# Efficient Real-Time Shadows

## SIGGRAPH 2012 Course Notes

www.realtimeshadows.com

**Elmar Eisemann**
Télécom ParisTech / CNRS-LTCI

**Ulf Assarsson**
Chalmers University

**Michael Schwarz**

**Michal Valient**
Guerrilla Games / Sony Computer Entertainment

**Michael Wimmer**
Vienna University of Technology

# About the Authors

## Elmar Eisemann (Organizer)

*Associate professor,*
*Télécom ParisTech / CNRS-LTCI, France*

Before being associate professor at Télécom ParisTech, Elmar Eisemann headed a research group in the Cluster of Excellence (Saarland University/Max-Planck-Institut Informatik) (2008/2009). He studied at the École Normale Supérieure Paris (2001) and obtained Master (2004)/Ph.D. (2008) from Grenoble Universities. His interests include real-time rendering, shadow algorithms, global illumination, and GPU acceleration techniques. He was a local organizer of EGSR 2010 and is organizing EGSR 2012, HPG 2012. In 2011, he received the Eurographics Young Researcher Award.

## Ulf Assarsson

*Associate professor,*
*Department of Computer Science and Engineering, Chalmers University of Technology, Sweden*

Ulf Assarsson received his M.Sc. in Engineering Physics in 1997 and Ph.D. in Computer Graphics in 2003, and he is now head of a research group focusing primarily on real-time and non-real-time soft shadows as well as ray tracing, GPU-techniques and global illumination.

## Michael Schwarz

Michael Schwarz received a Diploma in 2005 and a Ph.D. in 2009 from the University of Erlangen-Nuremberg. His graphics-related research interests include real-time computer graphics, GPU techniques, global illumination, procedural modeling, scalable approaches, and perception-aware graphics.

## Michal Valient

*Lead Tech,*
*Guerrilla Games / Sony Computer Entertainment, Netherlands*

Michal Valient leads the technology team at Guerrilla. He spends his time working on the engine technology powering highly acclaimed games such as Killzone 2 and Killzone 3 as well as some yet unreleased projects. Prior, he worked as a programmer and a lead at Caligari where he developed the shader-based real-time rendering engine for Caligari trueSpace7. His interests include many aspects of light transfer, shadows and parallel processing. He believes in sharing the knowledge and gave GDC and Siggraph talks and wrote graphics papers published in ShaderX books and conference journals.

## Michael Wimmer

*Associate professor,*
*Institute of Computer Graphics and Algorithms, Vienna University of Technology, Austria*

Michael Wimmer received an M.Sc. in 1997 and a Ph.D. in 2001. His current research interests are real-time rendering, computer games, real-time visualization of urban environments, point-based rendering and procedural modeling. He has coauthored many papers in these fields, and was papers co-chair of EGSR 2008 and is papers co-chair of Pacific Graphics 2012.

# Course Schedule

- Introduction (Eisemann, 10 min)
  - Why shadows? What are shadows? What are the problems?
  - Course overview
- Basic algorithms (Assarsson, 15 min)
  - Shadow volumes
    *Z-pass, Z-fail, ZP+, CUDA shadow volumes, ...*
  - Shadow mapping
    *Bias, dual-depth shadow maps, ...*
- Hard shadows (Wimmer, 45 min)
  - Shadow-map aliasing
  - Shadow map reparameterization
    *Perspective shadow maps, ...*
  - Shadow map partitioning
    *Cascaded shadow maps, sample distribution shadow maps, adaptive shadow maps, ...*
  - Shadow reconstruction
    *Silhouette shadow maps, ...*
  - Precise hard shadows
    *Irregular z-buffer, alias-free shadow maps, sub-pixel antialiased shadow maps, ...*
  - Other topics
    *Temporal reprojection, shadow caster culling, ...*
- Filtering hard shadows (Eisemann, 20 min)
  - Introduction
    *Percentage closer filtering*
  - Efficient filtering
    *Variance shadow maps, convolution shadow maps, exponential shadow maps, ...*
  - Issues
    *PCF bias, accelerations, ...*

- Break
- Soft shadows (Schwarz & Eisemann, 40 min)
  - Introduction
  - Image-based solutions
    * Percentage-closer soft shadows and variants
      *PCSS, CSSM, VSSM, …*
    * Occlusion textures
    * Soft shadow mapping (backprojection)
  - Geometry-based solutions
    * Penumbra wedges
    * Soft shadow volumes
  - Accurate soft shadows
    *View-sample mapping, depth-complexity sampling, …*
- Volumetric shadows (Assarsson, 15 min)
  - Ray-marching
    *1D min-max mipmaps*
  - Shadow volumes
    *Polygonal light volumes, voxelized shadow volumes, …*
- Practical considerations for games (Valient, 30 min)
  - Realistic budgets for performance/memory
  - Upsampling techniques
    *Varying samples, bilateral upsampling, …*
  - Combination with other effects
    *AO, SSAO, …*
  - Showcases
    *AAA titles (including Killzone 2 & 3) and upcoming games*
- Conclusion, outlook, and Q&A (All, 5 min)

# Contents

# Preface

**What is this course about?** This course is a resource for efficient real-time shadow algorithms. It gives an overview of various techniques and addresses practical and game-relevant solutions. Besides conventional topics, such as hard or soft shadows, we also address recent practically-relevant topics, e.g., volumetric shadows.

We will provide the theoretical background but also discuss details on implementation issues in order to facilitate efficient realizations. These elements are of relevance to experts but also to practitioners, as they support the general understanding and provide interesting views. A particular focus will be put on budget considerations and the analysis of existing performance trade-offs; physical accuracy can sometimes be replaced by plausible shadows, and we will describe scenarios in which approximate methods are likely to work or fail. In particular, we will present showcases that illustrate the techniques behind major game titles and upcoming engines.

**Who should participate?** The course will be useful for many different fields, ranging from the game industry (as an overview and guide) over the movie industry (as a resource for previsualization techniques) to other branches, such as the visualization community (shadows have a strong impact on spatial perception). It will allow its participants to gain insight into several methods that find application in high-performance simulations and to receive practical advice.

**Where can I get more information?** The course builds upon a solid foundation in form of previous courses, as well as the recent book "Real-time Shadows" [Eisemann et al., 2011] (A K Peters/CRC Press) that appeared in 2011 and was written by four of the presenters. The book is a compendium of many topics in the realm of shadow computation on 400 pages. Furthermore, the course features one contributor from the industry, who has a strong background in shadow techniques and was a lead for several AAA titles.

Please also visit our webpage that offers additional information on shadow algorithms:

**http://www.realtimeshadows.com/**

# 1 Introduction

## 1.1 Qualitative Definition of a Shadow

When asking what a shadow is, people will give various answers and for most, you will find counterexamples of shadow phenomena that would not have been captured. Even dictionaries have a hard time coming up with proper formulations. A mathematical way to define a shadow in the context of computer graphics is to consider a point $\mathbf{p}$ in space. If each ray from the light source $\mathcal{L}$ reaches $\mathbf{p}$ without encountering any scene object (so-called *blockers* or *shadow casters*) on its way, $\mathbf{p}$ is considered *lit*, otherwise *in shadow*. In other words, we are in shadow, if from our point $\mathbf{p}$ the light source $\mathcal{L}$ is at least partially occluded. The region of space where the light source is fully occluded is called *umbra*, whereas the remaining shadow is referred to as *penumbra*. The different regions are indicated in Fig. 1.1.

## 1.2 Quantitative Definition of a Shadow

While the previous definition allows us to define where we find shadows, it does not allow us to say anything about the way its appearance is influenced by being in shadow. To clarify this point, we will place ourselves in a simple context. This first definition will be sufficient for most of this course, but we will later give an outlook on extensions. The physical interaction can be described well by a *soft shadow equation* that builds upon a modification of the *rendering equation* introduced by Kajiya [1986] where we only consider direct light from the source:

$$L_{\mathrm{o}}(\mathbf{p}, \omega) = \int_{\mathcal{L}} f_{\mathrm{r}}(\mathbf{p}, \omega, \mathbf{p} \to \mathbf{q})\, G(\mathbf{p}, \mathbf{q})\, L_{\mathrm{e}}(\mathbf{q}, \mathbf{q} \to \mathbf{p})\, V(\mathbf{p}, \mathbf{q})\, d\mathbf{q}, \tag{1.1}$$

where $L_{\mathrm{o}}(\mathbf{p}, \omega)$ is the outgoing radiance in direction $\omega$ from point $\mathbf{p}$, $f$ is the BRDF (bidirectional reflectance function) encoding the material surface properties, $G$ is a geometric term taking the configuration of source and receiver into account, $L_{\mathrm{e}}(\mathbf{q}, \mathbf{q} \to \mathbf{p})$ is the emitted energy from the source point $\mathbf{q}$ towards $\mathbf{p}$, and finally $V(\mathbf{p}, \mathbf{q})$ encodes the visibility and is zero if there is a blocker between the two points $\mathbf{q}$ and $\mathbf{p}$, and otherwise one.

If we further assume that all surfaces are Lambertian (perfectly diffuse), the BRDF becomes independent of direction, i.e. $f_{\mathrm{r}}(\mathbf{p}, \omega, \hat{\omega}) = \rho(\mathbf{p})/\pi$ where $\rho(\mathbf{p})$ denotes reflectance. Consequently,

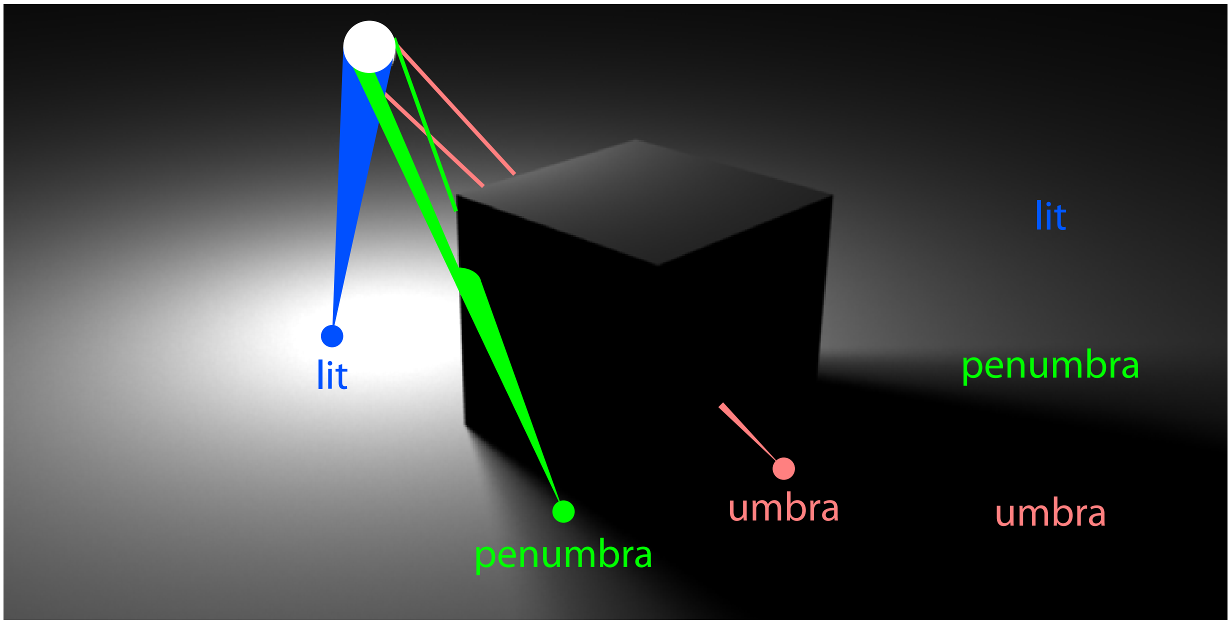



**Figure 1.1** Depending on the visibility of the source, there are three different regions in space; the umbra region, where the light is invisible, and the penumbra region, where it is partially visible constitute the shadow, the rest of the scene is lit.

the outgoing radiance $L_o$ also no longer depends on the outgoing direction. The equation simplifies to:

$$L_o(\mathbf{p}) = \frac{\rho(\mathbf{p})}{\pi} \int_{\mathcal{L}} L_e(\mathbf{q}, \mathbf{q} \to \mathbf{p})\, G(\mathbf{p}, \mathbf{q})\, V(\mathbf{p}, \mathbf{q})\, d\mathbf{q},$$

In many situations, another simplification can be applied in form of a separation of the integral:

$$L_o(\mathbf{p}) = \underbrace{\frac{\rho(\mathbf{p})}{\pi} \int_{\mathcal{L}} G(\mathbf{p}, \mathbf{q})\, d\mathbf{q}}_{\text{Shading}} \cdot \underbrace{\frac{1}{|\mathcal{L}|} \int_{\mathcal{L}} L_e(\mathbf{q}, \mathbf{q} \to \mathbf{p})\, V(\mathbf{p}, \mathbf{q})\, d\mathbf{q}}_{\text{Shadow}}.$$

Shading and shadows are hereby decoupled. Most existing shadow algorithms aim at evaluating the shadow term of the above equation, or even assume a homogenous light source of emission $\bar{L}_c$, leading to a *visibility integral* that modulates the shading and represents the actual shadow component of the equation:

$$\bar{L}_c \int_{\mathcal{L}} V(\mathbf{p}, \mathbf{q})\, d\mathbf{q}. \tag{1.2}$$

Most real-time applications, aim at solving this equation when aiming for realistic shadows. Nevertheless, some methods allow us to compute Equation 1.1 at a supplementary cost.

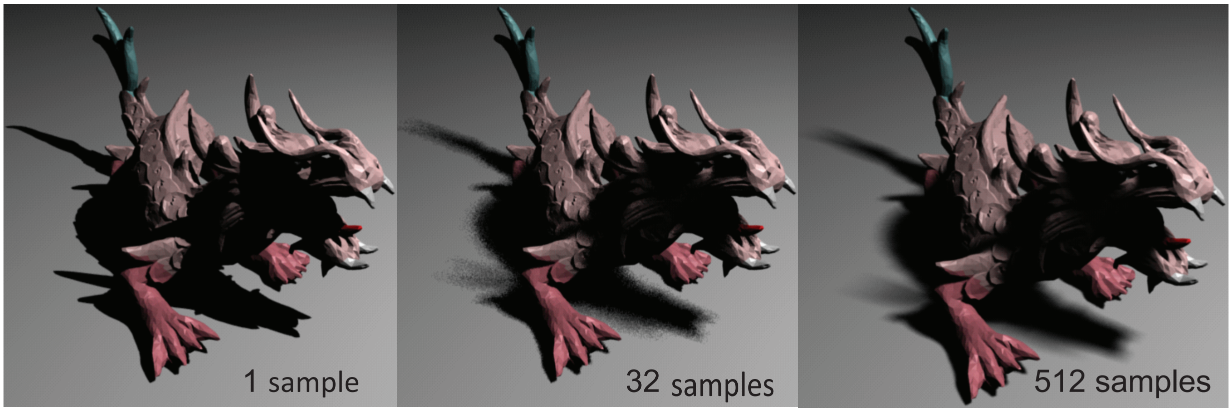



**Figure 1.2** To evaluate the shadow of an area light source, it is possible to sample the source. A single sample leads to a hard shadow (left), more samples approach the realistic solution, but an insufficient amounts of samples leads to noise (middle). Only a costly evaluation with many samples leads to a high quality picture (right). In this course, we will see several alternatives to such a brute-force solution.

## 1.3 Shadow Types and Computation

In many cases, the Equation (1.2) is solved by sampling the source point-wise. This has an interesting consequence, for a point light source, the whole integral becomes a single visibility query; shadows are binary. This is also reflected by the absence of a penumbra region. Shadows are thus *hard* because the transition between light and darkness is immediate. For volumetric or area sources, the transition is smooth, hence, the name *soft shadows*. Fig. 1.2 shows the result obtained when approximating a soft shadow by evaluating several positions on the light source.

Even though Equation (1.2) does not seem to complicated, one should notice, that it means testing several rays from each point in the scene towards the source. The outcome can no longer be locally decided on the point, but all triangles in the scene are potentially involved. This makes an efficient solution difficult.

In the following of this course, we will see various solutions of differing degrees of accuracy and performance. As we will see, the best algorithmic choice will depend heavily on the configuration of the scene, the type of light source, the wanted accuracy, the way the scene is represented, or even the viewpoint. The realm of possibilities is large and to make a good choice for your particular needs, a good comprehension, a performance and quality analysis, and a comprehensive overview are needed. Our goal is to provide this information with this course.

# 2 | Basic Algorithms

In this course section, the most basic algorithms for hard shadows will be described. Most of the more advanced techniques are derived from these techniques—including the ones targeting soft shadows. There are three main classes of such methods: *projection shadows*, *shadow maps*, and *shadow volumes*. Projection shadows typically mean that the shadow is projected onto a plane and drawn there as a dark object. This will not be further treated here. Shadow maps and shadow volumes correspond to two different ways of thinking about shadows: as places not seen by the light source and as volumes of space that are dark. The former corresponds to shadow maps, explained next, and the latter to shadow volumes.

## 2.1 Shadow Maps

The shadow maps method was introduced in 1978 [Williams, 1978]. The algorithm starts by rendering an image from the light source. Here, only the depth is stored for each pixel. This image (called the shadow map) represents all locations in space that are in light. Next, the scene is rendered from the camera. For each pixel, the fragment shader tests if the sampled point is represented in the light's view, i.e., the shadow map. If so, the point is in light. Else, the point is in shadow.

Due to the discrete resolution of the shadow map, a view sample will rarely be exactly rep-

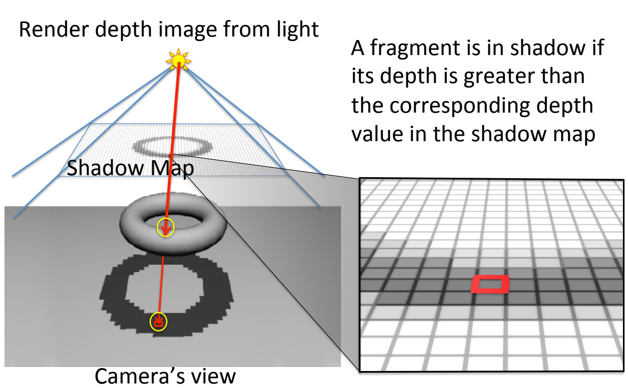
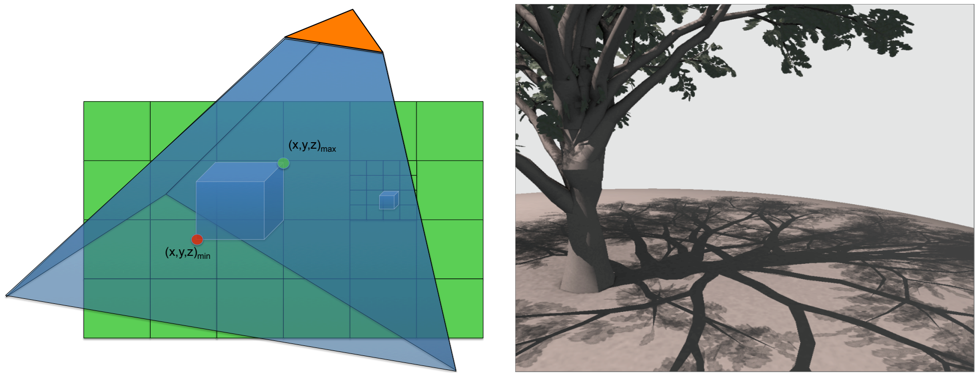



**Figure 2.1** Illustration of the shadow map algorithm.




(a) (b)

**Figure 2.2** (a) Illustrating the rasterization of per-triangle shadow volumes to a hierarchical frame buffer. (b) This efficient enables transparent shadow casters.

resented in the shadow map. This results in two problems: jagged shadows and the need to introduce a tolerance threshold for the comparison. The threshold, or *bias*, must be fine tuned for each scene, and no bias is guaranteed to exist that avoids artifacts. A too large bias results in light leakage at contact shadows, while a too small bias results in incorrect self-shadowing. Increasing the shadow map resolution can be helpful, since then a smaller bias can be used. But it does not remove the problem. Solving this is the main target of most of the more advanced methods for hard shadows, described further on in the course. Relatively simple methods exist, however, that mostly pushes the biasing problem to near the silhouette edges, as seen from the light source, of the shadow casting objects [Hourcade and Nicolas, 1985; Wang and Molnar, 1994; Woo, 1992; Weiskopf and Ertl, 2003].

## 2.2 Shadow Volumes

In its most basic form, the shadow volume algorithm creates a volume of ”space in shadow” from each triangle. Each view sample is then tested for inclusion in any such shadow volume by using the stencil buffer to perform the test [Heidmann, 1991]. Although the technique was introduced already in 1977 [Crow, 1977], it was not until 1991 that the algorithm was efficiently mapped onto graphics cards. Then, it was, however, no longer fully robust, and fixing this became an area for the next 20 years of research, resulting in for instance the *Z*-fail, ZP+, and ++ZP algorithms [Bilodeau and Songy, 1999; Carmack, 2000; Eisemann et al., 2011]. Another problem of the shadow volume algorithm is that their rasterization puts a high demand on the fill-rate capacity of the graphics hardware. This inherently makes the algorithm slower than shadow maps. Thus, major focus has been put on lowering the amount of necessary rasterization. Shadow volumes are created per object instead of per triangle [Bergeron, 1986; Aldridge and Woods, 2004; Kim et al., 2008], and culling and clamping of the volumes are used [Clark, 1976; Lloyd et al., 2004; Stich et al., 2007; Eisemann and Décoret, 2006a].

One of the most recent developments of shadow volumes is the Per-Triangle Shadow Volume algorithm [Sintorn et al., 2011]. CUDA is used to rasterize per-triangle shadow volumes onto a hierarchical frame buffer resulting in a low overdraw and that transparent shadow casters trivially can be supported (see Figure 2.2).

# 3 Hard Shadows

In the previous part of this course, the basic hard shadow algorithms have been explained. In this part, we will discuss several methods to reduce shadow-map aliasing artifacts. After analyzing aliasing in more detail and showing the different components of aliasing, we will show different strategies to reduce sampling error.

## 3.1 Shadow-Map Reparameterization

When projecting the view frustum into the shadow map, it becomes apparent that higher sampling densities are required near the viewpoint and lower sampling densities far from the viewpoint. In some cases, it is possible to apply a single transformation to the scene before projecting it into the shadow map such that the sampling density is globally changed in a useful way (Perspective Shadow Maps (PSM) [Stamminger and Drettakis, 2002]). It can be shown that a logarithmic transformation along the z-axis of the viewer provides an optimal sampling rate for the whole depth range in the view frustum [Wimmer et al., 2004], however, this requires logarithmic rasterization, which is currently infeasible.

Practical warping schemes use perspective transformations to redistribute samples towards the near plane [Wimmer et al., 2004; Martin and Tan, 2004; Chong, 2003; Chong and Gortler, 2004]. Figure 3.1 shows the idea based on light-space perspective shadow mapping (LiSPSM), where a perspective transform in the shadow-map plane is used to redistribute samples. A recent approach tries to adapt the warping locally according to the scene content while still using only a single shadow map, which is possible using a rectilinear warping grid [Rosen, 2012].

## 3.2 Global Shadow-Map Partitioning

While warping works very well in some configurations, especially if the light is overhead, there are other configurations where warping degenerates to uniform shadow mapping. A better alternative is to use more than one shadow map.

The most prominent approach and one of the most practical algorithms is to subdivide the view frustum along the z-axis, and calculate a separate equal-sized shadow map for each sub-frustum. This algorithm goes by the names of Cascaded Shadow Maps (CSM) [Engel, 2006], Parallel Split Shadow Maps (PSSM) [Zhang et al., 2006], or z-partitioning [Lloyd et al., 2006],

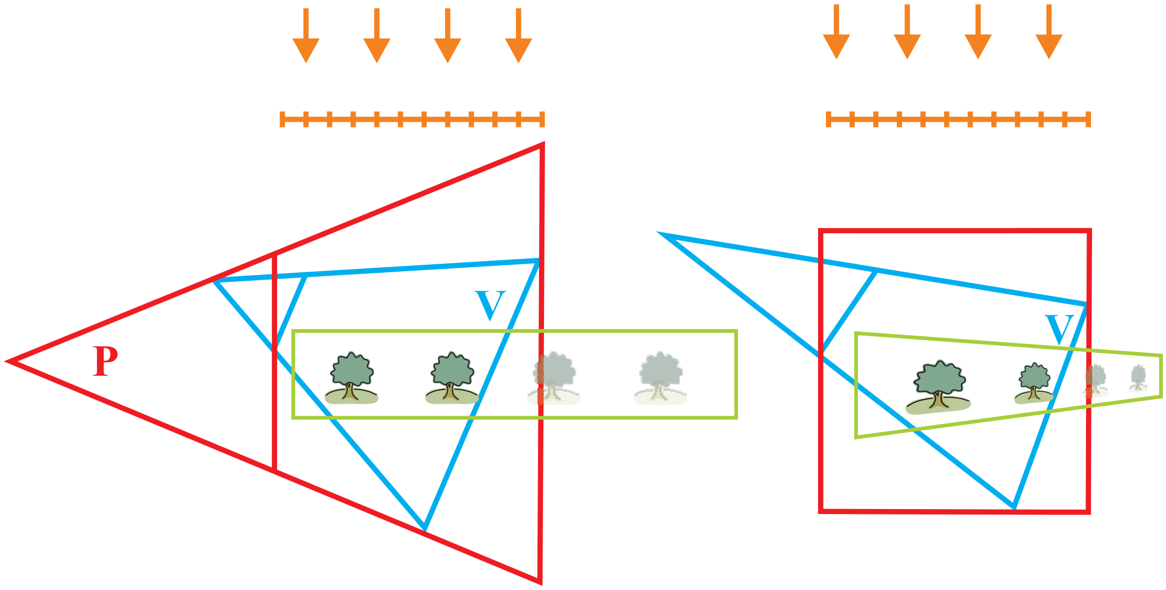



**Figure 3.1** In LiSPSM, a perspective transform $P$ is used to warp the view frustum $V$ (left). After the warp, objects near the viewer appear bigger in the shadow map and therefore receive more samples (right).

but was actually already discovered earlier [Tadamura et al., 1999, 2001]. Using this approach, the sampling density decreases for each successive partition, because the same number of shadow map samples cover a larger and larger area. The benefit can be maximized by analyzing the actual distribution of the depth values in the view frustum [Lauritzen et al., 2011] in so-called sample distribution shadow maps. Figure 3.2 shows an example configuration for PSSM.

Reparametrization and partitioning can also be combined (see Figure 3.3).

## 3.3 Adaptive Shadow-Map Partitioning

The advantage of global partitioning algorithm is that they are very fast. On the other hand, they completely ignore surface orientation and can therefore not improve undersampling due to surfaces that are viewed almost edge-on by the light source (projection aliasing).

There are a number of algorithms that try to allocate samples in a more optimal way by analyzing the scene before creating the shadow map. This inevitably incurs some overhead due to the analysis step (which often necessitates a costly read-back), but leads to much better results in general cases. Prominent examples are Adaptive Shadow Maps (ASM) [Lefohn et al., 2005], Resolution Matched Shadow Maps (RSMS) [Lefohn et al., 2007], Queried Virtual Shadow Maps (QSM) [Giegl and Wimmer, 2007b], Fitted Virtual Shadow Maps (FVSM) [Giegl and Wimmer, 2007a], and Tiled Shadow Maps (TiledSM) [Arvo, 2004]. Figure 3.4 shows the effect of one such adaptive partitioning method, QVSM, in action.

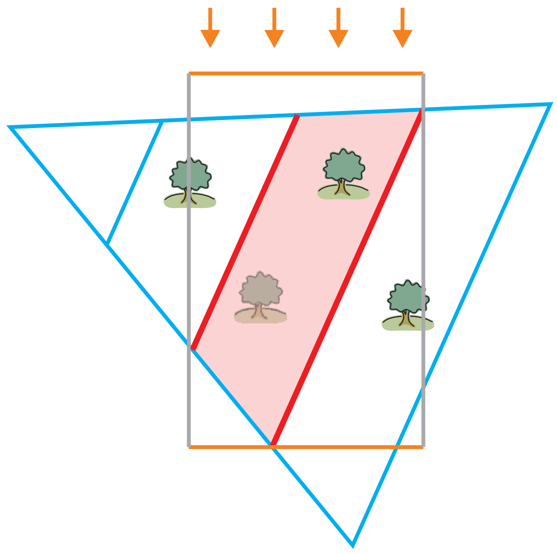


**Figure 3.2** PSSM: the shadow map for the middle of three partitions of the view frustum (side view).

## 3.4 Shadow Reconstruction

All methods discussed so far assume that shadow maps are sampled at certain positions, and reconstruction accesses these samples. However, we can also get a more accurate reconstruction of shadow edges by storing, for example, information about silhouettes in the shadow map. Notable techniques are forward shadow mapping [Zhang, 1998] and silhouette shadow maps [Sen et al., 2003], and reconstructable geometry shadow maps [Dai et al., 2008].

## 3.5 Precise Hard Shadows

The aliasing artifacts in hard shadow mapping stem from the fact that the shadow map query locations do not correspond to the shadow map sample locations. Ideally, one would like to create shadow map samples exactly in those positions that will be queried later on. Difficult as that may seem, it is actually possible and has been proposed independently by Aila and Laine [2004], and Johnson et al. [2005], implemented in hardware [Sintorn et al., 2008b], and later extended to provide efficient antialiasing [Pan et al., 2009].

## 3.6 Other Considerations

Finally, we will show how to improve shadow quality through temporal coherence [Scherzer et al., 2007], and to speed up shadow mapping for large scenes by shadow caster culling [Bittner et al., 2011].

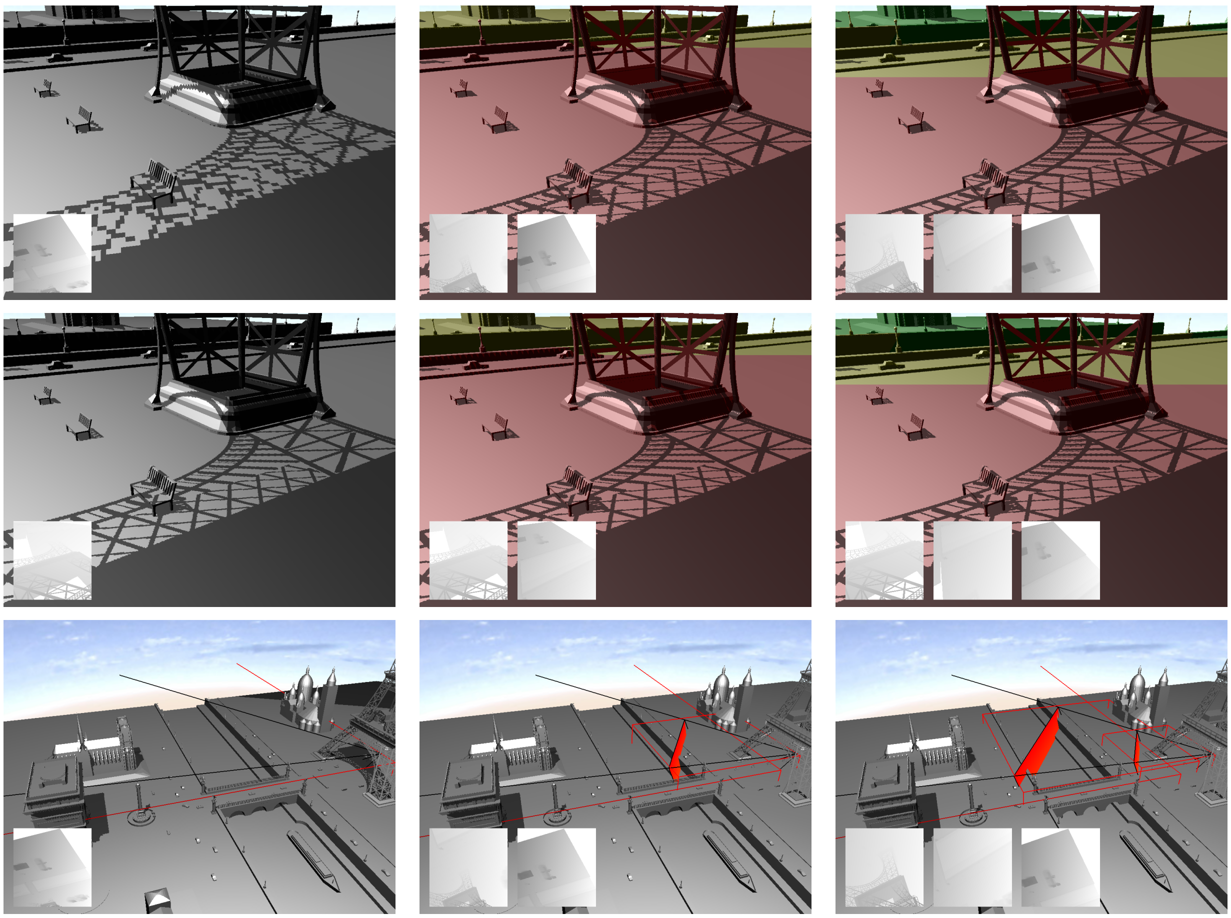


**Figure 3.3** Examples of global partitioning along the z-direction with and without warping for one to three partitions (left to right). The shadow map used for each fragment is color coded: camera view, uniform (top row); camera view, with reparametrization (middle row); and outside view showing the view frustum, the partitions, and the intersection bodies (bottom row). Inlays show the depth maps.

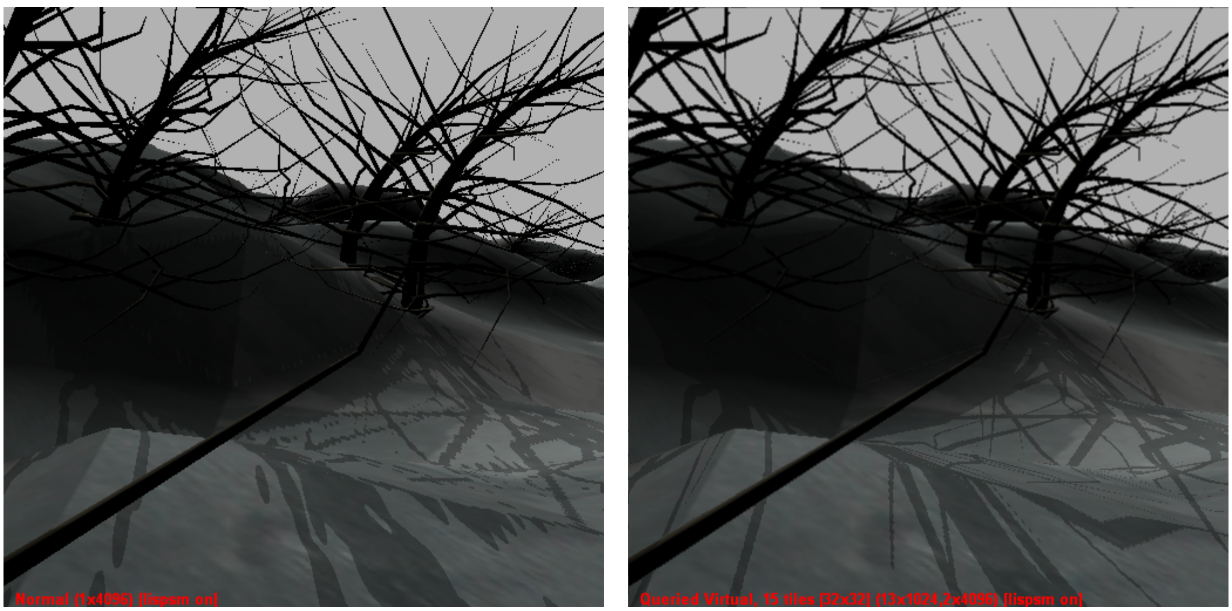



**Figure 3.4** Standard $4{,}096^2$ shadow map with perspective warping, rendering at 64 FPS (left). QVSM with a maximum refinement level of $32 \times 32$ and $1{,}024^2$ tiles, rendering at 32 FPS (right). Adaptive subdivision effectively removes aliasing in cases that are difficult for reparametrization and global partitioning methods.

# 4 Filtering Hard Shadows

In this part of the course, we discuss several filtering methods for shadow mapping, which are mainly useful for reducing resampling error. Filtering is also often used to hide the fact that the resolution of the shadow map is too low by smoothing or *blurring* the shadow boundaries, and sometimes even to provide a rough approximation to soft shadows. We start with a general technique, which quickly becomes infeasible for larger filter kernel sizes, and then show several efficient filtering techniques that rely on different levels of precomputation.

## 4.1 Introduction: Percentage-Closer Filtering

While the idea of shadow-map filtering is very close to texture mapping, in practice there is an important difference. It is in general not possible to apply a filter function to the shadow map and then shadow test the result. In that case, the depth values would be averaged, but the resulting shadows would still show the same aliasing artifacts because for each view sample, the shadow test still leads to a binary outcome.

Instead, one needs to filter the *shadow signal*, not the *depth signal*. This approach is called *percentage-closer filtering* (PCF) [Reeves et al., 1987]. Formally, it is as simple as changing the order of depth testing and filtering, i.e., for every sample in the filter kernel, a texture lookup is performed, the depth test is carried out, and only then is the filter applied.

This approach is very simple and cheap for small filter kernels. Thus, it is mostly useful when the shadow is magnified on screen. However, for larger filter kernels, this test implies a large performance penalty because the complete filter kernel needs to be evaluated for every shadow lookup. While PCF produces blurred shadows, these shadows do not correspond to soft shadows caused by an area light source, although for smaller filter kernels, the impression of a small area light source can be reasonably invoked. Figure 4.1 shows PCF for small and large filter kernels.

## 4.2 Efficient Filtering Approaches

While for texture mapping, the result of filtering with larger filters can be precomputed, for example using mip-mapping, this is not easily possible with shadow mapping. The main problem is that we need to filter the outcome of the shadow test, and not the depth signal. Since the shadow test function is not linear, we cannot change the order of computations.

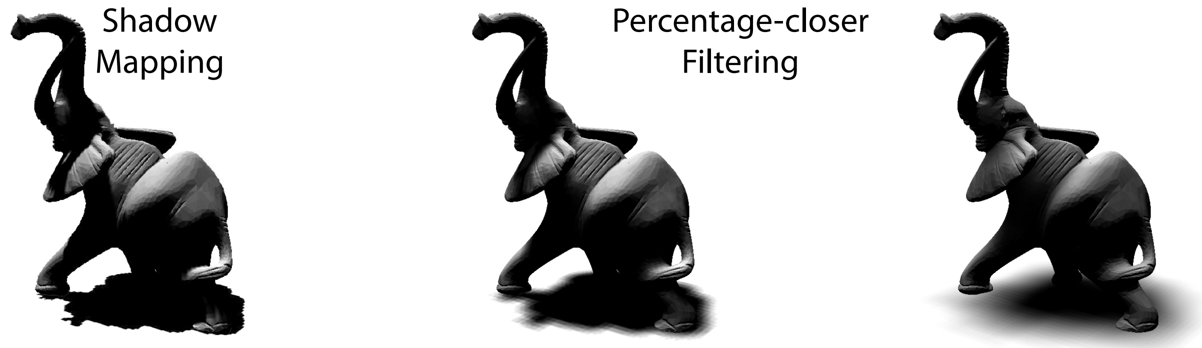



**Figure 4.1** PCF improves shadow-map aliasing (left) and results in reasonable shadows for smaller filter kernels (middle), resembling soft shadows from a small area light source. For larger filter kernels, performance is reduced, and some inconsistencies can occur (right, shadow under left foot).

There are two main ways of getting around this limitation: either we interpret the depth samples as a distribution and then model the depth comparison statistically. The other option is to approximate the depth comparison function with a linear combination of functions that are linear in the depth component.

### 4.2.1 Variance Shadow Maps

*Variance shadow maps* (VSM), introduced by Donnelly and Lauritzen [Donnelly and Lauritzen, 2006], are the first example of using statistics to facilitate precomputation of shadow-map filtering. In VSM, the depth distribution of samples that need to be evaluated by a filter kernel is modeled using first and second moments, and the percentage of samples that are hidden is estimated using the Chebyshev inequality. The first two moments correspond to the average depth and average squared depth, which can be easily precomputed in a manner similar to mipmapping. VSMs suffer from light leaks, which can be avoided using *layered variance shadow maps* (LVSM) [Lauritzen and McCool, 2008], albeit at higher cost.

### 4.2.2 Convolution Shadow Maps

*Convolution shadow maps* (CSM) [Annen et al., 2007] are the first approach to allow filter precomputation based on a linear signal-theory framework. The shadow-test function, which is a step function, is approximated by its (truncated) Fourier expansion. The coefficients of the Fourier expansion are stored in textures and can be mipmapped. Similar to VSMs, CSMs suffer from light leaks.

Figure 4.2 shows variance shadow maps and convolution shadow maps in comparison.

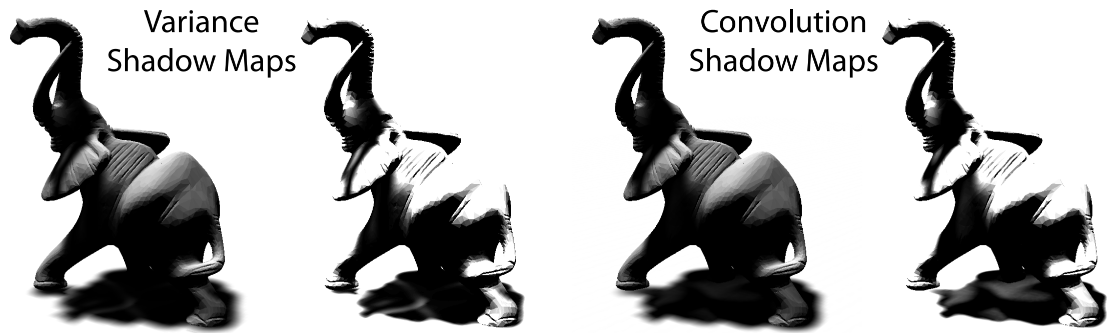



**Figure 4.2** CSM (left pair) and VSM (right pair) show light bleeding artifacts. This can be made more visible when scaling the light intensity by a factor of four (right of each pair). Newer approaches are able to reduce these artifacts significantly.

### 4.2.3 Exponential Shadow Maps

Annen et al. [Annen et al., 2008b] and Salvi [Salvi, 2008] proposed new basis functions for the CSM approach. They suggested replacing the Fourier expansion by a simple exponential. This choice voids much of the storage requirements and thus addresses one of the major issues. However, the exponential approximation is not valid where the depth is in front of the reference depth, and such cases have to be handled by resorting to PCF.

An interesting alternative is to combine the idea of using exponentials with the VSM approach. Lauritzen and McCool [Lauritzen and McCool, 2008] propose using the variance shadow-map approach and applying it to depth maps that were warped by an exponential function, leading to a fast and robust solution that mostly avoids light leaks.

# 5 Soft Shadows

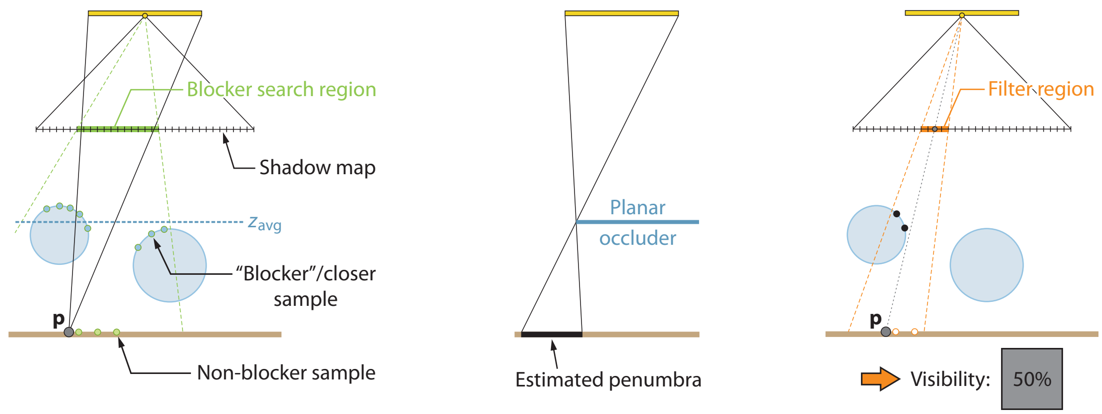
While point light sources are very popular in computer graphics, especially in the real-time domain, real light sources have a certain extent that often cannot be neglected. In particular, the shadows cast by them are not hard but feature partially lit penumbrae (transition regions from completely lit to fully occluded), leading to a soft appearance. In this part of the course, we will discuss the challenges faced when computing soft shadows and present various practical approaches of varying quality, speed, and accuracy.

Unlike with hard shadows, it is not enough to just determine whether the light source is visible from a receiver point or not. Instead, the visible fraction of the light source has to be computed, and it is primarily this point-region visibility problem that renders computing soft shadows hard and expensive. For instance, one faces the occluder fusion problem: as occluders may interact in non-trivial ways, it is generally not possible to simply process individual occluders, determine their respective occlusion factors, and then combine these to get the overall light visibility. Consequently, many research efforts were and are still dedicated to fast approximate solutions aiming at producing results that are reasonably close to the correct result or at least look plausible. On the other hand, advances in computational power and programmability nowadays enable approaches that yield accurate results at interactive rates even for complex scenes.

## 5.1 Image-Based Solutions

For the majority of real-time applications, approximate methods are currently most relevant thanks to their speed. Most of them employ an image-based scene representation, typically resorting to a standard shadow map.

One large group of approaches builds on the observation that blurring hard shadow test results yields a soft-shadow-like appearance. Percentage-closer soft shadows (PCSS) [Fernando, 2005] adaptively choose the amount of blurring using a single-planar-occluder approximation and then applies standard percentage-closer filtering (see Figure 5.1). As this scheme involves many shadow map accesses, several techniques for speeding up the computations were devised. Some advanced methods like convolution soft shadows [Annen et al., 2008a] and variance soft shadow mapping [Yang et al., 2010] employ alternative shadow map representations and prefiltering, rendering the soft-shadow computation basically a constant-time operation. Targeting higher quality, Shen et al. [2011] introduce an advanced filtering method and employ adaptive shadow-map partitioning, guided by a perceptual resolution prediction metric that exploits the




(a) Blocker search (b) Penumbra width estimation (c) Filtering

**Figure 5.1** Percentage-closer soft shadows [Fernando, 2005] first search the shadow map for blockers. Assuming a single planar occluder at their average depth $z_{avg}$, the penumbra width is derived, and a corresponding filter window size is determined. Finally, the shadow map results are filtered accordingly to get approximate soft shadows.

typically low-frequency nature of penumbrae.

A quite different technique is occlusion textures [Eisemann and Décoret, 2006b, 2008], where the scene gets decomposed into slices, representing each slice by a planar occluder. By adaptively blurring these occluders' image-based representations with a box filter [Soler and Sillion, 1998], which can efficiently be done via prefiltering, and combining them, approximate soft shadows can be obtained rapidly.

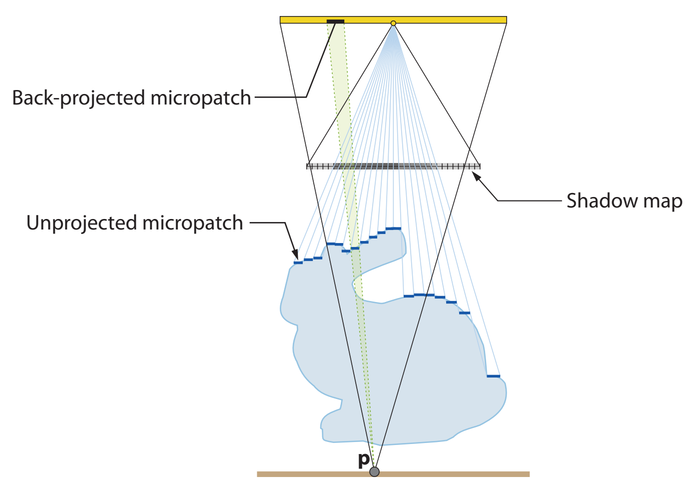



**Figure 5.2** Basic soft shadow mapping derives occluder approximations by unprojecting shadow map texels into world space. The resulting "micropatches" are backprojected onto the light source, and the occluded areas are accumulated, yielding the light's visibility from **p**.

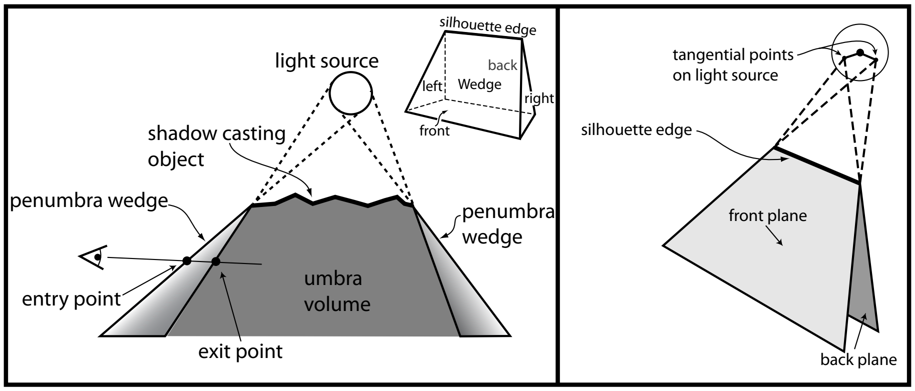



**Figure 5.3** For each light-silhouette edge, soft shadow volumes construct a penumbra wedge that encloses the resulting penumbra.

Soft shadow mapping [Atty et al., 2006; Guennebaud et al., 2006] is a rather accurate approach for which many variants exist. It employs the shadow map to reconstruct an approximation of the occluders, unprojecting the shadow map texels into world space. The resulting micro-occluders are then backprojected onto the light source, and by aggregating the occluded parts, the light's visibility is determined (see Figure 5.2). Like all previously mentioned methods, this aggregation combines scalar occlusion values for individual occluders and hence suffers from the occluder fusion problem. This is alleviated by bitmask soft shadows [Schwarz and Stamminger, 2007], where the extended light is represented by many point lights, and the binary visibility of these sample points is tracked with an occlusion bitmask. While, in principle, this enables accurate results (if a sufficiently high number of well-distributed samples is used), the inherently approximate nature of image-based representations ultimately precludes them.

## 5.2 Geometry-Based Solutions

By contrast, geometry-based solutions avoid aliasing problems of image-based approaches, but this typically comes along with a slower speed. Soft shadow volumes [Assarsson and Akenine-Möller, 2003] build on shadow volumes for hard shadows and additionally employ penumbra wedges [Akenine-Möller and Assarsson, 2002] to account for penumbra regions. These wedges are constructed for all silhouette edges and encompass the resulting penumbrae (see Figure 5.3). For each covered pixel, the edge is backprojected onto the light, and ultimately, the light area covered by the corresponding occluder is computed using Green's formula. However, the method suffers from the occluder fusion problem, leading to wrong results if occluders overlap. This is addressed by depth complexity sampling [Laine et al., 2005a] where the light area is represented by several light sample points, and a counter for each sample point is maintained, keeping track of the number of occluders overlapping the sample point. Originally developed for offline ray-tracing, a GPU variant exists as well [Forest et al., 2008].

Avoiding the high fill rate of shadow-volume-based methods, view-sample mapping [Sintorn et al., 2008b] inserts the view samples (i.e., the shadow-receiving pixel sample points) into an alias-free shadow map and then rasterizes the occluders' triangles into this map. For each shadow map entry, an occlusion bitmask is maintained, and the light sample points overlapped by a triangle are set. Ultimately, the number of occluded points yields the amount of occlusion. This method not only produces accurate results but is also reasonably fast for interactive applications. A related method is soft irregular shadow mapping [Johnson et al., 2009], which makes some compromises concerning accuracy in favor of visual smoothness, abandoning point sampling of the light visibility and resorting to silhouettes instead of triangles.

# 6 Volumetric Shadows

**Figure 6.1** Volumetric shadows in participating media can give rise to so-called *shafts of light*.

In this part of the course, we will talk about shadows in participating media. The word *participating media* means that the medium through which the light travels interacts with the light itself. In reality, the world around us is filled with participating media, e.g., air, fog, clouds and smoke. A participating medium scatters light, which means that photons bounce off the particles in the medium. The light undergoes reflections and possibly also minor refractions, for instance when hitting microscopic water drops. The light rays will bounce around before, at least some, will eventually reach the eye. This phenomenon makes the participating medium visible.

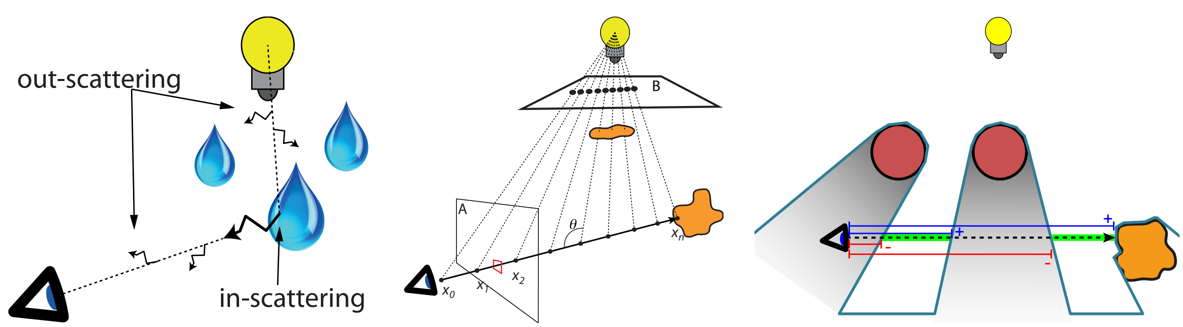
## 6.1 Single Scattering

For real-time purposes, it is common to only consider single scattering, i.e., only one light bounce is taken into account. The light travels from the light source, undergoes one reflective bounce and reaches the eye (so called *in-scattering*). This light, scattered towards the viewer and making the participating medium visible, is also referred to as *airlight*. Multiple scattering is mostly too expensive to compute in real-time. Nevertheless, for optically thin media, e.g., air, for which the transmittance is close to 100%, the single scattered rays constitute the dominating part of the visual appearance. It is also easy and common practice to account for *out scattering* along the light rays. This corresponds to attenuation of the intensity with the traveling distance. There are many ways to mathematically solve the airlight computation [Sun et al., 2005; Pegoraro et al., 2009, 2010]. There are semi-analytic solutions with texture lookups—which are used in real-time applications—and purely analytical, which are slow.

## 6.2 Approaches

Research on volumetric shadows started already in the early 1980s. Blinn introduced a model that describes light reflection for clouds and dusty surfaces consisting of many small particles [Blinn, 1982]. Soon, ray-tracing–based approaches were used to compute shadows in and by participating media [Kajiya and Von Herzen, 1984]. Shadow volumes were also tried early on to produce atmospheric shadows [Max, 1986b,a; Nishita et al., 1987].

With the introduction of programmable shading and compute shaders (e.g. CUDA), there has recently been a strong revival on how to compute the shadows in participating media with the single-scattering assumption. Many new solutions have been proposed. These are still divided into ray marching techniques [Dobashi et al., 2002; Mitchell, 2004; Imagire et al., 2007; Gautron et al., 2009; Tóth and Umenhoffer, 2009; Engelhardt and Dachsbacher, 2010; Baran et al., 2010; Chen et al., 2011; Wyman, 2010] and shadow-volume–based techniques [James, 2003; Biri et al., 2006; Wyman and Ramsey, 2008; Billeter et al., 2010]. These techniques will be presented briefly.




(a) (b) (c)

**Figure 6.2** (a) Light is scattered in the participating medium—here by microscopic water drops in the air. (b) Ray marching along a view ray to compute airlight contribution for a pixel (i.e., in-scattered light with attenuation from out scattering). (c) Alternatively, shadow-volume–based approaches can be used to compute the amount of single-scattered light towards the eye.

### 6.2.1 Ray-Marching Approaches

Ray-marching approaches step along each per-pixel view ray (see Figure 6.2(b)). For each delta step and position that is in light, the in scattering towards the eye is computed. Shadows are checked against a shadow map. The ray marching can be done by drawing alpha-blended planes [Dobashi et al., 2002; Imagire et al., 2007; Mitchell, 2004], looping in a fragment shader [Gautron et al., 2009; Tóth and Umenhoffer, 2009; Engelhardt and Dachsbacher, 2010], or using OpenCL or CUDA [Baran et al., 2010; Chen et al., 2011]. Wyman notes that shadow volume planes can be used to bound the ray marching [Wyman and Ramsey, 2008] . Tóth and Umenhoffer [Tóth and Umenhoffer, 2009] only ray marches a few samples per pixel and borrows results from nearby pixels. Chen et al. [Chen et al., 2011] instead reduces the ray marching steps by utilizing that all positions further from the light than another shadowed position, along the same light ray, have to remain in shadow. The most recent method [Wyman, 2010] is based on voxelizing the participating media, where each voxel in light stores a bit set to one, and each voxel in shadow stores a bit set to zero. The method then very rapidly computes the number of lit voxels along each eye ray by using GPU-based prefix-sums.

### 6.2.2 Shadow-Volume–Based Approaches

If shadow volumes from separate shadow casters are guaranteed to not overlap, airlight can be computed as a sum of order-independent terms by adding contribution for each front-facing shadow volume quad and subtracting for each back-facing quad. Otherwise, some sorting of the shadow quads is required. Biri et al. [Biri et al., 2006] sort the shadow volume quads, back-to-front, from the camera, while James [James, 2003] instead orders them using depth peeling. Billeter et al. notice that non-overlapping shadow volumes can be guaranteed by constructing them from a shadow map [Billeter et al., 2010].

## 6.3 Summary

We will here summarize the characteristics of the three most recent methods, of which the two first are ray-marching based [Chen et al., 2011; Wyman, 2010] and the third is shadow-volume based [Billeter et al., 2010].

All three methods are very fast and capable of producing frame rates in the order of a hundred fps for typical game scenes. The method by Chen et al. is based on utilizing coherency as much as possible to reduce the amount of ray marching that needs to be done. This method is the most versatile, since it is capable of handling textured light sources. Wyman's algorithm is probably the fastest, but is more approximate than the other two, since it does not properly consider the individual light attenuation for each voxel, i.e., the intensity and fading of the in-scattered light at each voxel. The approach by Billeter et al. avoids any ray marching and voxelization and instead computes the airlight integration using a technique very similar to shadow volumes. This one is the easiest to implement and does not require CUDA, which even makes it possible to use with WebGL.

# 7 Practical Considerations

The goal of this part of the course is to provide an insight into techniques that are often employed in professional game titles. We will investigate what happens behind the scenes and which techniques are efficient enough to be applied in practice. During the course, these elements will be refined, and we will provide statistics on the typical rendering budgets that are associated to the various aspects.

In these course notes, we will concentrate on some techniques that are used to accelerate computations and ameliorate shadow quality. In particular, we will focus on upsampling techniques and additional effects, such as ambient occlusion, that allow us to reach a more convincing illumination simulation.

## 7.1 Deferred Shading and Upsampling

The idea of deferred shading is to derive an image-based representation of the scene. In other words, one renders the scene once, while extracting a so-called *G-buffer*, a collection of images where each pixel stores information about the underlying surface, such as normals, depth, material properties, etc. Only in a second pass, these images are used to evaluate the actual lighting computation, which has proven useful in practice for several games (cf. Stalker [Shishkovtsov, 2006]). The advantage is that hidden geometry will never see their incoming light evaluated, and,

**Figure 7.1** Professional games make use of various acceleration techniques that we will analyze in this course. (Images courtesy of Guerilla Games/Sony Computer Entertainment.)

furthermore, the computations become more efficient. Basically, the calculations become more structured and map better to the hardware.

One can write several attributes of a G-buffer in a single render pass by using multiple render targets. This makes this process particularly efficient. Despite the fact that a pixel only stores attributes for one surface location, simple transparency effects are possible by relying on dithering strategies, similar to stochastic transparency [Enderton et al., 2010].

The second possibility to accelerate computations is to distribute calculations spatially over adjacent pixels. The idea is to only partly evaluate the shadow. For instance, when employing percentage-closer filtering one can evaluate a different set of samples for each pixel (to define the set of samples to evaluate, different sets are defined for different positions in the screen plane). Unfortunately, such a choice often leads to noisy results. To combat this artifact, one would like to filter the image, but a standard process would lead to visible artifacts at geometric discontinuities and clearly visible halos. In order to avoid this problem, a specialized filtering can be applied.

A good strategy is cross- or joint-bilateral filtering [Eisemann and Durand, 2004; Petschnigg et al., 2004]. The principle is to share irradiance values between pixels that are geometrically similar (share the same normal, are nearby in space); hereby, filtering is not performed across edges. Basically, the result of the filtering process for a pixel location $\mathbf{t}$ is then

$$I_{\text{filtered}}(\mathbf{t}) = \frac{\sum_{\mathbf{t}_i \in \mathcal{K}(\mathbf{t})} \omega(\mathbf{t}, \mathbf{t}_i) I(\mathbf{t}_i)}{\sum_{\mathbf{t}_i \in \mathcal{K}(\mathbf{t})} \omega(\mathbf{t}, \mathbf{t}_i)},$$

where $\mathcal{K}(\mathbf{t})$ is a neighborhood around pixel position $\mathbf{t}$ and the weights $\omega$ ensure that very different pixels (compared to $\mathbf{t}$) will not contribute to its filtered value. To test this similarity, the weights are based on geometric resemblance evaluated via the G-buffer. Mathematically, the weights can be defined as:

$$\omega(\mathbf{t}, \mathbf{t}_i) = G(\sigma_n, 1 - \text{normal}(\mathbf{t}) \cdot \text{normal}(\mathbf{t}_i))\, G(\sigma_p, \|\text{position}(\mathbf{t}) - \text{position}(\mathbf{t}_i)\|),$$

where $G$ is a Gaussian kernel, position($\cdot$) and normal($\cdot$) are the G-buffer values of the extracted position and surface normals. Hereby, only a fraction of the standard computation time is necessary.

To further increase computational efficiency, one can evaluate shadows also for a subset of the pixels in the full-resolution image. The result is then upsampled and spread to groups of pixels. In many cases, a good upsampling can be achieved by a slight adaptation of the joint/cross-bilateral filtering, the so-called joint-bilateral upsampling [Kopf et al., 2007]. Here, each pixel's irradiance is defined as a weighted sum of irradiance values in a low resolution image. Again, geometric similarity between the high and low resolution pixels affects the weights.

Finally, recent strategies making use of spatio-temporal upsampling [Herzog et al., 2010] and reprojection [Nehab et al., 2007; Sitthi-amorn et al., 2008; Herzog et al., 2010] start to become valuable assets to reuse shading over time. These solutions could even play a role for remote rendering configurations [Pajak et al., 2011].

## 7.2 Ambient Occlusion

To increase realism in the scene, it is also important to approximate complex effects, such as global illumination, where light bounces in the scene several times. Ambient occlusion approximates this effect by assuming a uniform white, hemispherical incident illumination. A survey [Méndez-Feliu and Sbert, 2009] and recent work [Bunnell, 2006; Hoberock and Jia, 2007] give a good overview of various techniques.

The most efficient strategies approximate the incoming illumination based on the depth buffer [Akenine-Möller et al., 2008]. Basically, the depth buffer from the point of view is used as a proxy of the original geometry. To evaluate how much light is impinging at a certain point a small neighborhood of surrounding pixels is evaluated. There are various techniques to derive an estimate, such as [Bavoil et al., 2008; Szirmay-Kalos et al., 2010; Oat and Sander, 2007; Huang et al., 2011].

Directional lighting effects become possible via directional occlusion [Ritschel et al., 2009], bent normals [Landis, 2002], or bent cones [Klehm et al., 2011, 2012]. The latter are defined in screen space and lead to a high performance, and variants are employed in recent AAA titles [Sousa et al., 2011].

# 8 Conclusion

Today, we have not yet found an ultimate algorithm to compute shadows. Nonetheless, for most situations, there are particular techniques that are most suitable. Furthermore, combining several methods can be a very good choice. For example, shadow-map repartition and reparameterization can be successfully combined to result in a better algorithm than both isolated strategies. Hence, an overview, such as in this course, can be very beneficial as it allows you to choose among the best options.

In the following, we will pinpoint a few good choices.

## 8.1 Hard Shadows

Hard shadows do not exist in the real world. Nonetheless, distant light sources (e.g., the sun) can often be well approximated with such techniques. An efficient accurate computation is possible by relying on shadow-volume solutions with specialized rasterization techniques [Sintorn et al., 2011]. In particular, omnidirectional light sources can be treated very easily. For a light frustum and depending on the triangle size, the irregular z-Buffer can also be an option [Johnson et al., 2005; Aila and Laine, 2004; Sintorn et al., 2008a].

Approximate solutions can rely on z-partitioning approaches [Engel, 2006; Zhang et al., 2006], where the view frustum is decomposed into several distances. Such techniques are widely used and a good compromise between quality and cost. Furthermore, partitions can even be steered from the viewpoint [Giegl and Wimmer, 2007a].

On top of partitioning, reparameterization is possible, such as light perspective shadow maps [Wimmer et al., 2004]. These come basically for free and do not inflict any performance penality. Hence, they are always a good choice for interactive applications.

A smarter reconstruction of the shadow boundary, e.g., via silhouette shadow maps [Sen et al., 2003] is another interesting option.

## 8.2 Filtered Hard Shadows

Reconstruction of the shadow signal also relates to filtering approaches. Most techniques propose accelerations of percentage-closer filtering [Reeves et al., 1987].

For low memory cost, variance shadow maps [Donnelly and Lauritzen, 2006] are a good choice. Nonetheless, light leaks may appear. Alternatives consuming more memory and resources exist [Lauritzen and McCool, 2008]. An interesting tradeoff that is exploitable in games are exponential shadow maps [Annen et al., 2008b; Salvi, 2008], whose memory consumption is acceptable, but some post treatment of artifacts might be necessary. Especially, the variant described in [Lauritzen and McCool, 2008] is a good option.

Filtered shadows are relatively cheap, but not physically based. Nonetheless, a designer can often tweak a scene in order to hide these shortcomings.

## 8.3 Soft Shadows

Soft shadows approach physically-reasonable shadow behavior and simulate penumbrae. The most efficient accurate solutions [Sintorn et al., 2008a] are too slow for games, but could be used for previsualization and be good additions to other offline processes [Laine et al., 2005b; Overbeck et al., 2007] .

For games and interactive applications approximate solutions are most adequate. PCSS-oriented solutions [Fernando, 2005] are already often applied in practice. Occlusion textures [Eisemann and Décoret, 2008] can be a practical option for scenes of a smaller extent, especially if the light source is relatively large.

## 8.4 Further Topics

In the future, other shadow-related topics will become important. Volumetric shadow effects result in god rays [Chen et al., 2011], but also semi-transparent objects, or even shadows cast from indirect sources, will play an increasingly important role.

## 8.5 Last Words . . .

We hope you enjoyed this course, and we invite you to visit our webpage:

http://www.realtimeshadows.com/.

Here, we will make the course slides available. Further, we will provide more information on recent topics and future trends. We want to keep track of the developments in shadow algorithms and be a guiding light that casts a long-lasting shadow.

# Acknowledgments

Many people helped with their input and support. Many thanks go to Louis Bavoil, Brandon Lloyd, Zhao Dong, Cyril Crassin, Tobias Ritschel, Thomas Annen, Robert Herzog, Erik Sintorn, Oliver Klehm, Emmanuel Turquin, Bert Buchholz, Aaron Lefohn, Andrew Lauritzen, Martin Eisemann, Hedlena Bezerra, Alice Peters, all of which also helped us with our book *Real-time Shadows* on which this course is based. We also thank our colleagues (Telecom ParisTech: Yves Grenier, Isabelle Bloch, Tamy Boubekeur, ...; TU Vienna: Daniel Scherzer, Werner Purgathofer, ...; Chalmers: Ola Olsson, Markus Billeter, ...). For the models we thank Marco Dabrovic, Martin Newell, the Stanford 3D Scanning Repository, INRIA, and Aim@shape. This work was partially funded by the Intel Visual Computing Institute (IVCI) at Saarland University.

# Bibliography


Timo Aila and Samuli Laine. Alias-free shadow maps. In *Proceedings of Eurographics Symposium on Rendering 2004*, pages 161–166, 2004.

Tomas Akenine-Möller and Ulf Assarsson. Approximate soft shadows on arbitrary surfaces using penumbra wedges. In *Proceedings of Eurographics Workshop on Rendering 2002*, pages 297–306, 2002.

Tomas Akenine-Möller, Eric Haines, and Natty Hoffman. *Real-Time Rendering*. A K Peters, 3rd edition, 2008.

Graham Aldridge and Eric Woods. Robust, geometry-independent shadow volumes. In *GRAPHITE '04: Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 250–253, 2004.

Thomas Annen, Tom Mertens, Philippe Bekaert, Hans-Peter Seidel, and Jan Kautz. Convolution shadow maps. In *Proceedings of Eurographics Symposium on Rendering 2007*, pages 51–60, 2007.

Thomas Annen, Zhao Dong, Tom Mertens, Philippe Bekaert, Hans-Peter Seidel, and Jan Kautz. Real-time, all-frequency shadows in dynamic scenes. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2008)*, 27(3):34:1–34:8, 2008a.

Thomas Annen, Tom Mertens, Hans-Peter Seidel, Eddy Flerackers, and Jan Kautz. Exponential shadow maps. In *Proceedings of Graphics Interface 2008*, pages 155–161, 2008b.

Jukka Arvo. Tiled shadow maps. In *Proceedings of Computer Graphics International 2004*, pages 240–246, 2004.

Ulf Assarsson and Tomas Akenine-Möller. A geometry-based soft shadow volume algorithm using graphics hardware. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2003)*, 22(3):511–520, 2003.

Lionel Atty, Nicolas Holzschuch, Marc Lapierre, Jean-Marc Hasenfratz, Charles Hansen, and François X. Sillion. Soft shadow maps: Efficient sampling of light source visibility. *Computer Graphics Forum*, 25(4):725–741, 2006.

Ilya Baran, Jiawen Chen, Jonathan Ragan-Kelley, Frédo Durand, and Jaakko Lehtinen. A hierarchical volumetric shadow algorithm for single scattering. *ACM Transactions on Graphics*, 29(6 (Proceedings of ACM SIGGRAPH Asia 2010)):178:1–178:10, 2010.

Louis Bavoil, Miguel Sainz, and Rouslan Dimitrov. Image-space horizon-based ambient occlusion. In *ACM SIGGRAPH 2008 Talks*, pages 22:1–22:1, 2008.

P. Bergeron. A general version of crow's shadow volumes. *IEEE Computer Graphics and Applications*, 6(9):17–28, 1986.

Markus Billeter, Erik Sintorn, and Ulf Assarson. Volumetric shadows using polygonal light volumes. In *Proceedings of High Performance Graphics 2010*, pages 39–45, 2010.

William Bilodeau and Mike Songy. Real time shadows, 1999. Creativity 1999, Creative Labs Inc. Sponsored game developer conferences, Los Angeles, California, and Surrey, England.

Venceslas Biri, Didier Arquès, and Sylvain Michelin. Real Time Rendering of Atmospheric Scattering and Volumetric Shadows. *Journal of WSCG*, 14(1):65–72, 2006.

Jiří Bittner, Oliver Mattausch, Ari Silvennoinen, and Michael Wimmer. Shadow caster culling for efficient shadow mapping. In *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games 2011*, pages 81–88, 2011.

James F. Blinn. Light reflection functions for simulation of clouds and dusty surfaces. *Computer Graphics*, 16(3 (Proceedings of ACM SIGGRAPH 82)):21–29, 1982. ISSN 0097-8930.

Michael Bunnell. Dynamic ambient occlusion and indirect lighting. In Matt Pharr, editor, *GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation*, pages 223–233, Reading, MA, USA, 2006. Addison-Wesley Professional. ISBN 0-321-33559-7.

John Carmack. Z-fail shadow volumes. Internet Forum, 2000.

Jiawen Chen, Ilya Baran, Frédo Durand, and Wojciech Jarosz. Real-time volumetric shadows using 1D min-max mipmaps. In *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games 2011*, pages 39–46, 2011.

Hamilton Y. Chong and Steven J. Gortler. A lixel for every pixel. In *Proceedings of Eurographics Symposium on Rendering 2004*, pages 167–172, 2004.

Hamilton Yu-Ik Chong. Real-time perspective optimal shadow maps. Senior thesis, Harvard University, 2003.

James H. Clark. Hierarchical geometric models for visible surface algorithms. *Communications of the ACM*, 19(10):547–554, 1976.

Franklin C. Crow. Shadow algorithms for computer graphics. *Computer Graphics*, 11(2 (Proceedings of ACM SIGGRAPH 77)):242–248, 1977.

Qinghua Dai, Baoguang Yang, and Jieqing Feng. Reconstructable geometry shadow maps. In *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games 2008: Posters*, page 4:1, 2008.

Yoshinori Dobashi, Tsuyoshi Yamamoto, and Tomoyuki Nishita. Interactive rendering of atmospheric scattering effects using graphics hardware. In *Proceedings of Graphics Hardware 2002*, pages 99–107, 2002.

William Donnelly and Andrew Lauritzen. Variance shadow maps. In *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games 2006*, pages 161–165, 2006.

Elmar Eisemann and Xavier Décoret. Fast scene voxelization and applications. In *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games 2006*, pages 71–78, 2006a.

Elmar Eisemann and Xavier Décoret. Plausible image based soft shadows using occlusion textures. In *Proceedings of SIBGRAPI 2006*, pages 155–162, 2006b.

Elmar Eisemann and Xavier Décoret. Occlusion textures for plausible soft shadows. *Computer Graphics Forum*, 27(1):13–23, 2008.

Elmar Eisemann and Frédo Durand. Flash photography enhancement via intrinsic relighting. *ACM Transactions on Graphics*, 23(3 (Proceedings of ACM SIGGRAPH 2004)):673–678, 2004. URL `http://artis.imag.fr/Publications/2004/ED04`.

Elmar Eisemann, Michael Schwarz, Ulf Assarsson, and Michael Wimmer. *Real-Time Shadows*. A K Peters/CRC Press, Boca Raton, FL, USA, 2011. ISBN 978-1-56881-438-4.

Eric Enderton, Erik Sintorn, Peter Shirley, and David Luebke. Stochastic transparency. In *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games 2010*, pages 157–164, 2010.

Wolfgang Engel. Cascaded shadow maps. In Wolfgang Engel, editor, *ShaderX$^5$: Advanced Rendering Techniques*, pages 197–206. Charles River Media, Hingham, MA, USA, 2006. ISBN 978-1-58450-499-3.

Thomas Engelhardt and Carsten Dachsbacher. Epipolar sampling for shadows and crepuscular rays in participating media with single scattering. In *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games 2010*, pages 119–125, 2010.

Randima Fernando. Percentage-closer soft shadows. In *ACM SIGGRAPH 2005 Sketches and Applications*, page 35, 2005.

Vincent Forest, Loïc Barthe, and Mathias Paulin. Accurate shadows by depth complexity sampling. *Computer Graphics Forum (Proceedings of Eurographics 2008)*, 27(2):663–674, 2008.

Pascal Gautron, Jean-Eudes Marvie, and Guillaume François. Volumetric shadow mapping. In *ACM SIGGRAPH 2009 Talks*, pages 49:1–49:1, 2009.

Markus Giegl and Michael Wimmer. Fitted virtual shadow maps. In *Proceedings of Graphics Interface 2007*, pages 159–168, 2007a.

Markus Giegl and Michael Wimmer. Queried virtual shadow maps. In *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games 2007*, pages 65–72, 2007b.

Gaël Guennebaud, Loïc Barthe, and Mathias Paulin. Real-time soft shadow mapping by back-projection. In *Proceedings of Eurographics Symposium on Rendering 2006*, pages 227–234, 2006.

Tim Heidmann. Real shadows real time. *IRIS Universe*, 18:28–31, 1991.

Robert Herzog, Elmar Eisemann, Karol Myszkowski, and Hans-Peter Seidel. Spatio-temporal upsampling on the GPU. In *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games 2010*, pages 91–98, 2010.

Jared Hoberock and Yuntao Jia. High-quality ambient occlusion. In Hubert Nguyen, editor, *GPU Gems 3*, pages 257–274. Addison-Wesley Professional, Reading, MA, USA, 2007. ISBN 978-0-321-51526-1.

J.-C. Hourcade and A. Nicolas. Algorithms for antialiased cast shadows. *Computers & Graphics*, 9(3):259–265, 1985.

Jing Huang, Tamy Boubekeur, Tobias Ritschel, Matthias Hollaender, and Elmar Eisemann. Separable approximation of ambient occlusion. In *Eurographics 2011 Short Papers*, 2011.

Takashi Imagire, Henry Johan, Naoki Tamura, and Tomoyuki Nishita. Anti-aliased and real-time rendering of scenes with light scattering effects. *The Visual Computer*, 23(9):935–944, 2007. ISSN 0178-2789.

Robert James. True volumetric shadows. In Jeff Lander, editor, *Graphics programming methods*, pages 353–366. Charles River Media, Rockland, MA, 2003. ISBN 1-58450-299-1.

Gregory S. Johnson, Juhyun Lee, Christopher A. Burns, and William R. Mark. The irregular z-buffer: Hardware acceleration for irregular data structures. *ACM Transactions on Graphics*, 24(4):1462–1482, 2005. ISSN 0730-0301.

Gregory S. Johnson, Warren A. Hunt, Allen Hux, William R. Mark, Christopher A. Burns, and Stephen Junkins. Soft irregular shadow mapping: Fast, high-quality, and robust soft shadows. In *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games 2009*, pages 57–66, 2009.

James T. Kajiya. The rendering equation. *Computer Graphics*, 20(4 (Proceedings of ACM SIGGRAPH 86)):143–150, 1986.

James T. Kajiya and Brian P Von Herzen. Ray tracing volume densities. *Computer Graphics*, 18 (3 (Proceedings of ACM SIGGRAPH 84)):165–174, 1984. ISSN 0097-8930.

Byungmoon Kim, Kihwan Kim, and Greg Turk. A shadow volume algorithm for opaque and transparent non-manifold casters. *Journal of Graphics Tools*, 13(3):1–14, 2008.

Oliver Klehm, Tobias Ritschel, Elmar Eisemann, and Hans-Peter Seidel. Bent normals and cones in screen-space. In *Vision, Modeling and Visualization Workshop*, 2011.

Oliver Klehm, Tobias Ritschel, Elmar Eisemann, and Hans-Peter Seidel. Screen-space bent cones: A practical approach. In Wolfgang Engel, editor, *GPU Pro$^3$: Advanced Rendering Techniques*, pages 191–207. A K Peters/CRC Press, Boca Raton, FL, USA, 2012.

Johannes Kopf, Michael F. Cohen, Dani Lischinski, and Matt Uyttendaele. Joint bilateral upsampling. *ACM Transactions on Graphics*, 26(3 (Proceedings of ACM SIGGRAPH 2007)): 96:1–96:5, 2007.

Samuli Laine, Timo Aila, Ulf Assarsson, Jaakko Lehtinen, and Tomas Akenine-Möller. Soft shadow volumes for ray tracing. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2005)*, 24(3):1156–1165, 2005a.

Samuli Laine, Timo Aila, Ulf Assarsson, Jaakko Lehtinen, and Tomas Akenine-Möller. Soft shadow volumes for ray tracing. *ACM Transactions on Graphics*, 24(3 (Proceedings of ACM SIGGRAPH 2005)):1156–1165, 2005b.

Hayden Landis. Production-ready global illumination, 2002. In *ACM SIGGRAPH 2002 Course Notes*, RenderMan in Production.

Andrew Lauritzen and Michael McCool. Layered variance shadow maps. In *Proceedings of Graphics Interface 2008*, pages 139–146, 2008.

Andrew Lauritzen, Marco Salvi, and Aaron Lefohn. Sample distribution shadow maps. In *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games 2011*, pages 97–102, 2011.

Aaron Lefohn, Shubhabrata Sengupta, Joe Kniss, Robert Strzodka, and John D. Owens. Dynamic adaptive shadow maps on graphics hardware. In *ACM SIGGRAPH 2005 Sketches and Applications*, page 13, 2005.

Aaron E. Lefohn, Shubhabrata Sengupta, and John D. Owens. Resolution matched shadow maps. *ACM Transactions on Graphics*, 26(4):20:1–20:17, 2007.

D. Brandon Lloyd, Jeremy Wendt, Naga K. Govindaraju, and Dinesh Manocha. CC shadow volumes. In *Proceedings of Eurographics Symposium on Rendering 2004*, pages 197–205, 2004.

D. Brandon Lloyd, David Tuft, Sung-eui Yoon, and Dinesh Manocha. Warping and partitioning for low error shadow maps. In *Proceedings of Eurographics Symposium on Rendering 2006*, pages 215–226, 2006.

Tobias Martin and Tiow-Seng Tan. Anti-aliasing and continuity with trapezoidal shadow maps. In *Proceedings of Eurographics Symposium on Rendering 2004*, pages 153–160, 2004.

Nelson Max. Light diffusion through clouds and haze. *Computer Vision, Graphics, and Image Processing*, 33(3):280–292, 1986a.

Nelson L. Max. Atmospheric illumination and shadows. *Computer Graphics*, 20(4 (Proceedings of ACM SIGGRAPH 86)):117–124, 1986b. ISSN 0097-8930.

Àlex Méndez-Feliu and Mateu Sbert. From obscurances to ambient occlusion: A survey. *The Visual Computer*, 25(2):181–196, 2009.

Jason Mitchell. Light shafts: Rendering shadows in participating media, 2004. Presentation, *Game Developers Conference 2004*. http://developer.amd.com/media/gpu_assets/Mitchell_LightShafts.pdf.

Diego Nehab, Pedro V. Sander, Jason Lawrence, Natalya Tatarchuk, and John R. Isidoro. Accelerating real-time shading with reverse reprojection caching. In *Proceedings of Graphics Hardware 2007*, pages 25–35, 2007.

Tomoyuki Nishita, Yasuhiro Miyawaki, and Eihachiro Nakamae. A shading model for atmospheric scattering considering luminous intensity distribution of light sources. *Computer Graphics*, 21(4 (Proceedings of ACM SIGGRAPH 87)):303–310, 1987. ISSN 0097-8930.

Christopher Oat and Pedro V. Sander. Ambient aperture lighting. In *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games 2007*, pages 61–64, 2007.

Ryan Overbeck, Ravi Ramamoorthi, and William R. Mark. A real-time beam tracer with application to exact soft shadows. In *Proceedings of Eurographics Symposium on Rendering 2007*, pages 85–98, 2007.

David Pajak, Robert Herzog, Elmar Eisemann, Karol Myszkowski, and Hans-Peter Seidel. Scalable remote rendering with depth and motion-flow augmented streaming. *Computer Graphics Forum*, 30(2 (Proceedings of Eurographics 2011)):415–424, 2011.

Minghao Pan, Rui Wang, Weifeng Chen, Kun Zhou, and Hujun Bao. Fast, sub-pixel antialiased shadow maps. *Computer Graphics Forum*, 28(7 (Proceedings of Pacific Graphics 2009)): 1927–1934, 2009.

Vincent Pegoraro, Mathias Schott, and Steven G. Parker. An analytical approach to single scattering for anisotropic media and light distributions. In *Proceedings of Graphics Interface 2009*, pages 71–77, 2009.

Vincent Pegoraro, Mathias Schott, and Steven G. Parker. A closed-form solution to single scattering for general phase functions and light distributions. *Computer Graphics Forum*, 29(4 (Proceedings of Eurographics Symposium on Rendering 2010)):1365–1374, 2010.

Georg Petschnigg, Richard Szeliski, Maneesh Agrawala, Michael Cohen, Hugues Hoppe, and Kentaro Toyama. Digital photography with flash and no-flash image pairs. *ACM Transactions on Graphics*, 23(3 (Proceedings of ACM SIGGRAPH 2004)):664–672, 2004.

William T. Reeves, David H. Salesin, and Robert L. Cook. Rendering antialiased shadows with depth maps. *Computer Graphics*, 21(4 (Proceedings of ACM SIGGRAPH 87)):283–291, 1987.

Tobias Ritschel, Thorsten Grosch, and Hans-Peter Seidel. Approximating dynamic global illumination in image space. In *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games 2009*, pages 75–82, 2009.

Paul Rosen. Rectilinear texture warping for fast adaptive shadow mapping. In *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games 2012*, pages 151–158, 2012.

Marco Salvi. Rendering filtered shadows with exponential shadow maps. In Wolfgang Engel, editor, *ShaderX$^6$: Advanced Rendering Techniques*, pages 257–274. Charles River Media, Hingham, MA, USA, 2008. ISBN 978-1-58450-544-0.

Daniel Scherzer, Stefan Jeschke, and Michael Wimmer. Pixel-correct shadow maps with temporal reprojection and shadow test confidence. In *Proceedings of Eurographics Symposium on Rendering 2007*, pages 45–50, 2007.

Michael Schwarz and Marc Stamminger. Bitmask soft shadows. *Computer Graphics Forum (Proceedings of Eurographics 2007)*, 26(3):515–524, 2007.

Pradeep Sen, Mike Cammarano, and Pat Hanrahan. Shadow silhouette maps. *ACM Transactions on Graphics*, 22(3 (Proceedings of ACM SIGGRAPH 2003)):521–526, 2003.

Li Shen, Gaël Guennebaud, Baoguang Yang, and Jieqing Feng. Predicted virtual soft shadow maps with high quality filtering. *Computer Graphics Forum*, 30(2 (Proceedings of Eurographics 2011)):493–502, 2011.

Oles Shishkovtsov. Deferred shading in S.T.A.L.K.E.R. In Matt Pharr, editor, *GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation*, pages 143–166. Addison-Wesley Professional, Reading, MA, USA, 2006. ISBN 0-321-33559-7.

Erik Sintorn, Elmar Eisemann, and Ulf Assarsson. Sample based visibility for soft shadows using alias-free shadow maps. *Computer Graphics Forum*, 27(4 (Proceedings of Eurographics Symposium on Rendering 2008)):1285–1292, 2008a.

Erik Sintorn, Elmar Eisemann, and Ulf Assarsson. Sample based visibility for soft shadows using alias-free shadow maps. *Computer Graphics Forum (Proceedings of Eurographics Symposium on Rendering 2008)*, 27(4):1285–1292, 2008b.

Erik Sintorn, Ola Olsson, and Ulf Assarsson. An efficient alias-free shadow algorithm for opaque and transparent objects using per-triangle shadow volumes. *ACM Transactions on Graphics*, 30(6 (Proceedings of ACM SIGGRAPH Asia 2011)):153:1–153:10, 2011.

Pitchaya Sitthi-amorn, Jason Lawrence, Lei Yang, Pedro V. Sander, Diego Nehab, and Jiahe Xi. Automated reprojection-based pixel shader optimization. *ACM Transactions on Graphics*, 27 (5 (Proceedings of ACM SIGGRAPH Asia 2008)):127:1–127:11, 2008.

Cyril Soler and François X. Sillion. Fast calculation of soft shadow textures using convolution. In *Proceedings of ACM SIGGRAPH 98*, pages 321–332, 1998.

Tiago Sousa, Nickolay Kasyan, and Nicolas Schulz. Secrets of CryENGINE 3 graphics technology, 2011. In *ACM SIGGRAPH 2011 Courses*, Advances in Real-Time Rendering in 3D Graphics and Games.

Marc Stamminger and George Drettakis. Perspective shadow maps. *ACM Transactions on Graphics*, 21(3 (Proceedings of ACM SIGGRAPH 2002)):557–562, 2002.

Martin Stich, Carsten Wächter, and Alexander Keller. Efficient and robust shadow volumes using hierarchical occlusion culling and geometry shaders. In Hubert Nguyen, editor, *GPU Gems 3*, pages 239–256. Addison-Wesley Professional, Reading, MA, USA, 2007. ISBN 978-0-321-51526-1.

Bo Sun, Ravi Ramamoorthi, Srinivasa G. Narasimhan, and Shree K. Nayar. A practical analytic single scattering model for real time rendering. *ACM Transactions on Graphics*, 24(3 (Proceedings of ACM SIGGRAPH 2005)):1040–1049, 2005. ISSN 0730-0301.

László Szirmay-Kalos, Tamás Umenhoffer, Balázs Tóth, László Szécsi, and Mateu Sbert. Volumetric ambient occlusion for real-time rendering and games. *IEEE Computer Graphics and Applications*, 30:70–79, 2010. ISSN 0272-1716. URL `http://dx.doi.org/10.1109/MCG.2010.19`.

Katsumi Tadamura, Xueying Qin, Guofang Jiao, and Eihachiro Nakamae. Rendering optimal solar shadows using plural sunlight depth buffers. In *Proceedings of Computer Graphics International 1999*, pages 166–173, 1999.

Katsumi Tadamura, Xueying Qin, Guofang Jiao, and Eihachiro Nakamae. Rendering optimal solar shadows with plural sunlight depth buffers. *The Visual Computer*, 17(2):76–90, 2001.

Balàzs Tóth and Tamás Umenhoffer. Real-time volumetric lighting in participating media. In *Eurographics 2009 Short Papers*, 2009.

Yulan Wang and Steven Molnar. Second-depth shadow mapping. Technical Report TR 94-019, University of North Carolina at Chapel Hill, 1994.

D. Weiskopf and T. Ertl. Shadow Mapping Based on Dual Depth Layers. In *Eurographics 2003 Short Papers*, pages 53–60, 2003.

Lance Williams. Casting curved shadows on curved surfaces. *Computer Graphics*, 12(3 (Proceedings of ACM SIGGRAPH 78)):270–274, 1978.

Michael Wimmer, Daniel Scherzer, and Werner Purgathofer. Light space perspective shadow maps. In *Proceedings of Eurographics Symposium on Rendering 2004*, pages 143–152, 2004.

Andrew Woo. The shadow depth map revisited. In David Kirk, editor, *Graphics Gems III*, pages 338–342. Academic Press, Boston, MA, 1992. ISBN 0-12-409673-5.

Chris Wyman. Interactive voxelized epipolar shadow volumes. In *ACM SIGGRAPH ASIA 2010 Sketches*, pages 53:1–53:2, 2010.

Chris Wyman and Shaun Ramsey. Interactive volumetric shadows in participating media with single scattering. In *Proceedings of the IEEE Symposium on Interactive Ray Tracing 2008*, pages 87–92, 2008.

Baoguang Yang, Zhao Dong, Jieqing Feng, Hans-Peter Seidel, and Jan Kautz. Variance soft shadow mapping. *Computer Graphics Forum (Proceedings of Pacific Graphics 2010)*, 29(7): 2127–2134, 2010.

Fan Zhang, Hanqiu Sun, Leilei Xu, and Lee Kit Lun. Parallel-split shadow maps for large-scale virtual environments. In *Proceedings of ACM International Conference on Virtual Reality Continuum and Its Applications 2006*, pages 311–318, 2006.

Hansong Zhang. Forward shadow mapping. In *Proceedings of Eurographics Workshop on Rendering 1998*, pages 131–138, 1998.