Dynamic Scenes using the Photon Map. In Günther Greiner, Heinrich Niemann, Thomas Ertl, Bernd Girod, and Hans-Peter Seidel, editors, *Proceedings of Vision, Modeling, and Visualization VMV 2002*, pages 69–76, Erlangen, Germany, November 2002. Akademische Verlagsgesellschaft Aka GmbH.

[72] T. Tawara, K. Myszkowski, and H.-P. Seidel. Efficient Rendering of Strong Secondary Lighting in Photon Mapping Algorithm. In *Theory and Practice of Computer Graphics 2004*, pages 174–178, University of Bournemouth, UK, 2004. UK chapter of the Eurographics Society, IEEE Computer Society.

[73] T. Tawara, K. Myszkowski, and H.-P. Seidel. Exploiting Temporal Coherence in Final Gathering for Dynamic Scenes. In *Computer Graphics International (CGI 2004)*, pages 110–119, Crete, Greece, 2004. Computer Graphics Society (CGS), IEEE Computer Society.

[74] X. Tong, D. Heeger, and C. van den Branden Lambrecht. Video Quality Evaluation Using ST-CIELAB. In *Human Vision and Electronic Imaging IV*, pages 185–196. Proceedings of SPIE Vol. 3644, 1999.

[75] J. Tumblin and H.E. Rushmeier. Tone Reproduction for Realistic Images. *IEEE Computer Graphics and Applications*, 13(6):42–48, November 1993.

[76] E. Veach. *Robust Monte Carlo Methods for Light Transport*. Ph.D. thesis, Stanford University, 1997.

[77] E. Veach and L.J. Guibas. Bidirectional Estimators for Light Transport. In *Fifth Eurographics Workshop on Rendering*, pages 147–162, 1994.

[78] C.P. Verbeck and D.P. Greenberg. A Comprehensive Light Source Description for Computer Graphics. *IEEE Computer Graphics & Applications*, 4(7):66–75, July 1984.

[79] V. Volevich, K. Myszkowski, A. Khodulev, and E.A. Kopylov. Using the Visible Differences Predictor to Improve Performance of Progressive Global Illumination Computations. *ACM Transactions on Graphics*, 19(2):122–161, 2000.

[80] V. Volevich, K. Myszkowski, A.B. Khodulev, and E.A. Kopylov. Perceptually-Informed Progressive Global Illumination Solution. Technical Report TR-99-1-002, Department of Computer Science, Aizu University, February 1999.

[81] I. Wald, C. Benthin, and P. Slusallek. Interactive Global Illumination in Complex and Highly Occluded Environments. In *Eurographics Symposium on Rendering: 14th Eurographics Workshop on Rendering*, pages 74–81, June 2003.

[82] I. Wald, T. Kollig, C. Benthin, A. Keller, and P. Slusallek. Interactive Global Illumination using Fast Ray Tracing. In *Rendering Techniques 2002: 13th Eurographics Workshop on Rendering*, pages 15–24, June 2002.

[83] I. Wald, T.J. Purcell, J. Schmittler, C. Benthin, and P. Slusallek. Realtime Ray Tracing and its use for Interactive Global Illumination. In *Eurographics State of the Art Reports*, 2003.

[84] I. Wald and P. Slusallek. State-of-the-Art in Interactive Ray-Tracing. In *State of the Art Reports, EUROGRAPHICS 2001*, pages 21–42, 2001.

[85] I. Wald, P. Slusallek, and C. Benthin. Interactive Distributed Ray Tracing of Highly Complex Models. In *Rendering Techniques 2001: 12th Eurographics Workshop on Rendering*, pages 277–288, June 2001.

[86] B.J. Walter. *Density Estimation Techniques for Global Illumination*. Ph.D. thesis, Cornell University, 1998.

[87] M. Wand and W. Straßer. Multi-Resolution Rendering of Complex Animated Scenes. *Computer Graphics Forum*, 21(3):483–491, 2002.

[88] G. Ward. Real Pixels. In J. Arvo, editor, *Graphics Gems II*, pages 80–83, 1991.

[89] G. Ward. The RADIANCE Lighting Simulation and Rendering System. In *SIGGRAPH 94 Conference Proceedings*, Annual Conference Series, pages 459–472, 1994.

[90] G. Ward and P. Heckbert. Irradiance Gradients. In *Eurographics Workshop on Rendering*, pages 85–98, 1992.

[91] G. Ward., F. Rubinstein, and R. Clear. A Ray Tracing Solution to Diffuse Interreflection. In *Computer Graphics (SIGGRAPH 88 Conference Proceedings)*, pages 85–92, 1988.

[92] G. Ward and R. Shakespeare. *Rendering with Radiance*. Morgan Kaufmann, Los Altos, California, 1998.

[93] A.B. Watson. Toward a Perceptual Video Quality Metric. In *Human Vision and Electronic Imaging III*, pages 139–147. Proceedings of SPIE Vol. 3299, 1998.

[94] M. Weber, M. Milch, K. Myszkowski, K. Dmitriev, P. Rokita, and H.-P. Seidel. Spatio-Temporal Photon Density Estimation Using Bilateral Filtering. In *Computer Graphics International (CGI 2004)*, pages 120–127, Crete, Greece, 2004. IEEE Computer Society.

[95] H. Weghorst, G. Hooper, and D.P. Greenberg. Improved Computational Methods for Ray Tracing. *ACM Transactions on Graphics*, 3(1):52–69, January 1984.

[96] T. Whitted. An Improved Illumination Model for Shaded Display. In *Communications of the ACM*, volume 23, pages 343–349, 1980.

[97] Y.L.H. Yee. *Spatiotemporal Sensitivity and Visual Attention for Efficient Rendering of Dynamic Environments*. M.Sc. thesis, Cornell University, 2000.