Volume 0 (1981), Number 0 pp. 1–27 COMPUTER GRAPHICS *forum*

The State of the Art in Interactive Global Illumination

Tobias Ritschel[1,2] Carsten Dachsbacher[3] Thorsten Grosch[4] Jan Kautz[5]

Télécom ParisTech / CNRS (LTCI)[1] and Intel Visual Computing Institute[2]
Karlsruhe Institute of Technology[3] University of Magdeburg[4] University College London[5]

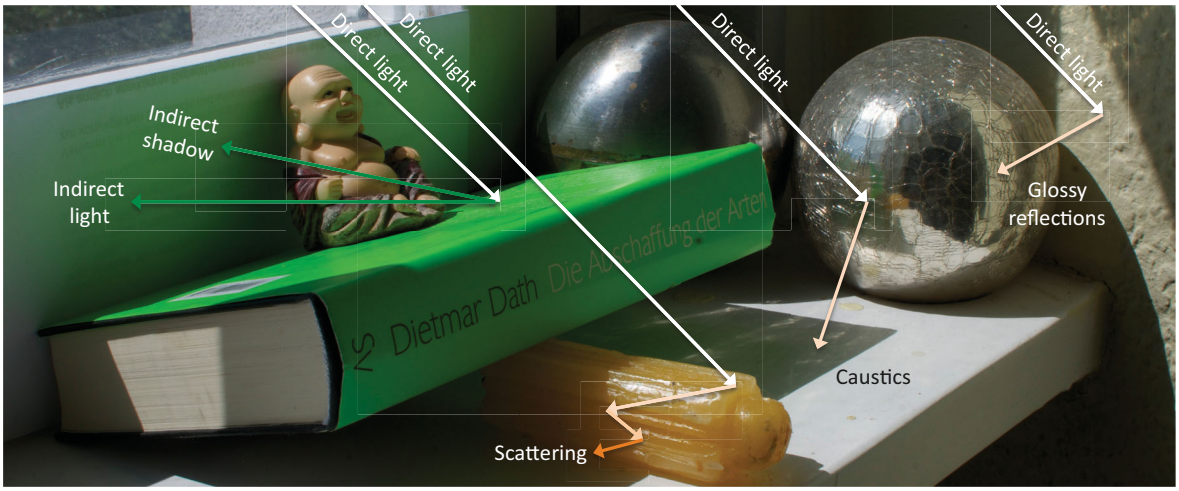

Figure 1: *Photography of a scene with global illumination: Multiple diffuse and specular bounces, caustics and scattering.*

Abstract

The interaction of light and matter in the world surrounding us is of striking complexity and beauty. Since the very beginning of computer graphics, adequate modeling of these processes and efficient computation is an intensively studied research topic and still not a solved problem. The inherent complexity stems from the underlying physical processes as well as the global nature of the interactions that let light travel within a scene. This article reviews the state of the art in interactive global illumination *computation, that is, methods that generate an image of a virtual scene in less than one second with an as exact as possible, or plausible, solution to the light transport. Additionally, the theoretical background and attempts to classify the broad field of methods are described. The strengths and weaknesses of different approaches, when applied to the different visual phenomena, arising from light interaction are compared and discussed. Finally, the article concludes by highlighting design patterns for interactive global illumination and a list of open problems.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture / Radiosity / Raytracing

1. Introduction

Light in the real world interacts with surrounding media and surfaces creating stunning visual phenomena such as color bleeding, reflections, crepuscular rays, and caustics. Needless to say that the reproduction of the real world has been the ultimate goal of computer graphics ever since. Although the underlying principles of light-matter interaction are well understood, the efficient computation is still a challenging problem. Many processes are so computationally expensive that they require the development of simplified models, tailored algorithms and data structures. In this article we will give a survey and classification focusing on interactive methods for computing light transport in virtual scenes (global

illumination). These methods resort to work in related fields of computer graphics: first, measuring or modeling of reflectance of surfaces or scattering of participating media itself is an active area of research beyond the scope of this report. Second, global illumination often builds on methods that compute visibility between a point and a surface, or two surfaces, e. g., when computing shadows. Again, we refer to an excellent survey [HLHS03] and a recent book [ESAW11]. Worth mentioning in this context are also a survey on ambient occlusion [MFS08] (which can be considered as an approximation to global illumination), a survey of specular effects oin the GPU [SKUP*09] and courses both on "Global Illumination Across Industries" [KFC*10] and scattering in participating media [GWWD09]. A book about global illumination in general is "Advanced Global Illumination" by Dutré and colleagues [DBBS06] as well as PBRT by Pharr and Humphreys [PH04].

Although we focus on the state of the art in *interactive global illumination* this article will reveal a huge body of work in this field that we faithfully attempted to collect, review, classify, and compare. In this selection we concentrate on methods that produce plausible global illumination (GI) solutions in less than one second (on contemporary hardware). Although other definitions of "interactive speed" exist, we consider this threshold adequate for computing solutions to this inherent complex problem. Interactive GI is usually based on certain simplifications and making tradeoffs in the continuum between computationally expensive high-quality approaches and simple models.

The rendering equation introduced by Kajiya [Kaj86] and the operator notation by Arvo et al. [ATS94] are two important concepts that allowed to compare and evaluate GI approaches in a formalized way. The insight that light transport can be described as an integral equation, however, is significantly older, e. g., see Yamauti [Yam26] and Buckley [Buc27], and interreflections have been studied and computed by Higbie [Hig34] and Moon [Moo40] to name at least a few of the pioneers in this field.

Traditionally, one important application of GI is architectural visualization. In recent years, the ever growing (and converging) markets of interactive media, such as computer games and feature films, have started to shift from ad-hoc modelling of visual effects to more physically-plausible lighting computation. GI also has applications in computer vision [Lan11]; while Forsyth and Zisserman [FZ91] consider GI an artifact that complicates pattern recognition, Nayar et al. [NIK90] exploit the information stemming from interreflections to support shape acquisition. Color constancy, i. e., discounting for the color of the illuminant to recover reflectance, can also benefit from GI computation [FDH91]. Removing illumination has applications, among others, when scanning books [WUM97] or compensating indirect illumination in virtual reality projector systems [BGZ*06]. Augmented Reality applications profit from interactive global illumination, because all virtual modifications seamlessly integrate into the real world [Deb98]. If the current lighting conditions are captured, this information can be used to illuminate the virtual changes, e. g., virtual objects appear with consistent illumination and mutual shadows [GCHH03] between real and virtual objects can be displayed using differential rendering. Typical applications are virtual furniture, virtual prototypes and feature films.

We will start by introducing the most important theoretical concepts, such as the rendering equation, reflectance, visibility, etc., as a basis for the subsequent discussion of different approaches, models, and simplifications (Sec. 2). The approaches include among others ray tracing, finite element methods, bi-directional methods such as photon mapping and instant radiosity (Sec. 3). Afterwards we will discuss the strengths and weaknesses of the different approaches when being applied to certain phenomena, e. g., diffuse or glossy interreflections, caustics, volumetric scattering (Sec. 4). The article highlights some design patterns and concepts that have proven to be useful when computing GI, e. g., precomputation, using surfels to decoupled from the input geometry, multi-resolution and screen-space techniques (Sec. 5). The design patterns in Sec. 5 are the building blocks of the approaches in Sec. 3 and at the same time the approaches in Sec. 3 consist of components which are strategies in Sec. 5, e. g., "Splatting" can be used for GPU "Photon mapper" and "Instant radiosity" at the same time. Therefore, Sec. 5 and Sec. 3 constitute two views on the same state of the art which are not fully orthogonal. Finally, we conclude and show a list of open problems (Sec. 6).

2. Theory

In this section we will introduce the basic theory of GI. Light transport between surfaces with no surrounding media (i. e., in vacuum) is described by the rendering equation (Sec. 2.1), which "links" light sources, surface reflectance (BRDFs), and visibility. The transport of light in the presence of participating media is more involved and described by the volumetric rendering equation (Sec. 2.2).

2.1. Rendering equation

The *rendering equation* (RE) [Kaj86] (Fig. 2) states that the outgoing radiance L_o at a surface location $\mathbf{x}$ in direction ω is the sum of emitted radiance L_e and reflected radiance L_r:

$$L_o(\mathbf{x},\omega) = L_e(\mathbf{x},\omega) + L_r(\mathbf{x},\omega). \quad (1)$$

The reflected radiance is computed as:

$$L_r(\mathbf{x},\omega) = \int_{\Omega^+} L_i(\mathbf{x},\omega_i) f_r(\mathbf{x},\omega_i \to \omega) \langle N(\mathbf{x}),\omega_i\rangle^+ \mathrm{d}\omega_i, \quad (2)$$

where Ω^+ is the upper hemisphere oriented around the surface normal $N(\mathbf{x})$ at $\mathbf{x}$, f_r the bi-directional reflectance function (BRDF) and $\langle\rangle^+$ a dot product that is clamped to zero.

To determine the incident radiance, L_i, the ray casting operator is used to determine from which other surface location this radiance is emitted and reflected. It can be seen that the rendering equation is a Fredholm integral equation of the second kind. The goal of global illumination algorithms is to compute $L_o(\mathbf{x},\omega)$ for a given scene, materials and lighting L_e.

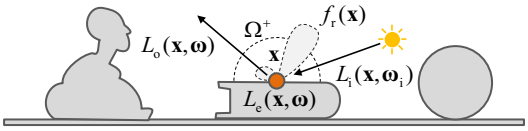

Figure 2: *The rendering equation*

Alternatively, the reflection integral can also be formulated over the scenes' surfaces $\mathcal{S}$ instead of directions. This implicates two modifications: first, a geometry term is introduced which accounts for the solid angle of a differential surface; second, the visibility function (sometimes considered part of the geometry term) determines the mutual visibility of two surface points:

$$L_r(\mathbf{x},\omega) := \int_{\mathcal{S}} L_i(\mathbf{s},\omega_i) f_r(\mathbf{x},\omega_i \to \omega) \langle N(\mathbf{x}),\omega_i\rangle^+ G(\mathbf{x},\mathbf{s}) d\mathbf{s}, \tag{3}$$

where $\mathcal{S}$ is the surface of the entire scene, $\omega' := \mathbf{s}-\mathbf{x}$ the difference vector from $\mathbf{x}$ to $\mathbf{s}$, $\omega_i := \omega'/||\omega'||$ the normalized difference vector and

$$G(\mathbf{x},\mathbf{s}) := \frac{\langle N(\mathbf{s}),(-\omega_i)\rangle^+ V(\mathbf{x},\mathbf{s})}{||\mathbf{s}-\mathbf{x}||^2},$$

a distance term where $V(\mathbf{x},\mathbf{s})$ is the visibility function that is zero if a ray between $\mathbf{x}$ and $\mathbf{s}$ is blocked and one otherwise.

Due to the repetitive application of the reflection integral, indirect lighting is distributed spatially and angularly and ultimately gets smoother with an increasing number of bounces.

Light In addition to geometry and material definition, the initial lighting in a scene L_e, obviously is an essential input to the lighting simulation. In computer graphics several models of light sources, such as point, spot, directional, and area lights, exist.

Point lights are the simplest type of light sources, where the emission is specified as the position of the light source and the directional distribution of spectral intensity. The incident radiance due to a point light at a surface is then computed from these parameters.

Real-world light sources have a finite area that emits light, where the spatial, directional and spectral distribution can, in principle, be arbitrary. In computer graphics, often a directional Lambertian emission is assumed, while a spatially varying emission is often referred to as "textured area-light".

Other commonly used models in computer graphics are: (1) spot lights, which can be considered as point lights with a directionally focused emission; (2) directional lights, assuming parallel light rays; and (3) environment maps which store incident radiance for every direction, however, assuming that $L_i(\mathbf{x},\omega_i)$ is independent of $\mathbf{x}$, i. e., $L_i(\mathbf{x},\omega_i) = L_{env}(\omega_i)$.

Reflectance The *Bi-directional Reflectance Distribution Function* (BRDF) is a 4D function that defines how light is reflected at a surface. The function returns the ratio of reflected radiance exiting along ω_o to the irradiance incident on the surface from direction ω_i. Physically plausible BRDFs have to be both symmetric $f_r(\omega_i \to \omega_o) = f_r(\omega_o \to \omega_i)$ and energy conserving $\int_{\Omega^+} f_r(\omega_i \to \omega_o) \langle N,\omega_i\rangle^+ d\omega_i < 1$. A special case is the Lambertian BRDF which is independent of the outgoing direction, and the perfect mirror BRDF which is a Dirac delta function in the direction of ω_i mirrored at the surface normal at $\mathbf{x}$. BRDFs inbetween these two extrema are often vaguely classified as directional-diffuse, glossy, and specular. BRDFs can be spatially invariant, or vary across the surface. In the latter case, BRDFs are called *spatially varying BRDFs* or *Bi-directional Texture Functions* (BTFs). Many analytical BRDFs models, ranging from purely phenomenological to physically-based models, exist, which can be either used as is, of fitted to measured BRDF or BTF data. If the material is not purely opaque, i. e., if light can enter or leave an object, then *Bi-directional Scattering Distribution Functions* (BSDFs) are used which extend the domain from the hemisphere to the entire sphere.

Visibility Both versions of the rendering equation imply some form of visibility computation: Eq. 1 uses the ray casting operator to determine the closest surface (for a given direction) and Eq. 3 explicitly uses a binary visibility function to test mutual visibility. Non-visible surfaces are usually referred to as being *occluded*.

If the visibility is computed between a point and a surface, then the surface is said to be in *shadow* or be *unshadowed*. The visibility between a surface point and an area light source is non-binary resulting in *soft shadows*. A full survey of existing (soft) shadow methods is beyond the scope of this report and we refer to the survey of Hasenfratz et al. [HLHS03] and a recent book [ESAW11]. Note that many methods dedicated to real-time rendering of soft-shadows often make simplifying assumptions like planar rectangular light sources, isotropic emittance, Lambertian receivers, and so forth, that allow for drastic speed-up compared to accurate computation (as in most GI methods), but also a loss of realism. Indirect light bouncing off a surface is comparable to lighting from an area light source.

2.2. Volume rendering equation

Light transport in the presence of participating media is described by the volume rendering equation (Fig. 3). It com-

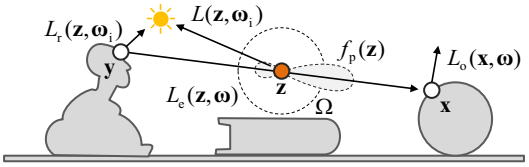

Figure 3: *The volume rendering equation*

bines *attenuated* surface radiance L_{att} from a surface at $\mathbf{y}$ and *scattered* radiance L_{att} collected in the medium between $\mathbf{x}$ and $\mathbf{y}$ as

$$L_o(\mathbf{x},\omega) = L_{att}(\mathbf{x},\omega) + L_{scatt}(\mathbf{x},\omega). \quad (4)$$

Scattering The scattered radiance at a point $\mathbf{x}$ incident from direction ω is:

$$L_{scatt}(\mathbf{x},\omega) = \int_0^{\|\mathbf{x}-\mathbf{z}\|} T_r(\mathbf{x},\mathbf{z}) L_i(\mathbf{z},-\omega) \mathrm{d}s, \quad (5)$$

with $\mathbf{z} = \mathbf{x} + s\omega$. The transmittance $T_r(\mathbf{x},\mathbf{z})$ accounts for both absorption and outscattering and defines the fraction of light that is not attenuated when light travels along the path from $\mathbf{z}$ to $\mathbf{x}$. The transmittance coefficient $\sigma_t(\mathbf{x}) = \sigma_a(\mathbf{x}) + \sigma_s(\mathbf{x})$ combines the absorption $\sigma_a(\mathbf{x})$ and scattering $\sigma_s(\mathbf{x})$ coefficients:

$$T_r(\mathbf{x},\mathbf{z}) = \exp\left(-\int_0^{\|\mathbf{x}-\mathbf{z}\|} \sigma_t(\mathbf{x}+s'\omega)\mathrm{d}s'\right). \quad (6)$$

The radiance at a point $\mathbf{z}$ in direction ω consists of the volume emission L_e and the inscattered radiance (second summand):

$$L_i(\mathbf{z},\omega) = L_e(\mathbf{z},\omega) + \sigma_s(\mathbf{z}) \int_\Omega f_p(\mathbf{z},\omega,\omega_i) L(\mathbf{z},\omega_i) d\omega_i. \quad (7)$$

The *phase function* f_p is the probability that light incident from direction ω_i is scattered into direction ω. In the simplest case of isotropic scattering $f_p(\mathbf{z},\omega,\omega_i) = \frac{1}{4\pi}$. It is a generalization of the BRDF in surface rendering.

Attenuation The radiance L_r leaving a surface point $\mathbf{y}$ in direction $-\omega$ undergoes attenuation (absorption) in the medium:

$$L_{att}(\mathbf{x},\omega) = T_r(\mathbf{x},\mathbf{y}) L_r(\mathbf{y},-\omega). \quad (8)$$

Even for offline rendering systems, the scattering equation is very costly to evaluate when multiple scattering is considered, i. e., if L in Eq. 7 accounts for previously scattered light as well.

2.3. Light path space notation

The light path notation of Heckbert [Hec90] will be used in this report. A light path starts at the light L, followed by a number of bounces that are either diffuse (D), specular (S) or volumetric (V) and end in the eye (E). One or more bounces are denoted as $^+$, two ore more bounces as $^{++}$, alternatives as |, such as in $\mathsf{L}\{\mathsf{D}|\mathsf{S}\}^+\mathsf{E}$ and an optional step as () such as the final optional specular in $\mathsf{LD}^+(\mathsf{S})\mathsf{E}$.

3. Approaches

This section reviews the classic approaches to compute interactive GI: Finite elements (Sec. 3.1), Monte Carlo ray tracing (Sec. 3.2), Photon mapping (Sec. 3.3), Instant radiosity (Sec. 3.4), Many-light-based GI (Sec. 3.5), Point-based GI (Sec. 3.6), Discrete ordinate methods (Sec. 3.7) and Precomputed radiance transfer (Sec. 3.8). Finally, caching strategies which are orthogonal to the aforementioned approaches are reviewed in Sec. 3.9. A qualitative comparison of the different approaches in terms of speed, quality, and so forth, can be found in Sec. 6.1. Other approaches, e. g., working with Eigenfunctions of the transport equation [KvD83] which are not used in interactive applications are not considered here.

3.1. Finite elements

Finite element (FE), or radiosity, is one classic approach to compute solutions to light transport problems. It was introduced to computer graphics by Goral et al. [GTGB84], which was the beginning of a very active period of research in that field. The underlying idea is to discretize the scene's surfaces into finite surface elements, often called patches, and then compute the light transport between them (Fig. 4). This has several implications: for every patch one needs to store the radiosity value for diffuse surfaces, or the directional distribution of incoming and outgoing light in the non-diffuse case. The light transport amounts to solving a linear system of equations once the form factors, which denote the amount of light transport between two patches, are known. The tempting property of radiosity is that once the solution is computed, it can be directly used for interactive walkthroughs. As these systems are solved numerically, e. g., using Southwell relaxation or progressive radiosity [CCWG88], the form factor computation is typically the most time consuming part [HSD94]. The first years of research concentrated on radiosity in diffuse scenes, reducing the form factors by hierarchical radiosity [HSA91], accounting for glossy materials [ICG86] and improving quality by final gathering [LTG93] and wavelet radiosity [GSCH93]. Hierarchical radiosity with clustering [SAG94, GWH01] overcomes the initial cost of $O(n^2)$ of linking n patches by grouping patches into clusters. One key limitation of FE is the issue of meshing the surface in such a way that light transport can be simulated accurately and efficiently [LTG93].

Early GPU diffuse radiosity computed a full-matrix solution without visibility [CHL04]. Dong et al. [DKTS07] precompute links, for a pre-defined geometry, similar to Immel et al. [ICG86], which can then be deformed at runtime, as long as the link structure remains valid. Links were used by Dachsbacher et al. [DSDD07, MD11] to exchange antiradiance to avoid visibility computation. Both methods in-

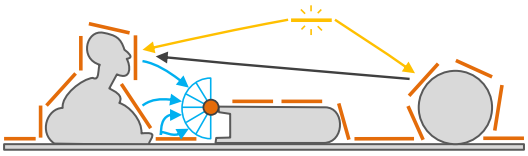

Figure 4: *Finite elements discretize the scene surface into patches (orange) that exchange energy via links (arrows). For directional exchange, a binning structure (blue fan) links a receiver with certain senders (blue arrows). The computational challenge lies in establishing the links and computing the exchange itself.*

clude non-diffuse radiosity by regular discretization of directions into bins. Recently, updating the link structure at runtime was considered [MESD09]. While these techniques (except Coombe et al. [CHL04]) support dynamic geometry, they are only well-suited to low-complexity scenes with moderate deformations.

Transport of light in screen space is also based on FE [RGS09] and hierarchical FE [NSW09, SHR10]. Here, the finite surface elements below each pixel (deferred shading [ST90]) replace the classic decomposition of a surface into FEs (meshing). Using GPUs to fill a deferred shading buffer by rasterization is extremely efficient, whereas classic meshing is complex and error-prone. Furthermore, a deferred shading buffer is output-sensitive and adapts the details to the relevant parts of the scene. The drawback is that such methods have to work with incomplete information, which leads to missing light or missing shadows from elements not present in the frame-buffer, due to clipping or occlusion. The use of other proxy geometry such as spheres [SGNS07] or voxels [RBA09, SHR10, THGM11] removes these problems, building a good compromise between speed and quality. Other elements, such as pixels in (multi-resolution) height fields [NS09] also allow for fast GI and avoid screen space problems, but are restricted in the geometry they can handle.

3.2. Monte-Carlo ray tracing

The rendering equation can be solved using Monte Carlo techniques [Kaj86]. To do so, a high number of directional

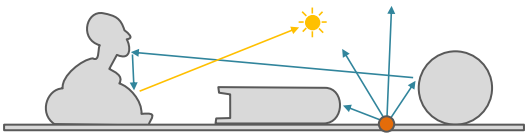

Figure 5: *Monte Carlo ray tracing at a sample (orange point) sends rays in random directions (blue arrows) and bounces them, before finally linking them (yellow arrow) with the light.*

samples is produced, the rendering function is evaluated for every sample and the average of all results converges to the true solution (Fig. 5). To evaluate a sample, the incoming light from one direction has to be computed. This is usually done using ray tracing: a ray is sent in the direction and the light emitted from the first hit point is computed, potentially, again by computing a solution of the RE at this location.

Instead of blindly shooting rays, many techniques exist to accelerate this process. A commonly employed method is importance sampling: Instead of sending rays everywhere, rays are sent where the rendering function's integrand (Eq. 1) (e. g., the illumination) has high values. This is easy for direct light, but more complex for indirect light, since it is unknown where the largest contribution originates from. Also, combining the individual factors of the rendering equation's integrand, such as BRDF f_r and incoming light L_i into one importance sampling approach is already difficult for direct light and even more for GI. One solution to this issue is bi-directional path tracing [LW93] (BDPT), the other is Metropolis Light Transport [Vea97] (MLT).

For interactive applications, fast CPU ray tracing [WKB*02] can be used. Based on upsampling of indirect light, interactive performance was demonstrated on a cluster of PCs for static scenes. Havran et al. [HDMS03] exploit temporal coherence by adapting their ray shooting and spatial acceleration structures to space time queries. Motion compensation and bi-directional path tracing are used to produce the individual frames. Multilevel ray tracing [RSH05] sends multiple rays at once, which could accelerate ray tracing-based final gathering. No GPU implementation of this algorithm or its application to GI have yet been demonstrated to our knowledge.

Using a GPU for ray tracing was demonstrated first by Purcell et al. [PBMH02] in 2002. While modern GPU ray tracers [ZHWG08, AL09, PBD*10] are able to cast around 100 M incoherent rays per second, the difficulty of mapping GI based on ray tracing to GPUs is the incoherency in bounced rays [BEL*07, AK10]. Boulos et al. [BEL*07] considered re-arranging incoherent rays to achieve interactive results. Wang et al. [WWZ*09] propose a full-GPU approach based on ray tracing, photon mapping and parallel clustering. Novak et al. [NHD10] address the special requirements due to tracing many bounces of incoherent rays. The special requirements for interactive ray tracing used in MLT and BDPT were recently considered by Pajot et al. [PBPP11] and van Antwerpen [vA11].

3.3. Photon Mapping

The next classic approach for interactive GI is Photon mapping [Jen96] (PM). This method works in two passes (Fig. 6). First, a large number of photons is emitted from the light source and bounced inside the scene based on ray tracing. At every hit point, a photon can be stored in a photon map and bounced further on. Second, the incoming light at every pixel is computed as a density estimate, i. e., locations at which the

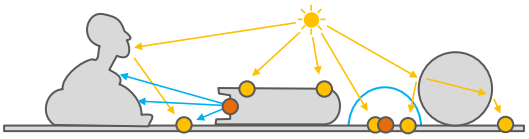

Figure 6: *Photon Mapping emits particles from the direct light (yellow arrows), that are bounced and finally stored (yellow circles). To compute the indirect lighting at a location (orange circle), final gathering (blue arrows) or density estimation (blue circle) is used.*

photon concentration is higher are brighter. Optionally, *final gathering* replaces the last step, and instead of performing a density estimate, for every pixel the incoming light is gathered from all visible surfaces. Because light is simulated both from the light source and from the eye, PM is well suited for all kinds of light transport, but excels in rendering caustics resulting from light paths like $\mathsf{LS^{+}DE}$.

PM can be adapted to achieve interactive results under some conditions. Early work on interactive PM by Purcell et al. [PDC*03] and Ma and McCool [MM02] allowed to compute GI at interactive rates. Both used spatial hashing to replace the nearest-neighbor queries required in density estimation, which fits better to fine-grained parallel GPUs. Linear-time intersection with all scene primitives is used to resolve visibility.

Using efficient GPU ray shooting and k-d tree building, Zhou et al. [ZHWG08] demonstrate efficient PM. A k-d tree is used both for tracing rays, as well as for density estimation. One attractive property of their approach is its similarity to the original PM idea, but using modern graphics hardware.

Instead of finding the nearest photons (gathering) it has shown beneficial to scatter (cf. Sec. 5.5) the photon distribution to the pixels they contribute to (e. g., [ML09]). This is done by drawing bounding volumes around every photon, that cover all pixel for which the density estimation's reconstruction kernel (e. g., Epanechnikov's kernel) has a significant value. For volumetric photon mapping, thick lines can be splatted [KBW06].

While classic photon mapping was proposed to emit photons using ray tracing, modern GPUs can achieve best results when replacing this step by rasterization [YWC*10]. The difficulty here, is to replace the infinitely-many secondary bounce sources by a finite set of camera positions. Again, their approach uses photon splatting.

Dmitriev et al. [DBMS02] invented selective photon tracing to enable dynamic scenes. Instead of pure random numbers, Quasi Monte Carlo sequences are used here to compute a photon's path. In a first step, so-called pilot photons are distributed in varying directions. Whenever a pilot photon hits a dynamic object, similar photon paths (corrective photons) can be determined quickly by generating similar Quasi random numbers. Günther et al. [GWS04] used this method for detecting caustic objects; a SIMD version was developed by Bärz et al. [BAM08].

To improve the quality of the caustics, Schjöth et al. [SFES07] presented photon differentials for splatting elliptical photons, which was used in a parallel CUDA implementation by Fabianowski and Dingliana [FD09], running at interactive frame rates.

A first approach to implement the time-consuming final gather step on the GPU was presented by Larsen and Christensen [LC04], but it is limited to small scenes. Point-based GI [Chr08] (PBGI) and Micro-rendering [REG*09] (Sec. 3.6) can also be used to perform more efficient final gathering. To this end, first photons are distributed in the scene, second a density estimate is performed at every PBGI-point to compute its lighting and third final gathering is used for final pixels using PBGI.

Image-space photon mapping [ML09] accelerates two key components of photon mapping: first, the initial bounce, and second, final gathering. For the first problem, reflective shadow maps are used [DS05] to emit photons from a single point light. All in-between photon bounces are computed on the CPU. Finally, for density estimation, a bounding volume is generated in a geometry shader to traverse all pixels that are influenced by the kernel density function, of that photon. The photon tracing step requires ray tracing and its acceleration structure, limiting the application to dynamic scenes and leading to a two-fold implementation effort.

Recently, a first GPU implementation of progressive photon mapping [HJ10] that overcomes certain scalability issues of the original technique and improves the reproduction of complex specular transport $\mathsf{LS^{+}DS^{+}E}$, e. g., caustics seen through a refractive surface was introduced. They avoid constructing irregular spatial data structures, which do not fit well to the balanced workload required for GPUs, for neighbor photon queries based on randomization. While their results are not yet interactive and the costly ray tracing-based photon shooting is not accelerated, its randomization approach to avoid irregular spatial data structures is appealing.

3.4. Instant Radiosity

Instant Radiosity [Kel97] (IR), similar to PM, is a two-pass technique (Fig. 7). In a first pass, photons are emitted and bounced inside the scene, similar to PM. The second pass however, is not based on density estimation, but on gathering instead. To this end, every photon is understood as a virtual point light (VPL) that emits light into the scene. To compute indirect light at a pixel, it simply has to be lit by every VPL. As the number of VPLs is small enough to compute a shadow map or shadow volume for each, this step can be substantially accelerated. In practice, the number of VPLs is at least one order of magnitude less than the usual number of photons.

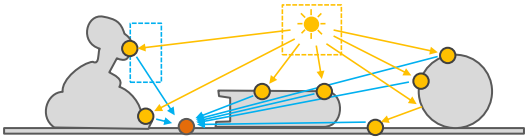

Figure 7: *Instant radiosity is similar to photon mapping, but instead of density estimation or final gathering, every stored photon becomes a virtual point light (yellow dot) that sends its light (blue arrow) to all receiver samples (orange point). It fits GPUs well, as both the photon emission (yellow dotted box) and the visibility between VPLs and sample points (blue dotted box) can use shadow mapping.*

Two notable extensions to IR are bi-directional Instant Radiosity [SIMP06a], and Metropolis IR [SIP07]. Here, the VPLs accuracy is improved in areas that contribute a lot to the final image and decreased in areas that contribute less. Effectively this leads to improved accuracy in less time.

Instead of using ray tracing, one-bounce photons can be distributed from a primary light source efficiently using *reflective shadow maps* [DS05, DS06] (RSMs). RSMs are produced using efficient GPU rasterization and store at every pixel the position, normal and color of the surface element visible in the direction of that pixel. A random subset of pixels of an RSM can now serve as VPLs. The ideas of RSMs and bi-directional importance were recently combined into bi-directional RSMs [REH*11].

However, RSMs do not provide a solution for secondary visibility: They allow to compute surface elements that send the first bounce, but for every VPL (hundreds) another shadow map is required for secondary visibility. To this end, Laine et al. [LLK07] propose Incremental IR. In their solution, VPLs are produced using RSMs but only a fixed subset of VPLs is replaced. A caching scheme is used to decide what VPL's shadow map has to be replaced. This approach allows for moving lights, but moving geometry leads to indirect shadows that lag behind the actual geometry. Depending on the frame rate and quality desired this can or cannot be acceptable.

Imperfect shadow maps [RGK*08] (ISM) exploit the fact, that a low-resolution shadow map for every VPL is sufficient. This is due to the low-pass nature of indirect light. While a low resolution like 32×32 is often sufficient, a high number of shadow maps (many hundred) is required. To save both transform time and fill rate that would visit the entire geometry for every triangle and filling at least one shadow map pixel, ISMs use a randomized point-based approximation of the scenes surface to be drawn into every shadow map. Every shadow map uses a different approximate point sampling to avoid the error to accumulate. The approximate point sampling is pre-computed, but deformed with the scene. In follow-up work this restriction was lifted [REH*11].

Instead of filling shadow maps with points to save fill-rate and transform time, such as ISM does, LoDs can be used to efficiently fill shadow maps for VPLs [HREB11]. Here, the difficulty lies in adapting existing LoD approaches that aim for a single primary view, to the requirements of extracting a high number of LoDs for a high number of VPLs. This is done by splitting the work into a fine-grained parallel cut-finding algorithm, combined with incremental LoDs.

One difficulty of IR is bias and clamping. The contribution of one VPL to a pixel involves a division by their squared distance. For VPLs and receiver pixels in proximity, this term becomes unbound. This is, because exchange between finite areas is replaced by infinitely small points with intensity instead of per-area quantities. The common solution is to just bound this term by clamping it to a small number. More advanced, screen space computations can be used to compensate for this bias [NED11].

Another extension of IR is the generalization of VPLs into either virtual area lights [DGR*09] or virtual spherical lights (VSLs) [HKWB09]. Dong et al. analyze a high number of VPLs and cluster them into virtual area lights (VALs). Visibility from one VAL can now be efficiently evaluated using soft shadow methods [ADM*08]. Radiance is still evaluated using random VPLs inside the cluster of one VAL. Virtual spherical lights do not aim to simplify and accelerate shadow computation, but avoid several of the bias and singularity problems found for virtual points lights. By deriving geometric terms for spheres instead of points, more robust lighting with less singularity artifacts can be computed.

VPLs on diffuse surfaces are Lambertian emitters. Some approaches experiment also with non-Lambertian materials leading to non-Lambertian VPLs [DS06,RGK*08,HKWB09]. Ritschel et al. [RGK*08] exploit the knowledge of the direct light's position to fit Phong lobes to allow Phong VPLs. For more general light paths, more advanced solutions will be required to produce such VPLs. Rendering light from such VPLs benefits from using VSLs [HKWB09].

3.5. Many lights-approaches

Many-light approaches are a variant of IR that abstracts from the way VPLs are produced. Instead, they are just called

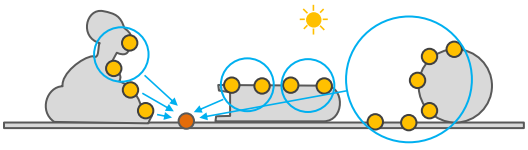

Figure 8: *Many-light approaches gather lighting for each sample (orange) from a large number of lights (violet) using a hierarchical spatial data structure (violet circles).*

"lights" which are assumed to be given. Besides indirect light,

also complex direct light such as local area lights or natural illumination from captured environment maps can be supported in one elegant framework. To compute GI from this representation, their contribution has to be gathered in sub-linear time with respect to the (high) number of lights (Fig. 8).

Classic IR solved the many-light problem by a simple linear-time for-loop over all lights. More advanced, Light cuts [WFA*05] builds a tree structure on the lights and instead of gathering from every individual light (a leaf node), illumination can also be gathered from a group of lights (an internal node) in $O(\log(n))$. The set of nodes that contribute to a pixel forms a cut in the light tree. Every node in the tree stores spatial and normal bounds for all nodes below it. The cut is found by comparing a node's bounds to a user-specified threshold: all nodes which are below the threshold, but their parent ist not, form the cut. However, visibility based on ray tracing has to be evaluated for every element in the cut and remains the computational bottleneck.

Finding a cut for every receiver can be costly and doesn't map well to GPUs. Instead, Hasan et al. [HPB07] propose to find a subset of rows and columns in a matrix of sender-receiver relations that can be computed by GPU rasterization to solve light transport involving many lights. This avoids the costly ray tracing. Despite the theoretical analysis made in their paper and convincing results for many scenes, it is not obvious that rasterizing a low number of row or column maps is sufficient for all scenes. As an extension, (tensor) clustering over multiple frames can be applied to known animations [HVAPB08]. Cuts form an interesting computational primitive, that can be pre-computed and re-used [AUW07], drawing a connection between many-light or cut-based and PRT approaches.

3.6. Point-based GI

Point-based GI [Chr08] (or independently "Micro-rendering" [REG*09]) is a relatively new approach to GI. It shares some ideas with many-light approaches and FE, but has also fundamental differences, e. g., the way visibility is treated.

It works in three passes (Fig. 9). In a first pass, the scene surface is approximated using lit surfels (the "points") on which a hierarchy is computed. Different from many-light approaches, the points also cover unlit surfaces. Every interior node in this hierarchy stores bounds for the light and the geometry it contains, similar to Light cuts. The hierarchy can be established in a pre-process if the scene undergoes only moderate deformations (coherency). Next, the points (which are the leaves of the hierarchy) are updated, i. e., they are shaded and the inner nodes of the hierarchy are updated. Third, for every pixel, the points are rasterized into a small framebuffer for every pixel. This resolves indirect visibility. Instead of rasterizing all points sequentially, a sub-linear Q-Splat rasterizer using LoD is used [RL00]. The rasterizer

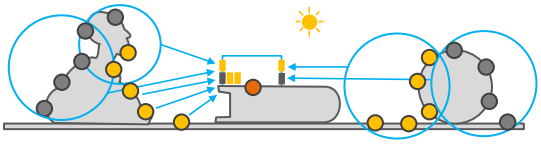

Figure 9: *PBGI is similar to many-light approaches, but uses a hierarchical spatial data structure (blue circles) on all geometry, not just on lights. Doing so, the entire surface is sampled into lit points (yellow/grey for lit/unlit points), which serve as lights and occluders at the same time when being gathered (blue arrow) into a micro-framebuffer (blue box), including resolution of occlusions (blue/grey splats for lit/unlit directions).*

effectively enumerates a pixel-sized node cut, i. e., the set of nodes for which for every node, its parent is bigger than a pixel and all of its children are smaller than one pixel.

Multiple bounces and final gathering for photon mapping can also benefit from this approach. Different from many-light approaches, PBGI is a full approach and not only a solution for final gathering.

Micro-rendering [REG*09] uses a GPU to perform all rasterizations for all pixels in parallel. Furthermore, warping is used to improve precision for specular surfaces: Instead of drawing into an evenly discretized frame-buffer and multiplying with the BRDF, the points are drawn into a non-evenly discretized frame-buffer stretched according to the BRDF and the multiplication is omitted. While the algorithm is parallelized well, finding the cuts recursively in each thread is costly. This high per-thread complexity leads to code and data divergence. Instead, modern GPUs prefer more fine grained operations that trigger more parallel threads with less divergence [HREB11].

Maletz and Wang [MW11] replace the recursive rasterization of pixel-sized nodes by a two-fold stochastic approximation. Instead of finding the exact cut, they group pixels into a much lower number of clusters. First, for every cluster, the importance, i. e., how many pixel-sized nodes it contains, is approximated using a few random samples. The importance of all clusters is interpreted as a probability density function (PDF). Second, a random cluster is selected with a probability proportional to the PDF and finally a random leaf node inside this cluster. Their approach fits well to modern hardware, as it does constant work and parallelizes well, however, the multiple approximations made, might not hold in some cases, i. e., corners, where a low number of points dominates, resulting in under-sampling of the true PDF.

3.7. Discrete ordinate methods

The evolution of light transport with participating media, including phenomena such as light absorption, scattering, and

emission effects, is formalized as the Radiative Transfer Equation (RTE). Discrete ordinate methods (DOMs) [Cha50] discretize the RTE in space and orientation (Fig. 10). Essentially,

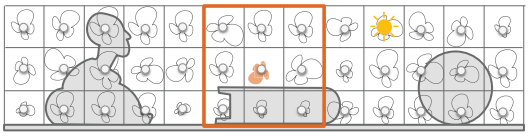

Figure 10: *Discrete Ordinate Methods discretize the scene spatially and directionally by computing light transport based on exchanging energy between neighboring grid cells.*

the light transport is modeled as energy exchange between neighboring volume elements, which is a beneficial property, considering that only local operations are necessary to compute light transport. However, this simplicity comes at a cost and DOMs suffer from two artifacts: first, the energy propagation between neighboring cells requires repeated interpolation, which does not allow light beams to maintain sharp profiles. Second, the angular discretization allows light to only travel at a finite number of directions (hence the name of the method), causing the so-called ray effect that is particularly noticeable in media where no or little scattering takes place. Nevertheless, DOMs are widely used by radiative transfer engineers [Coe04, LBC94] and in atmospheric radiative transfer [Eva98] (where these effects are not as severe as for light transport in vacuum for example). Evans [Eva98] describes Spherical Harmonics DOMs (SHDOMs) where the directional distribution of light within every cell is stored as a spherical harmonics approximation. Perez et al. [PPS97] refer to some early adoptions in computer graphics. Geist et al. [GRWS04] postulate a simple photon transport model that describes a diffusion process to compute light transport in participating media based on the Lattice-Boltzmann method and derives a method similar to the DOMs. Fattal [Fat09] describes an improvement of DOMs to reduce light smearing and ray effects by propagating light across the domain by solving the RTE along light rays with similar directions of propagation. These rays are stored in temporary two-dimensional Light Propagation Maps (LPMs), which contain only a fraction of all possible propagation directions. By this, the traditional transport between adjacent cells based on the discrete variables is avoided. Recently, Kaplanyan and Dachsbacher [KD10] described a variant of SHDOMs for realtime rendering in dynamic scenes. They use reflective shadow maps [DS05] to sample directly lit surfaces and initialize the simulation domain for computing indirect illumination. Large scenes are handled using a cascade of grids for the light transport simulation.

3.8. Pre-computation

Ideally, in interactive GI, the scene elements (geometry, reflectance, light and view) can be changed freely. When assuming some of those components fixed, significant simplifications can be made. Furthermore, other simplifications are made, such as distant lighting.

Relighting keeps the camera and geometry fixed and allows to change the lights and materials [JSN94, GH00, PVL*05, HPB06, RKKS*07]. Originally, relighting enabled lighting changes by computing a linear combination of prelit basis images [JSN94]. The original idea of storing the required information in a deep framebuffer was later presented by Gershbein and Hanrahan [GH00] and further extended to production scenes by Pellacini et al. [PVL*05] and Ragan-Kelley et al. [RKKS*07]. While these approaches compute direct light only, Hasan et al. [HPB06] consider direct-to-indirect transfer. To this end, they cache the contribution of the direct light arriving at a number of scene sample points to every pixel. When the direct light is changed, the indirect light can be updated interactively. However, the direct-to-indirect matrix can be huge.

Pre-computed radiance transfer (PRT) methods originally make the assumption that the scene's geometry is fixed (only light and view can change) (Fig. 11). Under this assumption

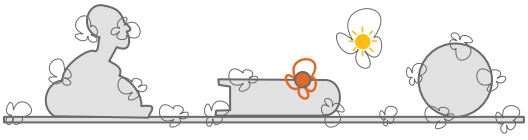

Figure 11: *Precomputed Radiance Transfer stores a compressed directional response function to an environment map (orange line) on the surface.*

parts of the light transport can be precomputed enabling real-time rendering of complex light transport effects (diffuse and glossy shading, indirect illumination, caustics, subsurface scattering, etc.).

The first PRT method proposed by Sloan et al. [SKS02] used spherical harmonics (SH) to store the transfer function (comprising diffuse or Phong shading as well as visibility) and the lighting function (environment maps). At run-time, it was then only necessary to compute the dot product between the coefficient vector representing the transport and the coefficient vector representing the lighting resulting in a very fast algorithm. The resulting quality is directly related to the number of basis functions used. However, even a few spherical harmonics basis functions yielded visually pleasing results (overly smooth shadows). Kautz et al. [KSS02] extended the work to allow for general BRDFs (using a 4D SH representation of each BRDF). Due to the required local rotation, the method was not quite real-time at the time.

Many other basis have been explored in order to represent light transport in the PRT context. Non-linear wavelets

were shown to work well (i. e., , interactively) for shadows and direct diffuse illumination [NRH03]. Unlike the SH basis, only the wavelet basis functions with large coefficients were retained yielding a non-linear basis, which in turn made GPU implementations more difficult. Real-time rendering of glossy BRDFs with wavelet basis was proposed by Liu et al. [LSSS04] and Wang et al. [WTL04, WTL06] by using separable approximations for the BRDFs [KM99]. Ng et al. [NRH04] and Sun et al. [SR09] have extended triple product integrals using basis functions to wavelets, thus also enabling all-frequency PRT.

Furthermore, Gaussians [GKMD06, GKD07] and radial basis functions [TS06] have also been used to represent light transport, yielding good results. Tensor representations for the BRDF have led to precomputed transfer tensors [SZC*07, SZS*08], enabling relighting with dynamic BRDFs as well as dynamic refractive objects. Piecewise constant functions have also been used as transport representation [XJF*08]. Finally, meshless hierarchical basis functions can be used [LZKF08], allowing to easily decouple the transport signal from the underlying vertex or texture resolution.

Light transport is often locally coherent in space, allowing for effective compression. Techniques include clustering [SHHS03], PCA [LK03], or, for instance, using a spectral mesh basis [WZH07]. Further extensions of PRT to handle normal maps [Slo06], subsurface scattering [WTL05], local lighting [AKDS04], soft GI [SGNS07] and portals [YID*09] have been shown. One particular use case that fit well in the PRT framework is the editing of BRDFs for material design with high quality interactive feedback [BAOR06, SZC*07, CPWAP08]. Instead of storing the full radiance transfer, it is also possible to store visibility only, and then use it to accelerate illumination. This was demonstrated for macro-geometry [HDKS00], as well as full scenes [RGKM07, YIDN07]. This is related to the many light approaches using cuts, e. g., [AUW07, CPWAP08].

The assumption of static geometry and static lighting, allows to precompute incoming light at a location in space. Animated characters can then be re-lit moving through space [GSHG98]. Similarly, ambient occlusion can be precomputed and stored in discrete grids to cover individual objects [KL05] that can undergo rigid transformations. Or more general, entire radiance transfer can be pre-computed for static objects that can then undergo rigid deformations [Pan07].

If the degrees of freedom of a scene are known, they can be sampled leading to a higher-dimensional pre-computed transport. In this spirit, Ambient Occlusion (AO) was precomputed for character skinning [KA06a, KA06b], and even the entire radiance transfer for deforming objects [SLS05]. If the transformations are rigid, then partial precomputation can be used to speed up rendering, for instance, coherent shadow maps [RGKM07, RGKS08] and triple product wavelets [SR09]. It has been also shown that some fully dynamic objects can be included in a PRT framework by re-computing visibility on the fly [KLA04].

3.9. Caching

Caching in the context of GI means to compute results sparsely and interpolated in-between. Caching is mostly orthogonal to the particular GI method used and should be compatible to many of the before mentioned approaches. Previously it was mainly used in combination with Monte Carlo ray tracing however (Sec. 3.2).

In classic irradiance caching [WRC88, TPWG02], irradiance is not computed for every pixel, but only for some pixels according a criterion based on the average distance to other surfaces. In irradiance volumes [GSHG98, NPG05], irradiance is sampled in a regular grid in space. An irregular version of irradiance volumes is the irradiance atlas [CB04]. Arikan et al. [AFO05] decompose GI in nearby effects and distant effects and use different procedures for either one in offline rendering. Similar ideas are used in interactive GI, e. g., when combining screen space-illumination for nearby and other approaches for far fields [KD10, REH*11].

4. Phenomena

While the previous section listed known approaches to interactive GI and suggesting some strengths and weaknesses, this section will list a range of important GI phenomena and discuss the strengths and weaknesses of applying a certain technique to a certain phenomenon.

Traditionally, early approaches to shading [Pho75] did account for missing indirect light using an ambient term. There are several popular shading techniques that reproduce the appearance of particular GI effects such as soft shadows [ESAW11] e. g., using percentage-closer soft shadows [Fer02] or the image-based modeling of (glossy) reflections e. g., by using pre-convolved environment maps [HS99]. This text will not detail such approaches, acknowledging their importance in existing, contemporary interactive applications such as games.

We will call all light paths of the from $\mathsf{L}\{\mathsf{D}|\mathsf{S}\}^{++}\mathsf{E}$, the *indirect light*. The technical difficulties found when computing complex direct lighting are similar to those found when dealing with indirect lighting. Therefore we will also consider complex direct light as GI. We will however differentiate between single diffuse and glossy bounces, and multiple bounces.

This section starts its exposition with Ambient occlusion (Sec. 4.1), consider natural illumination (Sec. 4.2), single bounces (Sec. 4.3), caustics (Sec. 4.4) and multiple bounces (Sec. 4.5) in detail and finally discuss different forms of scattering (Sec. 4.6).

4.1. Ambient occlusion

The remarkable importance of occlusion as a visual cue was first described by Langer and Zucker [LZ94]. They found that the human ability to understand shape from shading combines common directional lighting and shadows, as well as soft ambient lighting and *ambient occlusion* (AO) found on a cloudy day.

Miller [Mil94] introduced accessibility shading that was modified by Zhukov et al. [ZIK98] to AO. Since AO is a view-independent effect it can be pre-computed for static scenes and finally be stored in textures or at vertices [PG04].

Bunnell [Bun05] computes AO in dynamic scenes using a FE approach that approximates the surfaces as a hierarchy of discs. AO is gathered at vertices or pixels in a fast GPU multipole approach. The incorrect linear summation of visibility of different occluders can be improved with a multi-pass approach. Later, other proxies such as spheres were used [SA07, CK10].

Rigidly transformed static objects can benefit from a pre-computation that stores AO in a discrete 3D grid [KL05, MMAH07]. For character skinning, AO can be pre-computed for all poses and blended [KA06a, KA06b].

Screen space methods [Mit07b, SA07, HJ07, BS08, FM08, BS09, LS10, PMP10, SKUT*10] are very popular to compute AO. Mittring [Mit07b] first described the use of the depth buffer to compute AO, as done in the game Crysis. Shanmugam and Arikan [SA07] use a similar approach but combine screen-space occlusion from nearby objects with occlusion from spherical proxy geometry for far-field occlusion. While some approaches simplify ambient occlusion to count the number of blockers, horizon-based AO [BS08], finds the horizon line, that allows to compute a more accurate occlusion. Volumetric approaches use a slightly different definition of AO, where occlusion is the ratio of blocked and unblocked volume in a sphere around a sample [LS10, PMP10, SKUT*10, RSKU*10]. Doing so achieves a more stable result which is less prone to several screen-space artifacts. Instead of gathering the occlusion from nearby occluders, another option is to splat (Sec. 5.5) the occlusion using a spherical proxy [SA07] or prismatic volumes extracted from triangles [McG10].

4.2. Natural illumination

The change of light with varying directions at a fixed location in space is called the light probe, or the *distant* lighting. In several situations, it is a valid assumption that light does not change over space, but only directionally. The distinct variations in contrast and hue are called *natural illumination* [Deb98, DLAW01]. Natural illumination can be captured using a reflective sphere (light probe) and appropriate camera equipment [Deb98] which is finally stored as floating point HDR images.

Given an HDR video camera, animated natural illumination is applied to a scene using many lights as proposed by Havran et al. [HSK*05]. They importance-sample the environment map into a fixed number of point lights and compute a shadow map for each. Diffuse and glossy light paths L{S|D}E are demonstrated but glossiness is limited due to the decomposition in individual lights. Furthermore, this allows a seamless integration of correctly-lit virtual objects in real camera images, as shown in [GEM07, KTM*10].

Alternatively, a lower number of shadow maps is used by Annen et al. [ADM*08]. However, instead of splitting the environment map into point lights, it is decomposed into area lights in a pre-process. Then, a fast soft shadow-method is used to cast shadows from such area lights. Dong et al. [DKTS07] perform this decomposition interactively, allowing for real-time changes of lighting. Only LDE paths are demonstrated.

Ren et al. [RGW*06] use proxy geometry to compute an approximate directional visibility function. Instead of expensive SH multiplication, the log of SH coefficients are summed and only exponentiated in the end, arriving at visibility which can then be combined with the material and light SH. Due to the smoothness of the SH representation used, only L{D|S}E paths with low gloss are possible.

Sloan et al. [SGNS07] use spherical proxy geometry that is accumulated in screen space to efficiently cast shadows from natural illumination. While their method both provides pleasant natural illumination and is elegant and fast, the results inherently remain overly soft.

4.3. Single bounce

Single-bounce diffuse inter-reflections are paths of the form LDDE, that is, light hits a diffuse surface and then another diffuse surface before arriving at the eye. Because every inter-reflection is a convolution with the BRDF, it is also a low pass. After several bounces, light has diffused and varies more and more smoothly, which is exploited by many approaches.

As diffuse-only light paths are view-independent, they can be pre-computed for static scenes e. g., using FE, MC or photon mapping and stored in vertices or textures. For dynamic scenes, FE have been used [Bun05, DKTS07, DSDD07] to simulate diffuse bounces in dynamic scenes. However, the complexity of meshing and the inflexibility of linking limits these approaches to moderately-sized scenes. Later, combinations of FE and MC in screen space [RGS09] and hierarchical FE [NSW09, SHR10] were used to compute approximate solutions for large scenes. Instant radiosity [KH01] can display diffuse bounces interactively, if the VPLs are distributed efficiently [DS05, LLK07, RGK*08]. Sloan et al.'s [SGNS07] proxy accumulation also allows for one soft bounce of indirect light, but without visibility. PRT [SHHS03] can reproduce well diffuse indirect lighting, due to its smooth nature. Amongst the several restrictions of PRT, the one to distant

lighting is the most problematic one: the most prominent GI is achieved for local light sources, and can not be reproduced by distant light. Same as for direct light, indirect light can be blocked and only illuminates the first surface it hits after being reflected. As a simplification, this is sometimes ignored [DS05], updated incrementally [LLK07] or approximated [RGK*08]. Glossy indirect light of the form LDSE, is often easier to integrate by using a glossy BRDF at the final receiver. With IR, individual VPLs can become visible for high glossiness and material perception can change in an undesired way [KFB10]. PRT [SHHS03] also supports glossy transfer but usually blurs the result.

Bouncing light is perceptually important and its perception starts to become more important in interactive applications (Sec. 5.7). Gilchrist [Gil79] first put GI in a perceptual context when relating the perception of light to geometrical configuration: "What shade of gray a surface appears is related to the perceived distribution of light and shadow, which in turn depends on the perceived spatial relation between the surface and its neighbors". Later, Kersten and Hurlbert [KH96] demonstrated, how discounting for the color of mutual illumination helps to disambiguate geometry. Langer [Lan99] found a strong perceptual interplay between the cues provided by (Lambertian) shading, shadows and (multiple) interreflections. Thompson et al. [TSS*98] have shown how shadows and interreflections can act as "visual glue" that makes objects appear to be in contact with a ground plane. One of the conclusions made is, that the presence of cues is usually much more important than their correct simulation. In theoretic work, Chandraker et al. [CKK05] show, that even the (generalized) bas-relief ambiguity (infinitely many Lambertian surfaces illuminated by distant point lights result in the same image [BKY99]) can be resolved using mutual illumination. Some results indicate that low-level cues and simple image statistics [FJB04] are important for material perception. In conclusion, bounces are not "eye candy" but indeed contain important additional information that is provably perceived and exploited by the human visual system.

4.4. Caustics

Caustics are light paths of the form LSDE, that is, light is first bounced specular and then diffuse. In general, all light paths with at least one specular segment followed by one indirection (often diffuse), i. e., $\mathsf{LD^*S\{D|S\}^+E}$ are caustics. Caustics are perceptually important when depicting fluids or glass.

Because the last bounce is diffuse, caustics are a view-independent phenomenon, that can be pre-computed. It has a complex light path, but still it's view-independent.

In early work, Wand [Wan03] compute caustics by voxelizing a reflector or refractor surface and interpreting every voxel as a pinhole camera that projects lighting onto receivers. Notably, the local geometric structure, e. g., the curvature, is taken into account to appropriately filter the cameras. Caustic light is assumed to be unshadowed.

For dynamic scenes, caustic mapping [IDN03, BO04, EAMJ05], and its hierarchical [Wym08], and adaptive [WN09] extensions were proposed. They do not consider indirect visibility, i. e., caustics are never blocked.

A simpler form which can be approximately handled using non-Lambertian VPLs by IR are light paths of the form $\mathsf{LS\{S|D\}E}$, that is, light that is reflected once by a specular surface and then again by a specular or diffuse surface. Same as for diffuse direct light, indirect light can be blocked and only illuminates the first surface it hits after being reflected. As a simplification, this is sometimes ignored [DS06] or approximated [RGK*08].

Due to the complicated light path, splatting techniques [WD06, DS06, KBW06, SKP07, YLY07, Wym08, WD08] are especially useful to display caustics. Alternatively, photon impacts can be counted directly in a photon texture [HQ07]. The use of gathering from many non-Lambertian VPLs is inefficient since most of them have a neglectable contribution.

Umenhoffer et al. [UPSK08] propose to replace photon mapping that maps points to hit locations by a mapping of triangles to surfaces. This successfully avoids several undersampling artifacts but requires the scene configuration to be simple, i. e., caustics are produced by single triangles that then map to diffuse surfaces. Multiple caustics are not possible and visibility is ignored.

For dynamic height field-like water surfaces that cast caustics on a planar ground receiver closed solutions that allow for efficient GPU implementations exist [YK09].

PRT can reproduce all light paths including caustics. But caustics are high-frequency effects and PRT acts as a low-pass filter, so only smooth caustics can be reproduced and sharp discontinuities are not possible.

4.5. Multiple bounces

Multiple diffuse bounces ($\mathsf{LD^{++}E}$) are light paths of particular interest, e. g., in architectural visualization. Classic radiosity can do them well, and so does (interactive) FE [DKTS07, DSDD07]. Path tracing and Photon mapping are also well-suited for multiple bounces. Instant radiosity also supports multiple bounces of all forms. However, reflective shadow maps only support the first bounce [DS05]. Further bounces become efficient when using imperfect reflective shadow maps [RGK*08]. Many-lights approaches such as light cuts [WFA*05] are agnostic to the way the many lights are produced and therefore support multiple bounces naturally.

More general light paths $\mathsf{L\{D|S\}^{++}E}$ are difficult and rarely addressed in interactive GI. FE approaches that support single-bounce glossy support are usually easy to extend to multiple

glossy bounces, but at the expense of additional computation for each bounce. Monte Carlo ray tracing per definition can render all light paths, but with increasing complex lighting, an impractical number of rays is needed. The more complex the lighting gets, the more rays (threads) diverge and the performance (on GPUs) drops. Approaches based on Photon mapping can do multiple bounces well, but high-frequency transport requires a high number of photons, ideally using Progressive Photon Mapping [HJ10]. PRT supports multiple glossy bounces only at the cost of longer precomputation. At runtime, performance remains high, but results are blurred. Many-lights approaches and PBGI are targeting final gathering. Their use for in-between bounces has not yet been described. Laurijssen et al. [LWDB10] propose a method to handle multiple glossy bounces (indirect highlights) using Gaussian lobes to represent directional transport.

4.6. Scattering

Light that is traveling through participating media (most GI methods assume the surfaces to lie in vacuum) is scattered. Simulating this process accurately is a complex problem, and consequently there are methods focusing on special cases such as single and multiple scattering, subsurface scattering (SSS, usually referring to methods computing scattering light within objects made of highly scattering material such as marble), or participating media (fog, haze, etc.) rendering with homogeneous or non-homogeneous scattering parameters.

Subsurface scattering Participating media rendering is very expensive and thus there exist a variety of methods that focus on rendering (individual) translucent objects and simulating the scattering in its interior. In principle, light impinging on any surface location of the object can travel to any other location of the surface. Analogous to BRDFs, where interaction involves only one surface point and two directions, this process can be characterized by the *Bi-directional Scattering Reflectance Distribution Function* (BSSRDF) [JMLH01]. Note that the BSSRDF also depends on the geometry of the object and the material (also in its interior) and thus, generally, cannot be specified object-independent. This is only possible for special cases, e. g., for an infinite plane and highly scattering material the dipole-approximation can be used [JMLH01]. It is frequently used in many interactive GI methods that render objects of such materials, although their geometry violates the dipole assumption.

Most methods render highly scattering materials. Jensen and Buhler [JB02] compute incident radiance at randomly distributed samples across surfaces and integrate hierarchically using the dipole model. Lensch et al. [LGB*03] separate the scattered light into a global part (long distance transport via precomputed vertex-to-vertex transfer) and a local part approximate by a filter kernel in a texture atlas. Mertens et al. [MKB*03a] formulate SSS as a boundary element method similar to hierarchical radiosity and achieve interactive rendering. Dachsbacher and Stamminger [DS03] observe that directly lit surfaces are captured in shadow maps and store additional information (surface position, normal, etc.) in Translucent Shadow Maps (TSMs) such that the dipole model can be evaluated. A hierarchical integration scheme using mip-maps enables real-time rendering of translucent objects. The overhead for creating TSMs is negligible if shadow maps are rendered anyway, but the method is not applicable to natural lighting (Sec. 4.2).

Approximations in screen space, i. e., the scene as seen from the camera, have proven to be very efficient yet yielding plausible results. Mertens et al. [MKB*03b] gather incident light by importance sampling the BSSRDF in image space. Jimenez et al. [JSG09] approximate SSS for skin rendering by a screen-space diffusion approximation and validate their results by psychophysical experiments. Several methods resort to texture atlases instead of screen space to first compute the incident light and subsequently apply image filters that mimic SSS. These approaches sample all surfaces (visible to the camera or not), but require a parameterization of the surface; often surfaces close in world space are distant in texture space resulting in too dark renderings. D'Eon et al. [DLE07] render SSS for human faces using a cascade of Gaussian filters, which are fitted to the dipole BSSRDF. Note that there are several other methods and variants following these ideas and mentioning all of them is beyond the scope of this survey.

Depending on the material properties, the single scattering contribution can contribute significantly to the appearance of an object. Jensen et al. [JMLH01, JB02] compute single scattering separately using ray tracing. Walter et al. [WZHB09] derive an efficient technique for computing single scattering by computing light paths that connect points inside and outside a medium, bounded by a triangle mesh.

PRT can also account for scattering [SKS02, WTL05] yet with the typical limitations such as static geometry, often low frequency transport and distant illumination.

Participating media Another class of methods deals with the case that a scattering medium surrounds the objects in a scene. Cerezo et al. [CPP*05] provide an overview of non-interactive methods which deserves a survey on its own and is beyond the scope of this article. As these media are often not very dense and the single scattering contribution dominates, many interactive methods focus on this phenomenon.

Several approaches are based on shadow volumes to determine regions of illuminated and shadowed space in front of visible surfaces and thus compute light scattered towards the viewer. To this end, shadow volume polygons must be sorted back-to-front causing additional sorting cost [Max86, BAM06, Mec01, Jam03]. Gautron et al. [GMF09] compute light cones (instead of shadow volumes), and the intersection of an eye ray and a cone determines the length of the eye path through the light. Wyman and Ramsey [WR08] render inscattering from textured spot lights using ray marching

also using shadow volumes. They also use naïve image space sub-sampling to reduce the number of rays. Tóth and Umenhoffer [TU09] propose to use interleaved sampling in screen space, computing the inscattered light using ray marching and shadow mapping. Dobashi et al. [DYN02], Mitchell [Mit04], and Imagire et al. [IJTN07] use slicing techniques, known from volume rendering, to render volumetric shadows by rendering quads parallel to the image plane at varying distances from the eye. These slices are accumulated back-to-front using blending, while the shadows on each slice are computed using shadow mapping. Also simple post-processing techniques [Mit07a, Sou08], have been used to mimic the effects. Although very fast, these approaches suffer from inherent limitations including non-textured lights and false light shafts.

Max [Max86] computes single-scattering by intersecting shadow volumes with epipolar slices. Similarly, Billeter et al. [BSA10] generate shadow volumes from shadow maps and use the GPU rasterizer to compute the lit segments. This yields fast rendering for low-resolution shadow maps, but does not scale well with complex scenes requiring high-resolution shadow maps to capture details. Epipolar geometry is also used by Engelhardt and Dachsbacher [ED10] who speed up ray marching by detecting and sampling depth discontinuities along epipolar lines in screen space. Baran et al. [BCRK*10] use partial sum trees in epipolar coordinate systems for computing the scattering integrals in a single-scattering homogeneous medium. Chen et al. [CBDJ11] extend this work such that a GPU-friendly computation based on min-max mip-maps can be used. Note that for homogeneous media closed-form solutions exist if volumetric shadows are neglected, e. g., see [SRNN05, PSP09, PSP10, PSS11].

Visually interesting and challenging effects due to scattering are volume caustics. Hu et al. [HDI*10] create volumetric caustics by rendering large numbers of light rays as lines. Krüger et al. [KBW06] trace photons via layered depth images and accumulate their contributions in screen space to render surface caustics and light shafts. Liktor and Dachsbacher [LD10, LD11] extend the work of Ernst et al. [EAMJ05] and describe a GPU-friendly beam tracing variant enabling high-quality volumetric caustics in real-time. A more generic approach is Eikonal rendering [IZT*07], where light wavefronts are tracked over time in a precomputation step. It can account for complex material properties, such as arbitrarily varying refraction index, inhomogeneous attenuation, as well as spatially-varying anisotropic scattering and reflectance properties. Also requiring preprocessing, Cao et al. [CRGZ10] render non-constant, refractive media using the ray equations of gradient-index optics. Sun et al. [SZS*08] convert surfaces to volumetric data, and trace the curved paths of photons as they travel through the volume. Their method enables interactive rendering of static geometry rendering on contemporary GPUs.

5. Strategies

This section will summarize several key strategies that are found to have worked well when computing interactive GI: Rasterization (Sec. 5.1), Screen space (Sec. 5.3), Surfels, Hierarchies and Clusters (Sec. 5.4), Splatting (Sec. 5.5), Upsampling (Sec. 5.6) and accounting for human perception (Sec. 5.7).

5.1. Rasterization

GPUs are extremely efficient in rasterizing a high number of polygons, potentially involving complex deformation, tessellation and material maps into frame buffers, such as the one for the primary view or different forms of shadow maps. Therefore, many interactive GI applications make use of rasterization.

Most interactive GI applications are based on deferred shading [ST90], i. e., first a buffer with positions, normals and shading attributes such as diffuse colors is filled. Based on the combination of this buffer and other such buffers, GI is computed. Another useful buffer results from rendering the scene from the direct light's point of view into a reflective shadow map [DS05].

Alternatively, rasterizations based on points, such as in imperfect shadow maps [RGK*08, REH*11] can be used. Points are extremely efficient to rasterize, especially if only a low resolution is required. Also the photon emission [ML09] and tracing phase [YWC*10] which were traditionally based on ray tracing can be replaced by rasterization. Other, more involved forms of rasterization include hemispherical rasterization [KLA04] or voxelization [ED08]. Voxelization can be used to compute AO [RBA09] or full GI [THGM11]. For very low image resolutions, such as in PBGI, specialized rasterization is required [REG*09].

Matrix-column-row sampling [HPB07] puts a many-lights approach in the context of efficient rasterization. They replace the general ray-scene intersections by computation of light (row) and shadow maps (columns) that can be rasterized efficiently.

The alternative to finding first hits using rasterization is ray tracing, where modern GPU ray tracers [ZHWG08, AL09, AK10] are able to cast more than 100 M rays per second.

Niessner and colleagues [NSS10] combine rasterization and ray tracing the other way around. First they first rasterize the scene into a layered depth image (LDI) that stores for every pixels not just the position, normal and color of the first visible surface element, but the same information for all other surfaces behind. Second, Monte Carlo ray queries are approximated by traversing the LDI allowing for diffuse and moderately glossy one-bounce transport. Their implementation does not support dynamic scenes, because the LDI generation is implemented using depth peeling.

5.2. Importance sampling

Importance sampling is a classic approach to accelerate Monte Carlo ray tracing. To this end, the rendering function is evaluated more finely where its value is high and more coarsely where it is low. The difficulty is, that the rendering function's values are not known beforehand and can only be guessed or bound.

For distant illumination, Structured Importance Sampling [ARBJ03] was proposed as an offline technique, but later also used in interactive methods [HSK*05]. For unstructured light (point lights) Wang and Akerlund [WA09] propose an approach to perform bi-directional importance sampling, i. e., sampling visibility with a distribution of rays proportional to the product of incoming light and BRDF.

Finding VPLs that all have an equal contribution to the final image is importance sampling as well. When selecting the pixels from a reflective shadow map [DS05] to convert into VPLs, importance sampling is used [RGK*08]. Similar ideas use bi-directional importance based on ray tracing [SIMP06a, SIP07] and reflective shadow maps [REH*11]. Decomposing environment maps into area lights with equal energy is such an approach, e. g., like Annen et al. [ADM*08].

5.3. Screen-space

Instead of using the original polygonal geometry for computing the light transport, a deep framebuffer can be used. This buffer contains all the information that we require for each pixel: position, normal and material. This allows a coarse reconstruction of the surrounding geometry around a pixel, independent of the number of polygons. First, this was used to compute real-time ambient occlusion for dynamic scenes [BS08, BS09, PMP10, SKUT*10, LS10]. Later, additional GI [SGNS07, RGS09, NW09, NSW09, NPW10] effects were simulated in image space: directional occlusion and indirect light. Finally, also subsurface scattering was computed in screen space [MKB*03b, JSG09].

5.4. Surfels, Hierarchies and Clusters

While classic approaches to GI were based on using the input triangle meshes as their computation domain, points or surfels play an increasingly important role. Surfels were used as senders, such as in photon mapping [Jen96], or instant radiosity [KH01], as receivers [Bun05, LZKF08, RGK*08] or both [Chr08, REG*09]. Notably, the aforementioned screen space is a surfel cloud as well. It has the drawback of being incomplete due to clipping and occlusion, but the plus of being very easy and efficient to generate.

Instead of computing all the information at a single level, inspired by classic hierarchical radiosity [HSA91] and the sub-linear solution to n-body-problems, many current approaches use several resolutions of surface representation organized in a hierarchy.

A disk hierarchy is used by Bunnell [Bun05], Christensen [Chr08] and Ritschel et al. [REG*09]. Anti-radiance [DSDD07], as well as implicit visibility [DKTS07] use a pre-computed hierarchy of links. This restriction of the latter was removed by Meyer et al. [MESD09]. Imperfect shadow maps [RGK*08] and micro-rendering [RGS09] sample the surface into disks and assume coherency and area-preserving deformations. This limitation of Imperfect Shadow maps has recently been resolved [REH*11] by adaptively sampling the scene surface into disks where they are required most.

Points as simple primitives are well suited for clustering. Clustering considers proxy elements as placeholders for all individual elements in a group of elements. Such a grouping forms a cut in a tree: Elements in the cut are placeholders, everything above is too coarse to give a sufficient approximation, everything below is finer than required. Lightcuts [WFA*05] introduced this notion to GI, which was later applied to PRT [AUW07] and VPLs [DGR*09] cluster. Different from finding cuts in light, cuts in the geometry are found in PBGI [Chr08, REG*09, HREB11], but using similar computations.

5.5. Splatting

Instead of finding all sender elements that contribute to a receiver, it can be useful to rewrite this process into splatting. To do so, every element has to bound its spatial region of influence, that is, it must be able to efficiently enumerate all pixels it will contribute to. This can be done, for example, by drawing a bounding area such as a quad or a bounding volume such as a box or sphere. Doing so avoids building a k-d tree and the n nearest neighbor-queries required by the original method.

For FE, splatting was first used by Stürzlinger and Bastos [SB97] and later for photon mapping [LP03, ML09, YWC*10], radiance caches [GKBP05], VPLs [DS06], occlusion [SA07, McG10] as well as for basis functions in FE [DSDD07] and PRT [LZKF08].

One drawback of splatting is that it does not follow the currently preferred "gather" memory access pattern and requires blending, i. e., atomic operations and causes potential bank conflicts [KD10] on some GPUs. Also, the contribution has to be a simple summation, which is not always the case e. g., for occlusion [SA07, SGNS07, McG10].

5.6. Upsampling

Computing costly lighting solutions coarsely in space and then reconstructing it at a higher resolution is called upsampling. For interactive GI, such an approach is useful, as it allows to save computation time while preserving quality and commonly used. Upsampling can be roughly classified into edge-aware smoothing, multi-resolution, bilateral upsampling and interleaved sampling. In practice, only the indirect light

is upsampled, and, whenever possible, post-multiplication of diffuse color (e. g., textures) [LLK07] or geometric terms (e. g., bump maps) [SW09] is used.

Edge-aware smoothing In early work, McCool [McC99] reduced the noise in images generated using Monte Carlo ray tracing using blurring based on anisotropic diffusion guided by normals and positions. Later, Kontkanen et al. [KRK04] demonstrated how filtering irradiance (i. e., before multiplying with the reflectance) outperforms smoothing of radiance. Using GPUs, both upsampling and interleaved sampling can be combined with edge-aware filtering [SIMP06b]. In practice, the separable approximation to bilateral filtering is used [PvV05]. An interesting alternative is based on wavelet decomposition and edge-aware recombination, the so called "Á trous" algorithm [DSHL10]. Bauszat et al. [BEM11] address upsampling in the context of interactive ray tracing, which allows to send new rays if required.

Multi-resolution The classic approach [WRC88] is to compute indirect illumination only for some pixels. The original approach simply proceeds in scan line order and re-computes GI only for each pixel that deviates too much from the last GI result computed. Newer approaches are based on mipmaps and stencil buffers [NPW10] or clustering [WWZ*09], some include the radiance signal itself [MW11] and some not. Yang et al. [YSL08] propose a hierarchical framebuffer that could also be used to compute GI. Multi-resolution is not the best solution for low-quality / high-speed GPU approaches, but best fits medium-to-high quality. This is because the irregular and adaptive structure requires more irregular code and data-access, which does not fit to the GPU. So some speed is won by adaptivity and some speed is lost by the lower suitability for GPUs.

Bilateral upsampling The idea of joint upsampling was first proposed in the context of image processing [KCLU07]. Instead of using a smooth reconstruction filter to re-sample a low-resolution solution to a high-resolution solution, an edge aware kernel [TM99] is used to avoid blurring across edges. Note, that this process is not aiming to remove noise from the low-resolution solution, but only to upsample a piecewise-smooth signal.

Interleaved sampling [KH01] is an approach that repeats a suitable sampling pattern when estimating an integral using Monte Carlo (MC) estimation. It combines strengths of regular sampling (low noise) with the accuracy of taking a high number of samples (low bias). In practice, instead of computing a MC solution using a high number of samples for every pixel, different sampling patterns are used in every pixel, but the sampling pattern is repeated. In combination with VPLs [SIMP06b, LLK07, RGK*08] interleaved sampling means to evaluate only a subset (e. g., one order of magnitude less) of all VPLs for every pixel.

5.7. Perception

In many cases, interactive GI has to be perceptually plausible without achieving physical accuracy. To this end, a more advanced understanding of human perception can be useful. This was shown in a study [YCK*09], where imperfect visibility was used in indirect light computation. The effect of GI approximations on material appearance is analyzed by Křivánek et al. [KFB10]. Sundstedt et al. [SGA*07] and Jimenez et al. [JSG09] study the perceptual rendering of volumetric light and skin. While, as listed here and in Sec. 4.3, there is research to understand perception of diffuse bounces, the perception of more general indirect light transport is not yet explored.

6. Conclusion

We conclude this report by comparing a, hopefully representative, selection of interactive GI techniques with respect to performance, quality, flexibility and scalability. We also list open problems for interactive GI methods which are not solved satisfyingly, and obviously not easy to solve at all.

6.1. Comparison

In Table 1 we show a selection of interactive GI algorithms for our comparison. We focused on those techniques that seem promising to us, or those that are a basis for current and future work, which can be seen from recent citations. The following criteria have been used to rate each technique giving more (better) or less (worse) stars for every sub-score. All ratings are absolute, i. e., if two techniques run at 100 fps they achieve the same scores for speed, not matter which light transport phenomena they account for.

Please note that ratings regarding quality, support for dynamic scenes etc., can only be subjective. The main purpose was to compare the strengths and weaknesses of each individual algorithm and what improvement which technique made. The rating has been done by all authors of this paper individually and averaged afterwards. Also note that even the seemingly simple scoring for speed is not trivial as our comparison includes works published within one decade and not all methods are available for benchmarking on current hardware.

Speed This score represents the (estimated) absolute performance of a method on current hardware. Slow(er) methods in this overview render images in the order of seconds, the fastest ones at 100 Hz or more.

Quality This score represents a measure for the absolute quality of the method laying weight on the absence of artifacts, and to a lesser degree on a comparison to reference solution. That is, emphasis is given to perceptual plausibility, not necessarily physical correctness, and e. g., flickering

is considered to be more problematic than a lack of energy preservation.

Dynamics Important for any interactive method is how static or dynamic the environment is. This score roughly classifies the methods according to static scenes; dynamic lights, camera, and materials; rigid transformations; deformable geometry; and lastly fully dynamic scenes. The score can thus be seen as an indication how many of these aspects or parameters can be changed at runtime.

Scalability This score is high if a method can retain its speed and quality with increasing complexity of geometry, light or materials as well as increasing image resolution.

GPU This score reflects, how well the approach maps to GPUs. We believe that it is important with regard to future hardware that fine-grained parallelism can be exploited. It also considers that on such hardware gathering operations are preferred over scattering, more computation is preferred over bandwidth, and memory and code coherency is important.

Implementation Lastly, this score tries to assess how complicated or costly it is to re-implement a method. That is, this score is influenced by the amount and complexity of code (not the theory behind the method), the usage of common data structures and the reusability of libraries.

6.2. Open problems

Although there is a lot of research in GI over the last 30 years, there are still many open problems.

Scalability to large scenes Most current GI methods allow interactive to real-time framerates for small- to medium sized scenes. Typical tests scenes that can be found in many papers are the Cornell box, Sponza, Sibenik, and other similar scenes. Future work will have to focus on handling larger scenes, also accounting for out-of-core methods that are widely used to (non-GI) rendering. Screen-space methods are independent of the actual geometric complexity, but obviously they only account for the acquired, limited information and thus deliver incomplete results.

Filtering Many methods rely on filtering in image space to reduce the computation cost and basically to hide (small) artifacts or noise. The downside of this is that small details in the illumination can typically not be displayed. This is not a problem for low-frequency diffuse illumination, but becomes more important when glossy materials and high-frequency illumination is used. Applying filters to the image signal results in smeared reflections and effectively changes the BRDF. It is thus important to improve filtering techniques beyond those operating on screen space information only.

Temporal coherence is one of the main problems in dynamic scenes. Since many methods are built on approximations, a small modification in the scene, like an object movement, can result in large changes in the approximated illumination. While this is not a problem for still images, a disturbing flickering can appear in animated scenes. Many current methods use filtering in image/object space or in the time domain to hide these artifacts.

Glossy bounces Recent methods often compute only the first indirect bounce (from the camera) and restrict the rest of the computation to diffuse interreflections. The reasoning is simple: diffuse illumination results in low-frequency indirect lighting (especially with multiple bounces), while glossy reflections can results in high energy light bundles. Thus, we believe that interactive GI supported multiple bounces with arbitrary BRDFs is a challenging topic for future work. A first step in this direction is made by Laurijssen et al. [LWDB10], where Gaussian lobes are used to represent directional transport with more than one glossy bounce (indirect highlights).

Volumes When participating media are included in the computation, then the single scattering assumption is often made to simplify the computation and enable interactive framerates. Recently, the first solutions for interactive multiple scattering in volumes have been presented, opening up new perspectives and a new research topic.

Complex lights Many of the mentioned methods assume the direct illumination to stem from point or spot lights, e. g., to sample the directly lit surfaces using reflective shadow maps. The obvious goal is to provide unified support for all types of light sources in interactive GI methods, to obviate combining several different techniques. Sampling the light surface into VPLs only is a partial solution to this issue.

Parameter tweaking The possibly most important aspect is that most of the interactive GI methods typically require a lot of parameter adjustment to determine the correct settings for a given scene. Otherwise they often suffer from disturbing artifacts or suboptimal performance. At present, this tweaking is performed manually, there are usually no means to determine good parameters automatically.

This outlook concludes our attempt to summarize, classify, and weigh the works of an incredibly active research area of the past decades. Apparently, computer graphics research led to a point where stunningly plausible renderings of global illumination effects can be rendered in real-time. However, as the open problems indicate, none of these methods is the ultimate solution yet. We are optimistic that research will go on and please us with even more overwhelming results in the future.

Acknowledgements We would like to thank Matthias Holländer and Oliver Klehm for proofreading and the anonymous reviewers for their detailed and helpful comments.

References

[ADM*08] ANNEN T., DONG Z., MERTENS T., BEKAERT P., SEIDEL H.-P., KAUTZ J.: Real-time, all-frequency shadows in dynamic scenes. *ACM Trans. Graph (Proc. SIGGRAPH) 27*, 3 (2008), 34:1–34:8. 7, 11, 15

[AFO05] ARIKAN O., FORSYTH D. A., O'BRIEN J. F.: Fast and detailed approximate global illumination by irradiance decomposition. *ACM Trans. Graph (Proc. SIGGRAPH) 24*, 3 (2005), 1108. 10

[AK10] AILA T., KARRAS T.: Architecture considerations for tracing incoherent rays. In *Proc. High Performance Graphics* (2010), pp. 113–122. 5, 14

[AKDS04] ANNEN T., KAUTZ J., DURAND F., SEIDEL H.-P.: Spherical Harmonic Gradients for Mid-Range Illumination. In *Proc. EGWR* (2004), pp. 331–336. 10, 26

[AL09] AILA T., LAINE S.: Understanding the efficiency of ray traversal on GPUs. In *Proc. High Performance Graphics* (2009), p. 145. 5, 14

[ARBJ03] AGARWAL S., RAMAMOORTHI R., BELONGIE S., JENSEN H. W.: Structured importance sampling of environment maps. *ACM Trans. Graph (Proc. SIGGRAPH) 22*, 3 (2003), 605. 15

[ATS94] ARVO J., TORRANCE K., SMITS B.: A framework for the analysis of error in global illumination algorithms. In *Proc. SIGGRAPH* (1994), SIGGRAPH '94, pp. 75–84. 2

[AUW07] AKERLUND O., UNGER M., WANG R.: Precomputed Visibility Cuts for Interactive Relighting with Dynamic BRDFs. In *Proc. Pacific Graphics* (2007), pp. 161–170. 8, 10, 15, 26

[BAM06] BIRI V., ARQUÈS D., MICHELIN S.: Real Time Rendering of Atmospheric Scattering and Volumetric Shadows. *J WSCG 14(1)* (2006), 65–72. 13

[BAM08] BÄRZ J., ABERT O., MULLER S.: Interactive particle tracing in dynamic scenes consisting of NURBS surfaces. In *Proc. Interactive Ray Tracing* (2008), pp. 139–146. 6

[BAOR06] BEN-ARTZI A., OVERBECK R., RAMAMOORTHI R.: Real-time brdf editing in complex lighting. *ACM Trans. Graph. 25* (2006), 945–954. 10

[BCRK*10] BARAN I., CHEN J., RAGAN-KELLEY J., DURAND F., LEHTINEN J.: A Hierarchical Volumetric Shadow Algorithm for Single Scattering. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 29*, 6 (2010), 178:1–10. 14

[BEL*07] BOULOS S., EDWARDS D., LACEWELL J. D., KNISS J., KAUTZ J., SHIRLEY P., WALD I.: Packet-based whitted and distribution ray tracing. In *Proc. Graphics Interface* (2007), GI '07, pp. 177–184. 5

[BEM11] BAUSZAT P., EISEMANN M., MAGNOR M.: Guided Image Filtering for Interactive High-quality Global Illumination. *Computer Graphics Forum (Proc. EGSR) 30*, 4 (2011), 1361–1368. 16

[BGZ*06] BIMBER O., GRUNDHOFER A., ZEIDLER T., DANCH D., KAPAKOS P.: Compensating Indirect Scattering for Immersive and Semi-Immersive Projection Displays. In *Proc. Virtual Reality* (2006), pp. 151–158. 2

[BKY99] BELHUMEUR P. N., KRIEGMAN D. J., YUILLE A. L.: The Bas-Relief Ambiguity. *Int J Computer Vision 35*, 1 (1999), 33–44. 12

[BO04] BARCZAK J., OLANO M.: Interactive Shadowed Caustics Using Hierarchical Light Volumes, 2004. 12

[BS08] BAVOIL L., SAINZ M.: Screen Space Ambient Occlusion, 2008. 11, 15

[BS09] BAVOIL L., SAINZ M.: Multi-layer dual-resolution screen-space ambient occlusion. In *SIGGRAPH Talks* (2009). 11, 15

[BSA10] BILLETER M., SINTORN E., ASSARSSON U.: Real Time Volumetric Shadows using Polygonal Light Volumes. In *Proc. High Performance Graphics* (2010), pp. 39–45. 14

[Buc27] BUCKLEY H.: On the radiation from the inside of a circular cylinder. *Phil. Mag. 4* (1927), 753. 2

[Bun05] BUNNELL M.: *GPU Gems 2: Dynamic ambient occlusion and indirect lighting*, vol. 2. Addison-Wesley, 2005, ch. 14, pp. 223–233. 11, 15, 25

[CB04] CHRISTENSEN P., BATALI D.: An irradiance atlas for global illumination in complex production scenes. In *Proc. EGSR* (2004). 10

[CBDJ11] CHEN J., BARAN I., DURAND F., JAROSZ W.: Real-time volumetric shadows using 1D min-max mipmaps. In *Proc. I3D* (2011), I3D '11, pp. 39–46. 14

[CCWG88] COHEN M. F., CHEN S. E., WALLACE J. R., GREENBERG D. P.: A progressive refinement approach to fast radiosity image generation. *ACM SIGGRAPH Computer Graphics 22*, 4 (1988), 75–84. 4

[Cha50] CHANDRASEKHAR S.: *Radiative Transfer*. Dover Pubns, 1950. 9

[CHL04] COOMBE G., HARRIS M. J., LASTRA A.: Radiosity on graphics hardware. In *Proc. Graphics Interface* (2004), pp. 161–168. 4, 5, 25

[Chr08] CHRISTENSEN P.: *Point-Based Approximate Color Bleeding*. Tech. rep., Pixar, 2008. 6, 8, 15, 25

[CK10] COX A., KAUTZ J.: Dynamic Ambient Occlusion from Volumetric Proxies. In *SIGGRAPH Posters* (2010). 11

[CKK05] CHANDRAKER M., KAHL F., KRIEGMAN D.: Reflections on the Generalized Bas-Relief Ambiguity. In *Proc. CVPR* (2005), pp. 788–795. 12

[Coe04] COELHO P. J.: A modified version of the discrete ordinates method for radiative heat transfer modelling. *Computational Mechanics 33*, 5 (2004), 375–388. 9

[CPP*05] CEREZO E., PÉREZ F., PUEYO X., SERON F. J., SILLION F. X.: A survey on participating media rendering techniques. *The Visual Computer 21*, 5 (2005), 303–328. 13

[CPWAP08] CHESLACK-POSTAVA E., WANG R., AKERLUND O., PELLACINI F.: Fast, realistic lighting and material design using nonlinear cut approximation. *ACM Trans. Graph. 27* (2008), 128:1–128:10. 10

[CRGZ10] CAO C., REN Z., GUO B., ZHOU K.: Interactive Rendering of Non-Constant, Refractive Media Using the Ray Equations of Gradient-Index Optics. *Computer Graphics Forum (Proc. EGSR) 29*, 4 (2010), 1375–1382. 14

[DBBS06] DUTRE P., BALA K., BEKAERT P., SHIRLEY P.: *Advanced Global Illumination*. AK Peters Ltd, 2006. 2

[DBMS02] DMITRIEV K., BRABEC S., MYSZKOWSKI K., SEIDEL H.-P.: Interactive Global Illumination Using Selective Photon Tracing. In *Proc. EGWR* (2002), pp. 25–36. 6, 25

[Deb98] DEBEVEC P.: Rendering synthetic objects into real scenes. In *Proc. SIGGRAPH* (1998), pp. 189–198. 2, 11

[DGR*09] DONG Z., GROSCH T., RITSCHEL T., KAUTZ J., SEIDEL H.-P.: Real-time Indirect Illumination with Clustered Visibility. In *Proc. Vision Modeling and Visualization* (2009). 7, 15, 25

[DKTS07] DONG Z., KAUTZ J., THEOBALT C., SEIDEL H.-P.: Interactive Global Illumination Using Implicit Visibility. In *Proc. Pacific Graphics* (2007), vol. 2, pp. 77–86. 4, 11, 12, 15, 25

[DLAW01] DROR R., LEUNG T., ADELSON E., WILLSKY A.: Statistics of real-world illumination. In *Proc. CVPR* (2001), pp. II–164–II–171. 11

[DLE07] D'EON E., LUEBKE D., ENDERTON E.: Efficient rendering of human skin. In *Proc. EGSR* (2007). 13

[DS03] DACHSBACHER C., STAMMINGER M.: Translucent shadow maps. In *Proc. EGSR* (2003), pp. 197–201. 13

[DS05] DACHSBACHER C., STAMMINGER M.: Reflective shadow maps. In *Proc. I3D* (2005), p. 203. 6, 7, 9, 11, 12, 14, 15, 25

[DS06] DACHSBACHER C., STAMMINGER M.: Splatting indirect illumination. In *Proc. I3D* (2006), p. 93. 7, 12, 15, 25

[DSDD07] DACHSBACHER C., STAMMINGER M., DRETTAKIS G., DURAND F.: Implicit visibility and antiradiance for interactive global illumination. *ACM Trans. Graph (Proc. SIGGRAPH) 26*, 3 (2007), 61. 4, 11, 12, 15, 25

[DSHL10] DAMMERTZ H., SEWTZ D., HANIKA J., LENSCH H. P. A.: Edge-avoiding À-Trous wavelet transform for fast global illumination filtering. In *Proc. High Performance Graphics* (2010), pp. 67–75. 16

[DYN02] DOBASHI Y., YAMAMOTO T., NISHITA T.: Interactive rendering of atmospheric scattering effects using graphics hardware. In *Proc. Graphics Hardware* (2002), pp. 99–107. 14

[EAMJ05] ERNST M., AKENINE-MÖLLER T., JENSEN H. W.: Interactive rendering of caustics using interpolated warped volumes. In *Proc. Graphics Interface* (2005), p. 96. 12, 14

[ED08] EISEMANN E., DÉCORET X.: Single-pass GPU solid voxelization for real-time applications. In *Proc. Graphics Interface* (2008), pp. 73–80. 14

[ED10] ENGELHARDT T., DACHSBACHER C.: Epipolar sampling for shadows and crepuscular rays in participating media with single scattering. In *Proc. I3D* (2010), p. 119. 14

[ESAW11] EISEMANN E., SCHWARZ M., ASSARSSON U., WIMMER M.: *Real-time Shadows*. AK Peters, 2011. 2, 3, 10

[Eva98] EVANS K. F.: The Spherical Harmonics Discrete Ordinate Method for Three-Dimensional Atmospheric Radiative Transfer. *J Atmospheric Sciences 55*, 3 (1998), 429–446. 9

[Fat09] FATTAL R.: Participating media illumination using light propagation maps. *ACM Trans. Graph. 28*, 1 (2009), 7:1–7:11. 9, 25

[FD09] FABIANOWSKI B., DINGLIANA J.: Interactive Global Photon Mapping. *Computer Graphics Forum 28*, 4 (2009), 1151–1159. 6, 25

[FDH91] FUNT B. V., DREW M. S., HO J.: Color constancy from mutual reflection. *Int J Computer Vision 6*, 1 (1991), 5–24. 2

[Fer02] FERNANDO R.: *Percentage Closer Soft Shadow*. Tech. rep., Nvidia, 2002. 10

[FJB04] FLEMING R. W., JENSEN H. W., BÜLTHOFF H. H.: Perceiving translucent materials. In *Proc. APGV* (2004), p. 127. 12

[FM08] FILION D., MCNAUGHTON R.: Starcraft 2 : Effects & techniques. In *SIGGRAPH Classes* (2008), p. 133. 11

[FZ91] FORSYTH D., ZISSERMAN A.: Reflections on shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence 13*, 7 (1991), 671–679. 2

[GCHH03] GIBSON S., COOK J., HOWARD T., HUBBOLD R.: Rapid shadow generation in real-world lighting environments. In *Proc. EGSR* (2003), pp. 219–229. 2

[GEM07] GROSCH T., EBLE T., MUELLER S.: Consistent interactive augmentation of live camera images with correct near-field illumination. In *Proc. VRST* (2007), vol. 1, p. 125. 11

[GH00] GERSHBEIN R., HANRAHAN P.: A Fast Relighting Engine for Interactive Cinematic Lighting Design. In *ACM SIGGRAPH Computer Graphics* (2000), pp. 353–358. 9

[Gil79] GILCHRIST A. L.: The perception of surface blacks and whites. *Scientific American 240*, 3 (1979), 112–22, 124. 12

[GKBP05] GAUTRON P., KRIVÁNEK J., BOUATOUCH K., PATTANAIK S.: Radiance cache splatting. In *SIGGRAPH Sketches* (2005), p. 36. 15

[GKD07] GREEN P., KAUTZ J., DURAND F.: Efficient Reflectance and Visibility Approximations for Environment Map Rendering. *Computer Graphics Forum 26*, 3 (2007), 495–502. 10, 26

[GKMD06] GREEN P., KAUTZ J., MATUSIK W., DURAND F.: View-dependent precomputed light transport using nonlinear Gaussian function approximations. In *Proc. I3D* (2006), p. 7. 10, 26

[GMF09] GAUTRON P., MARVIE J.-E., FRANÇOIS G.: Volumetric Shadow Mapping. ACM SIGGRAPH 2009 Sketches, 2009. 13

[GRWS04] GEIST R., RASCHE K., WESTALL J., SCHALKOFF R. J.: Lattice-Boltzmann Lighting. In *Proc. EGSR* (2004), pp. 355–362. 9, 25

[GSCH93] GORTLER S. J., SCHRÖDER P., COHEN M. F., HANRAHAN P.: Wavelet radiosity. In *Proc. SIGGRAPH* (1993), pp. 221–230. 4

[GSHG98] GREGER G., SHIRLEY P., HUBBARD P., GREENBERG D.: The irradiance volume. *IEEE Computer Graphics and Applications 18*, 2 (1998), 32–43. 10, 26

[GTGB84] GORAL C. M., TORRANCE K. E., GREENBERG D. P., BATTAILE B.: Modeling the interaction of light between diffuse surfaces. *ACM SIGGRAPH Computer Graphics 18*, 3 (1984), 213–222. 4

[GWH01] GARLAND M., WILLMOTT A., HECKBERT P. S.: Hierarchical face clustering on polygonal surfaces. In *Proc. I3D* (2001), pp. 49–58. 4

[GWS04] GÜNTHER J., WALD I., SLUSALLEK P.: Realtime Caustics Using Distributed Photon Mapping. In *Proc. EGSR* (2004), Keller A., Jensen H. W., (Eds.), pp. 111–121. 6

[GWWD09] GUTIERREZ D., WANN JENSEN H., WOJCIECH J., DONNER C.: Scattering. In *SIGGRAPH Course* (2009). 2

[HDI*10] HU W., DONG Z., IHRKE I., GROSCH T., YUAN G., SEIDEL H.-P.: Interactive volume caustics in single-scattering media. In *Proc. I3D* (2010), p. 109. 14

[HDKS00] HEIDRICH W., DAUBERT K., KAUTZ J., SEIDEL H.-P.: Illuminating Micro Geometry Based on Precomputed Visibility. In *Proc. SIGGRAPH* (2000), pp. 455–464. 10

[HDMS03] HAVRAN V., DAMEZ C., MYSZKOWSKI K., SEIDEL H.-P.: An efficient spatio-temporal architecture for animation rendering. In *Proc. EGSR* (2003), pp. 106–117. 5

[Hec90] HECKBERT P. S.: Adaptive radiosity textures for bidirectional ray tracing. *Proc. SIGGRAPH 24*, 4 (1990), 145–154. 4

[Hig34] HIGBIE H. H.: *Lighting Calculations*. Wiley & Sons, 1934. 2

[HJ07] HOBEROCK J., JIA Y.: *High-Quality Ambient Occlusion*. Addison Wesley, 2007, ch. 12, pp. 257–274. 11

[HJ10] HACHISUKA T., JENSEN H. W.: Parallel progressive photon mapping on GPUs. In *SIGGRAPH Sketches* (2010), p. 54. 6, 13, 25

[HKWB09] HAŠAN M., KRIVÁNEK J., WALTER B., BALA K.: Virtual spherical lights for many-light rendering of glossy scenes. *ACM Trans. Graph (Proc. SIGGRAPH) 28*, 5 (2009), 143:1–143:6. 7, 25

[HLHS03] HASENFRATZ J. M., LAPIERRE M., HOLZSCHUCH N., SILLION F.: A Survey of Real-time Soft Shadows Algorithms. *Computer Graphics Forum 22*, 4 (2003), 753–774. 2, 3

[HPB06] HAŠAN M., PELLACINI F., BALA K.: Direct-to-indirect transfer for cinematic relighting. *ACM Trans. Graph (Proc. SIGGRAPH) 25*, 3 (2006), 1089. 9

[HPB07] HAŠAN M., PELLACINI F., BALA K.: Matrix row-column sampling for the many-light problem. *ACM Trans. Graph (Proc. SIGGRAPH) 26*, 3 (2007), 26. 8, 14

[HQ07] HU W., QIN K.: Interactive approximate rendering of reflections, refractions, and caustics. *IEEE Transactions on Visualization and Computer Graphics 13*, 1 (2007), 46–57. 12

[HREB11] HOLLÄNDER M., RITSCHEL T., EISEMANN E., BOUBEKEUR T.: ManyLoDs: Parallel Many-View Level-of-Detail Selection for Real-Time Global Illumination. *Computer Graphics Forum (Proc. EGSR) 30*, 4 (2011), 1233–1240. 7, 8, 15, 25

[HS99] HEIDRICH W., SEIDEL H.-P.: Realistic, hardware-accelerated shading and lighting. In *Proc. SIGGRAPH* (1999), SIGGRAPH '99, pp. 171–178. 10

[HSA91] HANRAHAN P., SALZMAN D., AUPPERLE L.: A rapid hierarchical radiosity algorithm. *ACM SIGGRAPH Computer Graphics 25*, 4 (1991), 197–206. 4, 15

[HSD94] HOLZSCHUCH N., SILLION F., DRETTAKIS G.: An Efficient Progressive Refinement Strategy for Hierarchical Radiosity. In *Proc. EGWR* (1994). 4

[HSK*05] HAVRAN V., SMYK M., KRAWCZYK G., MYSZKOWSKI K., SEIDEL H.-P.: Importance sampling for video environment maps. In *SIGGRAPH Sketches* (2005), p. 109. 11, 15

[HVAPB08] HAŠAN M., VELÁZQUEZ-ARMENDÁRIZ E., PELLACINI F., BALA K.: Tensor Clustering for Rendering Many-Light Animations. *Comptuter Graphics Forum (Proc. EGSR) 27*, 4 (2008), 1105–1114. 8, 25

[ICG86] IMMEL D. S., COHEN M. F., GREENBERG D. P.: A radiosity method for non-diffuse environments. *ACM SIGGRAPH Computer Graphics 20*, 4 (1986), 133–142. 4

[IDN03] IWASAKI K., DOBASHI Y., NISHITA T.: A Fast Rendering Method for Refractive and Reflective Caustics Due to Water Surfaces. *Computer Graphics Forum (Proc. EGSR) 22*, 3 (2003), 601–609. 12

[IJTN07] IMAGIRE T., JOHAN H., TAMURA N., NISHITA T.: Anti-aliased and real-time rendering of scenes with light scattering effects. *The Visual Computer 23*, 9 (2007), 935–944. 14

[IZT*07] IHRKE I., ZIEGLER G., TEVS A., THEOBALT C., MAGNOR M., SEIDEL H.-P.: Eikonal rendering. *ACM Trans. Graph (Proc. SIGGRAPH) 26*, 3 (2007), 59. 14, 25

[Jam03] JAMES R.: True volumetric shadows. In *Graphics programming methods* (2003), pp. 353–366. 13

[JB02] JENSEN H. W., BUHLER J.: A rapid hierarchical rendering technique for translucent materials. *ACM Trans. Graph (Proc. SIGGRAPH) 21*, 3 (2002), 576–581. 13

[Jen96] JENSEN H. W.: Global illumination using photon maps. In *Proc. EGWR* (1996), pp. 21–30. 5, 15

[JMLH01] JENSEN H. W., MARSCHNER S. R., LEVOY M., HANRAHAN P.: A practical model for subsurface light transport. In *Proc. SIGGRAPH* (2001), pp. 511–518. 13

[JSG09] JIMENEZ J., SUNDSTEDT V., GUTIERREZ D.: Screen-space perceptual rendering of human skin. *ACM Trans. Applied Perception 6*, 4 (2009), 23:1–23:15. 13, 15, 16

[JSN94] JEFFRY S. NIMEROFF EERO SIMONCELLI J. D.: Efficient re-rendering of naturally illuminated environments. In *Proc. EGWR* (1994), pp. 359–373. 9

[KA06a] KIRK A. G., ARIKAN O.: Precomputed ambient occlusion for character skins. In *SIGGRAPH Sketches* (2006), p. 104. 10, 11

[KA06b] KONTKANEN J., AILA T.: Ambient Occlusion for Animated Characters. In *Proc. I3D* (2006), pp. 343–348. 10, 11, 26

[Kaj86] KAJIYA J. T.: The rendering equation. In *Proc. SIGGRAPH* (1986), vol. 20, pp. 143–150. 2, 5

[KBW06] KRÜGER J., BÜRGER K., WESTERMANN R.: Interactive Screen-Space Accurate Photon Tracing on GPUs. In *Proc. EGSR* (2006), pp. 319–329. 6, 12, 14, 25

[KCLU07] KOPF J., COHEN M. F., LISCHINSKI D., UYTTENDAELE M.: Joint bilateral upsampling. *ACM Trans. Graph (Proc. SIGGRAPH) 26*, 3 (2007), 96. 16

[KD10] KAPLANYAN A., DACHSBACHER C.: Cascaded light propagation volumes for real-time indirect illumination. In *Proc. I3D* (2010), p. 99. 9, 10, 15, 26

[Kel97] KELLER A.: Instant radiosity. *Proc. SIGGRAPH 31*, 3 (1997), 49–56. 6, 25

[KFB10] KRIVÁNEK J., FERWERDA J. A., BALA K.: Effects of global illumination approximations on material appearance. *ACM Trans. Graph (Proc. SIGGRAPH) 29*, 4 (2010), 112:1–112:10. 12, 16

[KFC*10] KRIVÁNEK J., FAJARDO M., CHRISTENSEN P. H., TABELLION E., BUNNELL M., LARSSON D., KAPLANYAN A.: Global Illumination Across Industries. In *SIGGRAPH Course* (2010). 2

[KH96] KERSTEN D., HURLBERT A.: Discounting the color of mutual illumination: illumination: A 3D shapeinduced color phenomenon. *Investigative Ophthalmology and Visual Science 32*, 3 (1996). 12

[KH01] KELLER A., HEIDRICH W.: Interleaved Sampling. In *Proc. EGWR* (2001), pp. 269–276. 11, 15, 16

[KL05] KONTKANEN J., LAINE S.: Ambient occlusion fields. In *Proc. I3D* (2005), p. 41. 10, 11, 26

[KLA04] KAUTZ J., LEHTINEN J., AILA T.: Hemispherical Rasterization for Self-Shadowing of Dynamic Objects. In *Proc. EGSR* (2004), vol. pages, pp. 179–184. 10, 14

[KM99] KAUTZ J., MCCOOL M. D.: Interactive Rendering with Arbitrary BRDFs using Separable Approximations. In *Proc. EGWR* (1999), pp. 281–292. 10

[KRK04] KONTKANEN J., RÄSÄNEN J., KELLER A.: Irradiance Filtering for Monte Carlo Ray Tracing. In *Proc. MC2QMC* (2004), pp. 259–272. 16

[KSS02] KAUTZ J., SLOAN P.-P., SNYDER J.: Fast, arbitrary BRDF shading for low-frequency lighting using spherical harmonics. In *Proc. EGWR* (2002), pp. 291–296. 9, 26

[KTM*10] KNECHT M., TRAXLER C., MATTAUSCH O., PURGATHOFER W., WIMMER M.: Differential Instant Radiosity for Mixed Reality. In *Proc. ISMAR* (2010), pp. 99–107. 11

[KvD83] KOENDERINK J. J., VAN DOORN A. J.: Geometrical modes as a general method to treat diffuse interreflections in radiometry. *J Optical Society of America 73*, 6 (1983), 843. 4

[Lan99] LANGER M. S.: When Shadows Become Interreflections. *Int J Computer Vision 34*, 2-3 (1999), 193–204. 12

[Lan11] LANGER M. S.: Interreflections, 2011. 2

[LBC94] LANGUENOU E., BOUATOUCH K., CHELLE M.: Global Illumination in Presence of Participating Media with General Properties. In *Proc. EGWR* (1994), pp. 69–85. 9

[LC04] LARSEN B. D., CHRISTENSEN N. J.: Simulating photon mapping for real-time applications. In *Proc. EGSR* (2004), Keller A., Jensen H. W., (Eds.), pp. 123–132. 6

[LD10] LIKTOR G., DACHSBACHER C.: Real-Time Volumetric Caustics with Projected Light Beams. In *Proc. Graphics and Geometry* (2010), pp. 151–158. 14

[LD11] LIKTOR G., DACHSBACHER C.: Real-Time Volume Caustics with Adaptive Beam Tracing. In *Proc. I3D* (2011), pp. 47–54. 14

[LGB*03] LENSCH H. P. A., GOESELE M., BEKAERT P., KAUTZ J., MAGNOR M. A., LANG J., SEIDEL H. P.: Interactive rendering of translucent objects. *Computer Graphics Forum 22*, 2 (2003), 195–205. 13

[LK03] LEHTINEN J., KAUTZ J.: Matrix radiance transfer. In *Proc. I3D* (2003), I3D '03, pp. 59–64. 10

[LLK07] LAINE S., LEHTINEN J., KONTKANEN J.: Incremental Instant Radiosity for Real-Time Indirect Illumination. In *Proc. EGSR* (2007), pp. 4–8. 7, 11, 12, 16, 25

[LP03] LAVIGNOTTE F., PAULIN M.: Scalable photon splatting for global illumination. In *Proc. GRAPHITE* (2003), p. 203. 15

[LS10] LOOS B. J., SLOAN P.-P.: Volumetric obscurance. In *Proc. I3D* (2010), p. 151. 11, 15

[LSSS04] LIU X., SLOAN P.-P., SHUM H.-Y., SNYDER J.: All-Frequency Precomputed Radiance Transfer for Glossy Objects. In *Proc. EGSR* (2004), vol. 3, pp. 337–344. 10, 26

[LTG93] LISCHINSKI D., TAMPIERI F., GREENBERG D. P.: Combining hierarchical radiosity and discontinuity meshing. In *Proc. SIGGRAPH* (1993), pp. 199–208. 4

[LW93] LAFORTUNE E. P., WILLEMS Y. D.: Bi-Directional Path Tracing. In *Proc. SIGGRAPH* (1993), pp. 145–153. 5

[LWDB10] LAURIJSSEN J., WANG R., DUTRE P., BROWN B.: Fast estimation and rendering of indirect highlights. *Computer Graphics Forum 29*, 4 (2010), 1305–1313. 13, 17

[LZ94] LANGER M. S., ZUCKER S. W.: Shape-from-shading on a cloudy day. *J Optical Society of America A 11*, 2 (1994), 467. 11

[LZKF08] LEHTINEN J., ZWICKER M., KONTKANEN J., FRANC E. T.: Meshless Finite Elements for Hierarchical Global Illumination. *ACM Trans. Graph. (Proc. SIGGRAPH) 27* (2008), 37:1–37:11. 10, 15, 26

[Max86] MAX N. L.: Atmospheric illumination and shadows. In *Proc. SIGGRAPH* (1986), pp. 117–124. 13, 14

[McC99] MCCOOL M. D.: Anisotropic diffusion for Monte Carlo noise reduction. *ACM Trans. Graph. 18*, 2 (1999), 171–194. 16

[McG10] MCGUIRE M.: Ambient occlusion volumes. In *Proc. High Performance Graphics* (2010), HPG '10, pp. 47–56. 11, 15

[MD11] MÜCKL G., DACHSBACHER C.: Deducing Explicit from Implicit Visibility for Global Illumination with Antiradiance. *J WSCG 19* (2011), 59–68. 4

[Mec01] MECH R.: Hardware-accelerated real-time rendering of gaseous phenomena. *J Graphics Tools 6* (2001), 1–16. 13

[MESD09] MEYER Q., EISENACHER C., STAMMINGER M., DACHSBACHER C.: Data-Parallel Hierarchical Link Creation for Radiosity. In *Proc. Parallel Graphics and Visualization* (2009), pp. 65–70. 5, 15, 25

[MFS08] MÉNDEZ-FELIU A., SBERT M.: From obscurances to ambient occlusion: A survey. *The Visual Computer 25*, 2 (2008), 181–196. 2

[Mil94] MILLER G.: Efficient algorithms for local and global accessibility shading. In *Proc. SIGGRAPH* (1994), pp. 319–326. 11

[Mit04] MITCHELL J.: Light Shafts: Rendering Shadows in Participating Media. Game Developers Conference 2004 Presentations, 2004. 14

[Mit07a] MITCHELL K.: *Volumetric Light Scattering as a Post-Process*. Addison-Wesley, 2007, pp. 275–285. 14

[Mit07b] MITTRING M.: Finding next gen: CryEngine 2. In *SIGGRAPH Course* (2007), pp. 97–121. 11

[MKB*03a] MERTENS T., KAUTZ J., BEKAERT P., SEIDEL H.-P., VAN REETH F.: Interactive rendering of translucent deformable objects. In *SIGGRAPH Sketches* (2003). 13

[MKB*03b] MERTENS T., KAUTZ J., BEKAERT P., VAN REETH F., SEIDEL H.-P.: Efficient rendering of local subsurface scattering. In *Proc. Pacific Graphics* (2003), pp. 51–58. 13, 15

[ML09] MCGUIRE M., LUEBKE D.: Hardware-accelerated global illumination by image space photon mapping. In *Proc. High Performance Graphics* (2009), p. 77. 6, 14, 15, 25

[MM02] MA V. C. H., MCCOOL M. D.: Low Latency Photon Mapping Using Block Hashing. In *Proc. Graphics Hardware* (2002), pp. 89–99. 6, 25

[MMAH07] MALMER M., MALMER F., ASSARSSON U., HOLZSCHUCH N.: Fast precomputed ambient occlusion for proximty shadows. *J Graphics Tools 12*, 2 (2007), 51–75. 11

[Moo40] MOON P.: On Interreflections. *J Optical Society of America 30*, 5 (1940), 195. 2

[MW11] MALETZ D., WANG R.: Importance Point Projection for GPU-based Final Gathering. *Computer Graphics Forum (Proc. EGSR) 30*, 4 (2011), 1327–1336. 8, 16, 25

[NED11] NOVÁK J., ENGELHARDT T., DACHSBACHER C.: Screen-Space Bias Compensation for Interactive High-Quality Global Illumination with Virtual Point Lights. In *Proc. I3D* (2011), pp. 119–124. 7, 25

[NHD10] NOVÁK J., HAVRAN V., DACHSBACHER C.: Path Regeneration for Interactive Path Tracing. In *Eurographics Short papers* (2010), pp. 61–64. 5, 25

[NIK90] NAYAR S., IKEUCHI K., KANADE T.: Shape from interreflections. In *Proc. ICCV* (1990), pp. 2–11. 2

[NPG05] NIJASURE M., PATTANAIK S., GOEL V.: Real-time global illumination on gpus. *Journal of Graphics, GPU, & Game Tools 10*, 2 (2005), 55–71. 10

[NPW10] NICHOLS G., PENMATSA R., WYMAN C.: Interactive , Multiresolution Image-Space Rendering for Dynamic Area Lighting. *Computer Graphics Forum (Proc. EGSR) 29*, 4 (2010), 1279–1288. 15, 16

[NRH03] NG R., RAMAMOORTHI R., HANRAHAN P.: All-frequency shadows using non-linear wavelet lighting approximation. *ACM Trans. Graph. (Proc. SIGGRAPH) 22*, 3 (2003), 376–381. 10, 26

[NRH04] NG R., RAMAMOORTHI R., HANRAHAN P.: Triple product wavelet integrals for all-frequency relighting. *ACM Trans. Graph. 23*, 3 (2004), 477. 10, 26

[NS09] NOWROUZEZAHRAI D., SNYDER J.: Fast Global Illumination on Dynamic Height Fields. *Computer Graphics Forum 28*, 4 (2009), 1131–1139. 5

[NSS10] NIESSNER M., SCHÄFER H., STAMMINGER M.: Fast indirect illumination using layered depth images. *The Visual Computer (Proc. CGI) 26*, 6-8 (2010), 679–686. 14, 25

[NSW09] NICHOLS G., SHOPF J., WYMAN C.: Hierarchical Image-Space Radiosity for Interactive Global Illumination. *Computer Graphics Forum 28*, 4 (2009), 1141–1149. 5, 11, 15

[NW09] NICHOLS G., WYMAN C.: Multiresolution splatting for indirect illumination. In *Proc. I3D* (2009), vol. 1, p. 83. 15, 25

[Pan07] PAN M.: Precomputed Radiance Transfer Field for Rendering Interreflections in Dynamic Scenes. *Computer Graphics Forum 26*, 3 (2007), 485–493. 10

[PBD*10] PARKER S. G., BIGLER J., DIETRICH A., FRIEDRICH H., HOBEROCK J., LUEBKE D., MCALLISTER D., MCGUIRE M., MORLEY K., ROBISON A., STICH M.: Optix: a general purpose ray tracing engine. *ACM Trans. Graph. 29* (2010), 66:1–66:13. 5

[PBMH02] PURCELL T. J., BUCK I., MARK W. R., HANRAHAN P.: Ray tracing on programmable graphics hardware. In *Proc. SIGGRAPH* (2002), vol. 21, p. 703. 5

[PBPP11] PAJOT A., BARTHE L., PAULIN M., POULIN P.: Combinatorial Bidirectional Path-Tracing for Efficient Hybrid CPU / GPU Rendering. *Computer Graphics Forum (Proc. EG) 30*, 2 (2011), 315–324. 5

[PDC*03] PURCELL T. J., DONNER C., CAMMARANO M., JENSEN H. W., HANRAHAN P.: Photon mapping on programmable graphics hardware. In *Proc. Graphics Hardware* (2003), pp. 41–50. 6, 25

[PG04] PHARR M., GREEN S.: *GPU Gems*. Addison-Wesley, 2004, ch. Ambient Occlusion, pp. 279–292. 11

[PH04] PHARR M., HUMPHREYS G.: *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2004. 2

[Pho75] PHONG B. T.: Illumination for computer generated pictures. *Comm. ACM 18*, 6 (1975), 311–317. 10

[PMP10] PAPAIOANNOU G., MENEXI M. L., PAPADOPOULOS C.: Real-time volume-based ambient occlusion. *IEEE Transactions on Visualization and Computer Graphics 16*, 5 (2010), 752–762. 11, 15

[PPS97] PEREZ F., PUEYO X., SILLION F. X.: Global Illumination Techniques for the Simulation of Participating Media. In *Proc. EGWR* (1997), vol. 97, pp. 309–320. 9

[PSP09] PEGORARO V., SCHOTT M., PARKER S. G.: An analytical approach to single scattering for anisotropic media and light distributions. In *Proc. Graphics Interface* (2009), pp. 71–77. 14

[PSP10] PEGORARO V., SCHOTT M., PARKER S. G.: A Closed-Form Solution to Single Scattering for General Phase Functions and Light Distributions. *Computer Graphics Forum (Proc. EGSR) 29*, 4 (2010), 1365–1374. 14

[PSS11] PEGORARO V., SCHOTT M., SLUSALLEK P.: A Mathematical Framework for Efficient Closed-Form Single Scattering. In *Proc. Graphics Interface* (2011), pp. 151–158. 14

[PVL*05] PELLACINI F., VIDIMČE K., LEFOHN A., MOHR A., LEONE M., WARREN J.: Lpics: a Hybrid Hardware-Accelerated Relighting Engine for Computer Cinematography. *ACM Trans. Graph (Proc. SIGGRAPH) 24*, 3 (2005), 464–470. 9

[PvV05] PHAM T., VAN VLIET L.: Separable Bilateral Filtering for Fast Video Preprocessing. In *Proc. Multimedia and Expo* (2005), pp. 454–457. 16

[RBA09] REINBOTHE C. K., BOUBEKEUR T., ALEXA M.: Hybrid Ambient Occlusion. In *Eurographics Areas Papers* (2009). 5, 14

[REG*09] RITSCHEL T., ENGELHARDT T., GROSCH T., SEIDEL H. P., KAUTZ J., DACHSBACHER C.: Micro-rendering for scalable, parallel final gathering. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 28*, 5 (2009), 132:1–132:8. 6, 8, 14, 15, 25

[REH*11] RITSCHEL T., EISEMAN E., HA I., KIM J., SEIDEL H.-P.: Making Imperfect Shadow Maps View-Adaptive: High-Quality Global Illumination in Large Dynamic Scenes. *Computer Graphics Forum 30*, 3 (2011). 7, 10, 14, 15, 25

[RGK*08] RITSCHEL T., GROSCH T., KIM M. H., SEIDEL H. P., DACHSBACHER C., KAUTZ J.: Imperfect shadow maps for efficient computation of indirect illumination. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 27*, 5 (2008), 129:1–129:8. 7, 11, 12, 14, 15, 16, 25

[RGKM07] RITSCHEL T., GROSCH T., KAUTZ J., MÜLLER S.: Interactive Illumination with Coherent Shadow Maps. In *Proc. EGSR* (2007), pp. 61–72. 10, 26

[RGKS08] RITSCHEL T., GROSCH T., KAUTZ J., SEIDEL H.-P.: Interactive Global Illumination based on Coherent Surface Shadow Maps. In *Proc. Graphics Interface* (2008), pp. 185–192. 10, 26

[RGS09] RITSCHEL T., GROSCH T., SEIDEL H.-P.: Approximating dynamic global illumination in image space. In *Proc. I3D* (2009), p. 75. 5, 11, 15, 25

[RGW*06] REN Z., GUO B., WANG R., SNYDER J., ZHOU K., LIU X., SUN B., SLOAN P.-P., BAO H., PENG Q.: Real-time soft shadows in dynamic scenes using spherical harmonic exponentiation. *ACM Trans. Graph (Proc. SIGGRAPH) 25*, 3 (2006), 977. 11

[RKKS*07] RAGAN-KELLEY J., KILPATRICK C., SMITH B. W., EPPS D., GREEN P., HERY C., DURAND F.: The lightspeed automatic interactive lighting preview system. *ACM Trans. Graph (Proc. SIGGRAPH) 26*, 3 (2007), 25. 9

[RL00] RUSINKIEWICZ S., LEVOY M.: QSplat: A multiresolution point rendering system for large meshes. In *Proc. SIGGRAPH* (2000), pp. 343–352. 8

[RSH05] RESHETOV A., SOUPIKOV A., HURLEY J.: Multi-level ray tracing algorithm. *ACM Trans. Graph. 24*, 3 (2005), 1176. 5

[RSKU*10] RUIZ M., SZIRMAY-KALOS L., UMENHOFFER T., BOADA I., FEIXAS M., SBERT M.: Volumetric ambient occlusion for volumetric models. *The Visual Computer 26*, 6-8 (2010), 687–695. 11

[SA07] SHANMUGAM P., ARIKAN O.: Hardware accelerated ambient occlusion techniques on GPUs. In *Proc. I3D* (2007), I3D '07, p. 73. 11, 15

[SAG94] SMITS B., ARVO J., GREENBERG D.: A Clustering Algorithm for Radiosity in Complex Environments. *Computers & Graphics 28*, Annual Conference Series (1994), 435–442. 4

[SB97] STÜRZLINGER W., BASTOS R.: Interactive Rendering of Globally Illuminated Glossy Scenes. In *Proc. EGWR* (1997), pp. 93–102. 15

[SFES07] SCHJOETH L., FRISVAD J. R., ERLEBEN K., SPORRING J.: Photon Differentials. In *Proc. GRAPHITE* (2007), pp. 179–186. 6

[SGA*07] SUNDSTEDT V., GUTIERREZ D., ANSON O., BANTERLE F., CHALMERS A.: Perceptual rendering of participating media. *ACM Trans. Applied Perception 4*, 3 (2007), 15. 16

[SGNS07] SLOAN P.-P., GOVINDARAJU N. K., NOWROUZEZAHRAI D., SNYDER J.: Image-Based Proxy

Accumulation for Real-Time Soft Global Illumination. In *Proc. Pacific Graphics* (2007), pp. 97–105. 5, 10, 11, 15

[SHHS03] SLOAN P.-P., HALL J., HART J., SNYDER J.: Clustered principal components for precomputed radiance transfer. *Proc. SIGGRAPH 22*, 3 (2003), 382. 10, 11, 12

[SHR10] SOLER C., HOEL O., ROCHET F.: A deferred shading pipeline for real-time indirect illumination. In *SIGGRAPH Talks* (2010). 5, 11, 25

[SIMP06a] SEGOVIA B., IEHL J. C., MITANCHEY R., PÉROCHE B.: Bidirectional Instant Radiosity. In *Proc. EGSR* (2006), pp. 389–398. 7, 15, 25

[SIMP06b] SEGOVIA B., IEHL J. C., MITANCHEY R., PÉROCHE B.: Non-interleaved Deferred Shading of Interleaved Sample Patterns. In *Proc. Graphics Hardware* (2006), vol. 2, p. 60. 16

[SIP07] SEGOVIA B., IEHL J. C., PÉROCHE B.: Metropolis Instant Radiosity. *Computer Graphics Forum 26*, 3 (2007), 425–434. 7, 15

[SKP07] SHAH M. A., KONTTINEN J., PATTANAIK S.: Caustics mapping: an image-space technique for real-time caustics. *IEEE Transactions on Visualization and Computer Graphics 13*, 2 (2007), 272–280. 12

[SKS02] SLOAN P., KAUTZ J., SNYDER J.: Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments. *ACM Trans. Graph (Proc. SIGGRAPH) 21*, 3 (2002), 527–536. 9, 13, 26

[SKUP*09] SZIRMAY-KALOS L., UMENHOFFER T., PATOW G., SZÉCSI L., SBERT M.: Specular effects on the GPU: State of the art. *Computer Graphics Forum 28*, 6 (2009), 1586–1617. 2

[SKUT*10] SZIRMAY-KALOS L., UMENHOFFER T., TÓTH B., SZÉCSI L., SBERT M.: Volumetric Ambient Occlusion for Real-Time Rendering and Games. *IEEE Computer Graphics and Applications 30*, 1 (2010), 70–79. 11, 15

[Slo06] SLOAN P.-P.: Normal mapping for precomputed radiance transfer. In *Proc. I3D* (2006), p. 23. 10, 26

[SLS05] SLOAN P.-P., LUNA B., SNYDER J.: Local, deformable precomputed radiance transfer. *ACM Trans. Graph (Proc. SIGGRAPH) 24*, 3 (2005), 1216. 10

[Sou08] SOUSA T.: Crysis Next Gen Effects. Game Developers Conference 2008 Presentations, 2008. 14

[SR09] SUN B., RAMAMOORTHI R.: Affine double and triple product wavelet integrals for rendering. *ACM Trans. Graph. 28*, 2 (2009), 14:1–14:17. 10, 26

[SRNN05] SUN B., RAMAMOORTHI R., NARASIMHAN S. G., NAYAR S. K.: A practical analytic single scattering model for real time rendering. *ACM Trans. Graph. 24*, 3 (2005), 1040–1049. 14

[ST90] SAITO T., TAKAHASHI T.: Comprehensible rendering of 3-D shapes. *ACM SIGGRAPH Computer Graphics 24*, 4 (1990), 197–206. 5, 14

[SW09] SEGOVIA B., WALD I.: *Screen Space Spherical Harmonics Filters for Instant Global Illumination*. Tech. rep., Intel Research, 2009. 16

[SZC*07] SUN X., ZHOU K., CHEN Y., LIN S., SHI J., GUO B.: Interactive relighting with dynamic BRDFs. *ACM Trans. Graph (Proc. SIGGRAPH) 26*, 3 (2007), 27. 10, 26

[SZS*08] SUN X., ZHOU K., STOLLNITZ E., SHI J., GUO B.: Interactive relighting of dynamic refractive objects. *ACM Trans. Graph. (Proc. SIGGRAPH) 27* (2008), 35:1–35:9. 10, 14, 25

[THGM11] THIEDEMANN S., HENRICH N., GROSCH T., MÜLLER S.: Voxel-based Global Illumination. In *Proc. I3D* (2011), pp. 103–110. 5, 14, 25

[TM99] TOMASI C., MANDUCHI R.: Bilateral filtering for gray and color images. In *Proc. ICCV* (1999), pp. 839–846. 16

[TPWG02] TOLE P., PELLACINI F., WALTER B., GREENBERG D. P.: Interactive global illumination in dynamic scenes. *Proc. SIGGRAPH 21*, 3 (2002), 537. 10

[TS06] TSAI Y.-T., SHIH Z.-C.: All-Frequency Precomputed Radiance Transfer using Spherical Radial Basis Functions and Clustered Tensor Approximation. *ACM Trans. Graph. (Proc. SIGGRAPH) 1*, 3 (2006), 967–976. 10, 26

[TSS*98] THOMPSON W. B., SHIRLEY P., SMITS B., KERSTEN D. J., MADISON C.: *Visual Glue*. Tech. rep., U. Utha, 1998. 12

[TU09] TÓTH B., UMENHOFFER T.: Real-time Volumetric Lighting in Participating Media. In *Eurographics Short Papers* (2009). 14

[UPSK08] UMENHOFFER T., PATOW G., SZIRMAY-KALOS L.: Caustic triangles on the GPU. In *Proc. Computer Graphics International* (2008), pp. 1–7. 12

[vA11] VAN ANTWERPEN D.: Improving SIMD Efficiency for Parallel Monte Carlo Light Transport on the GPU. In *Proc. High Performance Graphics* (2011). 5, 25

[Vea97] VEACH E.: *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, 1997. 5

[WA09] WANG R., AKERLUND O.: Bidirectional Importance Sampling for Unstructured Illumination . *Computer Graphics Forum (Proc. Eurographics) 28*, 2 (2009), 269–278. 15

[Wan03] WAND M.: Real-Time Caustics. *Computer Graphics Forum 22*, 3 (2003), 611–620. 12

[WD06] WYMAN C., DAVIS S.: Interactive image-space techniques for approximating caustics. In *Proc. I3D* (2006), p. 153. 12

[WD08] WYMAN C., DACHSBACHER C.: Improving image-space caustics via variable-sized splatting. *J Graphics Tools 13*, 1 (2008), 1–17. 12

[WFA*05] WALTER B., FERNANDEZ S., ARBREE A., BALA K., DONIKIAN M., GREENBERG D. P.: Lightcuts: a scalable approach to illumination. *ACM Trans. Graph. (Proc. SIGGRAPH) 24*, 3 (2005), 1098–1107. 8, 12, 15

[WKB*02] WALD I., KOLLIG T., BENTHIN C., KELLER A., SLUSALLEK P.: Interactive global illumination using fast ray tracing. In *Proc. EGWR* (2002), pp. 15–24. 5, 25

[WN09] WYMAN C., NICHOLS G.: Adaptive Caustic Maps Using Deferred Shading. *Computer Graphics Forum 28*, 2 (2009), 309–318. 12

[WR08] WYMAN C., RAMSEY S.: Interactive volumetric shadows in participating media with single-scattering. In *Proc. Interactive Ray Tracing* (2008), pp. 87–92. 13

[WRC88] WARD G. J., RUBINSTEIN F. M., CLEAR R. D.: A ray tracing solution for diffuse interreflection. *ACM SIGGRAPH Computer Graphics 22*, 4 (1988), 85–92. 10, 16

[WTL04] WANG R., TRAN J., LUEBKE D.: All-Frequency Relighting of Non-Diffuse Objects using Separable BRDF Approximation. In *Proc. EGSR* (2004). 10

[WTL05] WANG R., TRAN J., LUEBKE D.: All-frequency interactive relighting of translucent objects with single and multiple scattering. *ACM Trans. Graph. 24*, 3 (2005), 1202. 10, 13

[WTL06] WANG R., TRAN J., LUEBKE D.: All-frequency relighting of glossy objects. *ACM Trans. Graph. 25*, 2 (2006), 293–318. 10, 26

[WUM97] WADA T., UKIDA H., MATSUYAMA T.: Shape from Shading with Interreflections Under a Proximal Light Source: Distortion-Free Copying of an Unfolded Book. *Int J Computer Vision 24*, 2 (1997), 125–135. 2

[WWZ*09] WANG R., WANG R., ZHOU K., PAN M., BAO H.: An efficient GPU-based approach for interactive global illumination. *ACM Trans. Graph (Proc. SIGGRAPH) 28*, 3 (2009), 91:1–91:8. 5, 16, 25

[Wym08] WYMAN C.: Hierarchical caustic maps. In *Proc. I3D* (2008), p. 163. 12

[WZH07] WANG R., ZHU J., HUMPHREYS G.: Precomputed Radiance Transfer for Real-Time Indirect Lighting using A Spectral Mesh Basis. In *Proc. EGSR* (2007). 10

[WZHB09] WALTER B., ZHAO S., HOLZSCHUCH N., BALA K.: Single scattering in refractive media with triangle mesh boundaries. *ACM Trans. Graph (Proc. SIGGRAPH) 28*, 3 (2009), 92:1–92:8. 13

[XJF*08] XU K., JIA Y.-T., FU H., HU S., TAI C.-L.: Spherical piecewise constant basis functions for all-frequency precomputed radiance transfer. *IEEE Transactions on Visualization and Computer Graphics 14*, 2 (2008), 454–467. 10, 26

[Yam26] YAMAUTI Z.: The light flux distribution of a system of interreflecting surfaces. *JOSA 13*, 5 (1926), 561–571. 2

[YCK*09] YU I., COX A., KIM M. H., RITSCHEL T., GROSCH T., DACHSBACHER C., KAUTZ J.: Perceptual influence of approximate visibility in indirect illumination. *ACM Trans. Applied Perception 6*, 4 (2009), 24:1–24:14. 16

[YID*09] YUE Y., IWASAKI K., DOBASHI Y., DOBASHI Y., NISHITA T.: Interactive Rendering of Interior Scenes with Dynamic Environment Illumination. *Computer Graphics Forum 28*, 7 (2009), 1935–1944. 10

[YIDN07] YUE Y., IWASAKI K., DOBASHI Y., NISHITA T.: Global Illumination for Interactive Lighting Design Using Light Path Pre-Computation and Hierarchical Histogram Estimation. In *Proc. Pacific Graphics* (2007), pp. 87–96. 10

[YK09] YUKSEL C., KEYSER J.: Fast real-time caustics from height fields. *The Visual Computer 25*, 5-7 (2009), 559–564. 12

[YLY07] YU X., LI F., YU J.: Image-Space Caustics and Curvatures. In *Proc. Pacific Graphics* (2007), pp. 181–188. 12

[YSL08] YANG L., SANDER P. V., LAWRENCE J.: Geometry-Aware Framebuffer Level of Detail. *Computer Graphics Forum 27*, 4 (2008), 1183–1188. 16

[YWC*10] YAO C., WANG B., CHAN B., YONG J., PAUL J.-C.: Multi-Image Based Photon Tracing for Interactive Global Illumination of Dynamic Scenes. *Comptuter Graphics Forum (Proc. EGSR) 29*, 4 (2010), 1315–1324. 6, 14, 15, 25

[ZHWG08] ZHOU K., HOU Q., WANG R., GUO B.: Real-time KD-tree construction on graphics hardware. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 27*, 5 (2008), 126:1–126:11. 5, 6, 14, 25

[ZIK98] ZHUKOV S., IONES A., KRONIN G.: An Ambient Light Illumination Model. In *Proc. EGWR* (1998), pp. 45–56. 11

Table 1: *Comparison between different methods (Please see the text in Sec. 6.1)*

Class / Method		Speed	Quality	Dynam.	Scalab.	Implem.	GPU	Transport
Finite Elements (Surface)								
Coombe et al.	[CHL04]	★★★☆☆	★★★☆☆	☆☆☆☆☆	★☆☆☆☆	★★★★☆	★☆☆☆☆	LD^+E
Bunnell et al.	[Bun05]	★★★★☆	★☆☆☆☆	★★★☆☆	★☆☆☆☆	★★★★☆	★★★★☆	LDDE
Dong et al.	[DKTS07]	★★★☆☆	★★☆☆☆	★★★☆☆	★☆☆☆☆	★★★☆☆	★★★☆☆	LD^+E
Dachsbacher et al.	[DSDD07]	★★★☆☆	★★★☆☆	★★★☆☆	★★☆☆☆	★★☆☆☆	★★★★☆	$L\{S\|D\}^+E$
Meyer et al.	[MESD09]	★★★☆☆	★★★☆☆	★★★★☆	★★☆☆☆	★★☆☆☆	★★★★☆	$L\{S\|D\}^+E$
Finite Elements (Screen space)								
Ritschel et al.	[RGS09]	★★★★★	★☆☆☆☆	★★★★☆	★★★★☆	★★★★☆	★★★★☆	LDDE
Nichols et al.	[NW09]	★★★★☆	★★☆☆☆	★★★★☆	★★★★☆	★★★☆☆	★★★★☆	$LD\{S\|D\}E$
Soler et al.	[SHR10]	★★★☆☆	★★☆☆☆	★★★★☆	★★★★★	★★★☆☆	★★★★☆	$LD\{S\|D\}E$
Finite Elements (Voxel)								
Thiedemann et al.	[THGM11]	★★★★☆	★★★☆☆	★★★☆☆	★★☆☆☆	★★★☆☆	★★★★☆	LDDE
Monte Carlo								
Wald et al.	[WKB*02]	★☆☆☆☆	★★★★☆	★★★☆☆	★★★★☆	★★★☆☆	★★★☆☆	$L\{S\|D\}^+E$
Novak et al.	[NHD10]	★☆☆☆☆	★★★★☆	★★★☆☆	★★★★☆	★★★★☆	★★★☆☆	$L\{S\|D\}^+E$
van Antwerpen	[vA11]	★★☆☆☆	★★★★★	★★★☆☆	★★★★☆	★★★☆☆	★★★☆☆	$L\{S\|D\}^+E$
Niessner et al.	[NSS10]	★★☆☆☆	★★☆☆☆	★☆☆☆☆	★★★☆☆	★★★★☆	★★★★☆	$LD\{S\|D\}^+E$
Photon Mapping								
Ma and McCool	[MM02]	★★☆☆☆	★★★☆☆	★☆☆☆☆	★★☆☆☆	★★☆☆☆	★★★☆☆	$L\{S\|D\}^+E$
Dmitriev et al.	[DBMS02]	★★★☆☆	★★★☆☆	★★★☆☆	★★★☆☆	★★★☆☆	★★★☆☆	$L\{S\|D\}^+DE$
Purcell et al.	[PDC*03]	★★★☆☆	★★★☆☆	★☆☆☆☆	★★☆☆☆	★★☆☆☆	★★★☆☆	$L\{S\|D\}^+E$
Krüger et al.	[KBW06]	★★★☆☆	★★☆☆☆	★★★★☆	★★★☆☆	★★★☆☆	★★★★☆	LS^+DE
Zhou et al.	[ZHWG08]	★★★☆☆	★★★☆☆	★★★☆☆	★★★☆☆	★★★☆☆	★★★☆☆	$L\{S\|D\}^+E$
Wang et al.	[WWZ*09]	★★★☆☆	★★★★☆	★★★☆☆	★★★☆☆	★☆☆☆☆	★★★★☆	$L\{S\|D\}^+E$
McGuire et al.	[ML09]	★★★★☆	★★☆☆☆	★★★☆☆	★★★★☆	★★★☆☆	★★★☆☆	$L\{S\|D\}^+E$
Fabianowski and Dingliana	[FD09]	★★☆☆☆	★★★☆☆	★☆☆☆☆	★★★☆☆	★★★☆☆	★★★★☆	$L\{S\|D\}^+E$
Yao et al.	[YWC*10]	★★★★☆	★★☆☆☆	★★★★☆	★★★☆☆	★★☆☆☆	★★★★☆	$L\{S\|D\}^+E$
Hachisuka and Jensen	[HJ10]	☆☆☆☆☆	★★★★★	☆☆☆☆☆	★★★☆☆	★★★☆☆	★★★☆☆	$L\{S\|D\}^+E$
Photon Mapping (Eikonal)								
Ihrke et al.	[IZT*07]	★★☆☆☆	★★★★☆	★☆☆☆☆	★☆☆☆☆	★☆☆☆☆	★★☆☆☆	$L\{V\|S\|D\}^+E$
Sun et al.	[SZS*08]	★★★☆☆	★★★★☆	★★★☆☆	★☆☆☆☆	★★☆☆☆	★★★★☆	$L\{V\|S\|D\}^+E$
Instant Radiosity								
Keller	[Kel97]	★★★☆☆	★★☆☆☆	★★★☆☆	★★☆☆☆	★★★☆☆	★★★☆☆	$LD^+\{S\|D\}E$
Dachsbacher and Stamminger	[DS05]	★★★★☆	★☆☆☆☆	★★★★★	★★★☆☆	★★★★☆	★★★★☆	LDDE
Dachsbacher and Stamminger	[DS06]	★★★★★	★★☆☆☆	★★★★★	★★★☆☆	★★★★☆	★★★★☆	$LD\{S\|D\}E$
Segovia et al.	[SIMP06a]	★★★☆☆	★★★☆☆	★★★☆☆	★★★☆☆	★★★☆☆	★★★☆☆	$L\{S\|D\}^+DE$
Laine et al.	[LLK07]	★★★★☆	★★★☆☆	★★☆☆☆	★★☆☆☆	★★★☆☆	★★★★☆	LD^+E
Ritschel et al.	[RGK*08]	★★★★☆	★★☆☆☆	★★★★☆	★★★☆☆	★★★☆☆	★★★★☆	$LD\{S\|D\}E$
Dong et al.	[DGR*09]	★★★★★	★☆☆☆☆	★★★☆☆	★★☆☆☆	★★☆☆☆	★★★★☆	LDDE
Novak et al.	[NED11]	★★★★☆	★★★☆☆	★★★★☆	★★★☆☆	★★☆☆☆	★★★★☆	$LD^+\{S\|D\}E$
Ritschel et al.	[REH*11]	★★★★☆	★★★☆☆	★★★★☆	★★★★★	★☆☆☆☆	★★★☆☆	$LD\{S\|D\}E$
Holländer et al.	[HREB11]	★★★★☆	★★☆☆☆	★★★★☆	★★★★☆	★★☆☆☆	★★★★☆	$LD\{S\|D\}E$
Many Lights								
Hašan et al.	[HVAPB08]	★☆☆☆☆	★★★★☆	★★☆☆☆	★★★☆☆	★★☆☆☆	★★★☆☆	$L\{S\|D\}DE$
Hašan et al.	[HKWB09]	★☆☆☆☆	★★★★☆	★★☆☆☆	★★★☆☆	★★★☆☆	★★★☆☆	$L\{S\|D\}DE$
Point-based								
Christensen	[Chr08]	★☆☆☆☆	★★★★☆	★★★☆☆	★★★★☆	★★☆☆☆	☆☆☆☆☆	$LD\{S\|D\}E$
Ritschel et al.	[REG*09]	★★★☆☆	★★★★☆	★★☆☆☆	★★★☆☆	★★☆☆☆	★★★☆☆	$LD\{S\|D\}E$
Holländer et al.	[HREB11]	★★★★☆	★★★☆☆	★★☆☆☆	★★★☆☆	★☆☆☆☆	★★★★☆	LDDE
Maletz and Wang	[MW11]	★★☆☆☆	★★☆☆☆	★★☆☆☆	★★★☆☆	★☆☆☆☆	★★★★★	$LD\{S\|D\}E$
Discrete Ordinate Methods								
Geist et al.	[GRWS04]	★★☆☆☆	★★☆☆☆	★☆☆☆☆	★☆☆☆☆	★★★★☆	★★★★★	LV^+E
Fattal	[Fat09]	☆☆☆☆☆	★★★★☆	☆☆☆☆☆	★★☆☆☆	★★☆☆☆	★★★★☆	LV^+E

(Continued on next page)

Class / Method		Speed	Quality	Dynam.	Scalab.	Implem.	GPU	Transport
Kaplanyan and Dachsbacher	[KD10]	★★★★★	★★☆☆☆	★★★★☆	★★★★★	★★★☆☆	★★★★★	$LD^+\{D\|V\|S\}E$
PRT								
Sloan et al.	[SKS02]	★★★★☆	★★☆☆☆	★★☆☆☆	★★☆☆☆	★★★★☆	★★★★☆	$L\{S\|D\}^+E$
Kautz et al.	[KSS02]	★★☆☆☆	★★☆☆☆	★☆☆☆☆	★★☆☆☆	★★★☆☆	★★★☆☆	$L\{S\|D\}^+E$
Ng et al.	[NRH03]	★★☆☆☆	★★★★★	★☆☆☆☆	★★☆☆☆	★★★☆☆	★★☆☆☆	LDE
Ng et al.	[NRH04]	★★☆☆☆	★★★☆☆	★☆☆☆☆	★★☆☆☆	★★★☆☆	★★★☆☆	$L\{S\|D\}^+E$
Annen et al.	[AKDS04]	★★☆☆☆	★★★☆☆	★☆☆☆☆	★★☆☆☆	★★☆☆☆	★★★☆☆	$L\{S\|D\}^+E$
Liu et al.	[LSSS04]	★★★☆☆	★★★☆☆	★☆☆☆☆	★★☆☆☆	★★★★☆	★★★★☆	$L\{S\|D\}^+E$
Wang et al.	[WTL06]	★★★☆☆	★★★★☆	★☆☆☆☆	★★☆☆☆	★★★★☆	★★★★☆	$L\{S\|D\}^+E$
Sloan	[Slo06]	★★★☆☆	★★★☆☆	★★☆☆☆	★★☆☆☆	★★★☆☆	★★★☆☆	$L\{S\|D\}^+E$
Tsai and Shih	[TS06]	★★★☆☆	★★★★☆	★☆☆☆☆	★★★☆☆	★★☆☆☆	★★☆☆☆	$L\{S\|D\}E$
Green et al.	[GKMD06]	★★☆☆☆	★★★★☆	★☆☆☆☆	★★☆☆☆	★★☆☆☆	★★★☆☆	$L\{S\|D\}^+E$
Green et al.	[GKD07]	★★★★★	★★★☆☆	★☆☆☆☆	★★★☆☆	★★★☆☆	★★★★☆	$L\{S\|D\}E$
Akerlund et al.	[AUW07]	★★★☆☆	★★★★☆	★☆☆☆☆	★★☆☆☆	★★☆☆☆	★★★☆☆	$L\{S\|D\}^+E$
Sun and Ramamoorthi	[SZC*07]	★★★☆☆	★★★☆☆	★★☆☆☆	★☆☆☆☆	★★☆☆☆	★☆☆☆☆	$L\{S\|D\}^+E$
Xu et al.	[XJF*08]	★★☆☆☆	★★★★☆	★★★☆☆	★☆☆☆☆	★★☆☆☆	★★★☆☆	$L\{S\|D\}E$
Lehtinen et al.	[LZKF08]	★★★☆☆	★★★★★	★☆☆☆☆	★★★☆☆	★★☆☆☆	★★★☆☆	$L\{S\|D\}^+E$
Sun et al.	[SR09]	★☆☆☆☆	★★★★☆	★☆☆☆☆	★★☆☆☆	★☆☆☆☆	★★☆☆☆	$L\{S\|D\}^+E$
PRT (Visibility)								
Ritschel et al.	[RGKM07]	★★☆☆☆	★★★☆☆	★☆☆☆☆	★☆☆☆☆	★★☆☆☆	★★★☆☆	$L\{S\|D\}E$
Ritschel et al.	[RGKS08]	★★★☆☆	★★★☆☆	★★☆☆☆	★★☆☆☆	★★☆☆☆	★★★☆☆	LD^+SE
PRT (Irradiance)								
Greger et al.	[GSHG98]	★★★☆☆	★★☆☆☆	★☆☆☆☆	★★☆☆☆	★★★★☆	★★★☆☆	LDE
PRT (AO)								
Kontkanen and Laine	[KL05]	★★★★☆	★☆☆☆☆	★★☆☆☆	★★☆☆☆	★★★☆☆	★★★★★	AO
Kontkanen and Aila	[KA06b]	★★★★☆	★☆☆☆☆	★★★☆☆	★★☆☆☆	★★☆☆☆	★★★★☆	AO