Real-Time Ray-Traced Soft Shadows of Environmental Lighting by Conical Ray Culling

YANG XU, YUANFA JIANG, JUNBO ZHANG, KANG LI*, and GUOHUA GENG, School of Information Science and Technology, Northwest University, China

Fig. 1. Real-time rendering results of two scenes using our proposed ray-traced soft shadows of environmental lighting by conical ray culling. Left: two characters on a plane illuminated by a global Grace Cathedral environment map. Right: two characters and the Crytek Sponza Atrium model under the Uffizi Gallery environment map. The Crytek Sponza Atrium is illuminated by diffuse precomputed radiance transfer, and the characters are illuminated by an irradiance volume represented by 3D textures.

Soft shadows of environmental lighting provide important visual cues in realistic rendering. However, rendering of soft shadows of environmental lighting in real-time is difficult because evaluating the visibility function is challenging. In this work, we present a method to render soft shadows of environmental lighting at real-time frame rates based on hardware-accelerated ray tracing. We assume that the scene contains both static and dynamic objects. To composite the soft shadows cast by dynamic objects with the precomputed lighting of static objects, the incident irradiance occluded by dynamic objects, which is obtained by accumulating the occluded incident radiances over the hemisphere using ray tracing, is subtracted from the precomputed incident irradiance. Conical ray culling is proposed to exclude the rays that cannot intersect dynamic objects, which significantly improves rendering efficiency. Rendering results demonstrate that our proposed method can achieve real-time rendering of soft shadows of environmental lighting cast by dynamic objects.

CCS Concepts: • **Computing methodologies** → **Rendering**; **Ray tracing**; *Rasterization.*

Additional Key Words and Phrases: Real-time rendering, soft shadows, environmental lighting, spherical harmonics

ACM Reference Format:
Yang Xu, Yuanfa Jiang, Junbo Zhang, Kang Li, and Guohua Geng. 2022. Real-Time Ray-Traced Soft Shadows of Environmental Lighting by Conical Ray Culling. *Proc. ACM Comput. Graph. Interact. Tech.* 5, 1 (May 2022), 15 pages. https://doi.org/10.1145/3522617

*Corresponding author

Authors' address: Yang Xu, xuyang@nwu.edu.cn; Yuanfa Jiang, hermesjang@foxmail.com; Junbo Zhang, zhangjunbo@stumail.nwu.edu.cn; Kang Li, likang@nwu.edu.cn; Guohua Geng, ghgeng@nwu.edu.cn, School of Information Science and Technology, Northwest University, Xi'an, China.

1 INTRODUCTION

Soft shadows from environmental light sources play an important role in realistic rendering because they provide significant visual cues about the spatial relationships of objects in a scene. Real-time rendering of soft shadows of environmental lighting in dynamic scenes is difficult because the computation of the visibility function is challenging when the objects in a scene are animated. Ray tracing can be applied to estimate the visibility function by shooting a set of rays over the hemisphere at a surface point. However, real-time frame rates cannot be achieved when a large number of ray samples are used, even if hardware-accelerated ray tracing is utilized, and insufficient ray samples lead to noisy rendering results. Recently, temporal filtering [Chaitanya et al. 2017; Hasselgren et al. 2020; Schied et al. 2017] is applied to denoise the rendering results with low sample numbers. However, temporal accumulation leads to temporal artifacts such as ghosting and flickering, especially in dynamic scenes.

In this work, we present a method to render soft shadows from environmental light sources at real-time frame rates using hardware-accelerated ray tracing. We assume that the scene contains both static and dynamic (deformable) objects. Since the global illumination for static scenes has been well addressed by precomputed global illumination solutions such as lightmaps and *precomputed radiance transfer* (PRT) [Sloan et al. 2002], only the self-shadows of dynamic objects, the shadows among dynamic objects, and the shadows between dynamic and static objects should be rendered. To composite the soft shadows cast by dynamic objects with the precomputed lighting, the incident irradiance at a surface point can be split into two terms: (1) the incident irradiance of precomputed/unshadowed environmental lighting; (2) the incident irradiance occluded by dynamic objects representing the soft shadows of environmental lighting, which can be acquired by accumulating the occluded incident radiances over the hemisphere. Hence, the composite incident irradiance can be obtained by subtracting the incident irradiance occluded by dynamic objects from the precomputed/unshadowed incident irradiance. The environmental light source can be either a global environment map or an irradiance volume represented by 3D textures.

To compute the irradiance occluded by dynamic objects, only the rays that can intersect dynamic objects are required to be traced. Based on this observation, conical ray culling is performed at each surface point to exclude the rays whose directions are outside the circular cones defined by the surface point and the bounding spheres of dynamic objects. Furthermore, the ray segments beyond the cones can be excluded as well. Consequently, fewer and shorter rays are required to be traced, which significantly improves the efficiency of rendering soft shadows of environmental lighting cast by dynamic objects.

Our main contributions are:

- a method to render soft shadows of environmental lighting, which can be easily integrated with the baked global illumination solutions for static scenes, such as lightmaps and PRT,
- conical ray culling, which can significantly improve rendering efficiency based on the observation that only the rays that can intersect dynamic objects are required to be traced.

With our proposed conical ray culling, ghosting and flickering artifacts can be completely avoided because the number of ray samples can be high enough. Thus, temporal filtering of the noise caused by an insufficient number of samples is not required. High-frequency contact shadows can also be kept because a small spatial filtering kernel can suppress the noise, which prevents overblurring lighting details.

2 RELATED WORK

Precomputed methods. Sloan et al. [2002] proposed PRT which precomputes a transfer function mapping the distant incident lighting from the environment map to the outgoing radiance. SH

coefficients are used to represent both the environment map and the transfer function. At runtime, the outgoing radiance can be obtained by a simple dot product for diffuse surfaces or a matrix-vector multiplication for glossy surfaces. They also proposed neighborhood transfer, which stores transfer matrices in a volume surrounding the object to support single dynamic rigid object. Zhou et al. [2005] precomputed *shadow fields* around dynamic rigid objects, which store SH visibility vectors. The visibility at each surface point is accumulated by computing the SH products over all the shadow fields. Xin et al. [2021] sped up the rendering of mid-frequency shadow fields by conducting SH products in the Fourier space. Kontkanen and Laine [2005] precomputed *ambient occlusion fields* around the rigid objects, which store solid angles and average occluded directions. At runtime, ambient occlusion (AO) is determined by retrieving and combining the precomputed values from the fields. Precomputed methods only support static or rigid objects under dynamic environmental lighting.

Data-driven methods. Kontkanen and Aila [2006] learned a linear model mapping the poses of an animated character to AO values over the entire space of character poses. Kirk and Arikan [2007] learned a multilinear model over a local subspace of character poses. Nowrouzezahrai et al. [2007] learned a reduced dimensional linear model mapping the poses of an animated character to diffuse PRT coefficients to reproduce soft self-shadows of distant environmental lighting. Keinert et al. [2018] learned a regression model mapping a small number of signed distance samples to AO values by training a feed-forward neural network. Le et al. [2019] learned a two-layer model by bi-level regression. The non-linear layer transforms proxy spheres and key points according to the current pose of a character, and the linear layer interpolates the per-vertex AO from the values at key points. Li et al. [2019] presented a compact representation of PRT for deformable objects, which solves diverse non-linear deformations. A deep *convolutional neural network* (CNN) is trained to predict the transfer function for a given pose at runtime.

Proxy-based methods. Kautz et al. [2004] proposed a hemispherical rasterizer that can efficiently compute the visibility by rasterizing spherical blocker triangles into bitmasks. Bunnell [2005] approximated the surface geometry with a hierarchy of disks and computed AO using these disks at each vertex by two passes, which can be applied to deformable objects. Based on Bunnell's method, Hoberock and Jia [2007] eliminated the disk-shaped artifacts by interpolating the contribution between parent and child in a fuzzy zone and the pinching artifacts by evaluating the form factors using actual mesh triangles instead of disks at the lowest level. Ren et al. [2006] approximated deformable objects with a set of spheres. Cheap SH additions instead of expensive SH products are used to accumulate the visibility in log SH space, and then *spherical harmonic exponentiation* (SHEXP) is used to obtain the SH visibility. Shanmugam and Arikan [2007] splatted spheres into a receiver buffer to accumulate AO. Sloan et al. [2007] extended SHEXP by splatting spheres into a receiver buffer and accumulating indirect radiance from the spherical proxies. Guerrero et al. [2008] extended SHEXP by regarding spheres as spherical light to support diffuse indirect illumination. Giraud and Nowrouzezahrai [2015] extended SHEXP to render soft shadows of environmental lighting from a dynamic height field and represented the BRDF in log SH space to avoid expensive triple-product shading. Iwanicki [2013] split soft shadows of environmental lighting into an ambient term defined by the cosine-weighted percentage of the hemisphere occluded by a sphere and a directional term represented by the area of the intersection between a sphere and a cone with the direction of the dominant lighting. Proxy-based methods suffer from inaccurate representation of the original geometries, resulting in incorrect soft shadows, especially contact shadows.

Screen-space methods. Mittring [2007] proposed screen-space AO (SSAO), which compares the sample points inside a sphere around the surface point against the depth buffer to estimate AO by the ratio of visible to occluded sample points. Bavoil and Sainz [2008] presented horizon-based AO (HBAO), which performs ray marching on the depth buffer along several directions to compute

horizon angles indicating the amounts of occlusion in those directions. Ritschel et al. [2009] extended SSAO to screen-space directional occlusion (SSDO), which accounts for the direction of incoming light to render directional shadows. However, the hidden geometries or the geometries outside the frustum cannot contribute to AO because the depth buffer only captures the first visible surfaces inside the frustum, resulting in an underestimation of AO and inconsistencies as the camera moves. Multiple layers [Bavoil and Sainz 2009; Mara et al. 2016; Ritschel et al. 2009], multiple views [Ritschel et al. 2009; Vardis et al. 2013], and enlarged field of view [Bavoil and Sainz 2009] are suggested to provide the missing scene information. Vermeer et al. [2021] proposed stochastic-depth AO, which uses a stochastic depth map where each pixel stores depth from random depth layers. Therefore, the hidden geometries can be taken into account.

Ray-traced methods. With the introduction of hardware-accelerated ray tracing, object space ray-traced AO (RTAO) and Monte Carlo path tracing can be efficiently performed at a low sample number on the GPU and used in real-time applications. Rendering results with low numbers of samples can be denoised by temporal filtering [Chaitanya et al. 2017; Hasselgren et al. 2020; Schied et al. 2017], whereas temporal artifacts such as ghosting and flickering are introduced by temporal accumulation when the scene is animated.

3 PRINCIPLE

3.1 Soft Shadows of Environmental Lighting

Assuming that the entire scene is illuminated by an environmental light source and ignoring indirect illumination in the scene, the outgoing radiance L_o at a surface point $\boldsymbol{p}$ in the direction $\boldsymbol{\omega}_\mathrm{o}$ can be expressed as

$$L_\mathrm{o}(\boldsymbol{p},\boldsymbol{\omega}_\mathrm{o}) = \int_{\Omega^+} L_\mathrm{i}(\boldsymbol{p},\boldsymbol{\omega}_\mathrm{i}) V(\boldsymbol{p},\boldsymbol{\omega}_\mathrm{i}) f(\boldsymbol{p},\boldsymbol{\omega}_\mathrm{i},\boldsymbol{\omega}_\mathrm{o}) \cos\theta_\mathrm{i} \mathrm{d}\boldsymbol{\omega}_\mathrm{i}, \tag{1}$$

where Ω^+ denotes the upper hemisphere, $L_\mathrm{i}(\boldsymbol{p},\boldsymbol{\omega}_\mathrm{i})$ is the incident radiance from the environmental light source in the direction ω_i, $V(\boldsymbol{p},\boldsymbol{\omega}_\mathrm{i})$ is the visibility function between $\boldsymbol{p}$ and the environmental light source, $f(\boldsymbol{p},\boldsymbol{\omega}_\mathrm{i},\boldsymbol{\omega}_\mathrm{o})$ is the BRDF of $\boldsymbol{p}$, and θ_i is the angle between $\boldsymbol{\omega}_\mathrm{i}$ and the surface normal at $\boldsymbol{p}$. If the surface is diffuse, $L_\mathrm{o}(\boldsymbol{p},\boldsymbol{\omega}_\mathrm{o})$ can be expressed as

$$L_\mathrm{o}(\boldsymbol{p},\boldsymbol{\omega}_\mathrm{o}) = \frac{\rho}{\pi} \int_{\Omega^+} L_\mathrm{i}(\boldsymbol{p},\boldsymbol{\omega}_\mathrm{i}) V(\boldsymbol{p},\boldsymbol{\omega}_\mathrm{i}) \cos\theta_\mathrm{i} \mathrm{d}\boldsymbol{\omega}_\mathrm{i} = \frac{\rho}{\pi} E(\boldsymbol{p}), \tag{2}$$

where ρ denotes the diffuse albedo and $E(\boldsymbol{p})$ is the incident irradiance at $\boldsymbol{p}$.

Ray tracing can be used to compute $E(\boldsymbol{p})$ by estimating the visibility function $V(\boldsymbol{p},\boldsymbol{\omega}_\mathrm{i})$ with Monte Carlo sampling. As exhibited in Figure 2, a number of rays originated from $\boldsymbol{p}$ are generated over the hemisphere oriented according to the surface normal of $\boldsymbol{p}$. If a ray does not intersect the scene geometry, the incident radiance coming from the environmental light source in the direction of the ray contributes to the incident irradiance at $\boldsymbol{p}$.

3.2 Occluded Irradiance

In our proposed method, the objects in the scene are classified into two categories: static objects and dynamic objects. Static objects can be illuminated by precomputed global illumination solutions such as lightmaps and PRT. Therefore, only the self-shadows of dynamic objects, the shadows among dynamic objects, and the shadows between dynamic objects and static objects are required to be computed at runtime. To composite the soft shadows cast by dynamic objects with the

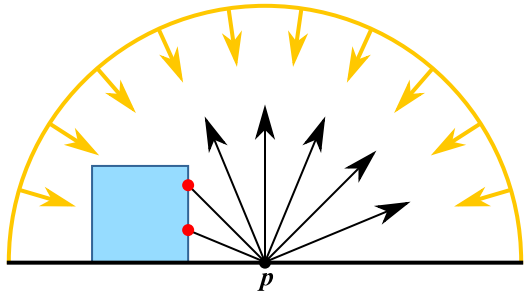


Fig. 2. Incident irradiance from the environmental light source (yellow) computed by ray tracing. A number of rays originated from $\boldsymbol{p}$ are generated over the hemisphere oriented according to the surface normal of $\boldsymbol{p}$. If a ray does not intersect the scene geometry (blue), the incident radiance coming from the environmental light source in the direction of the ray contributes to the incident irradiance at $\boldsymbol{p}$.

precomputed lighting, we can rewrite the incident irradiance at $\boldsymbol{p}$ as the following form

$$\begin{aligned} E(\boldsymbol{p}) &= \int_{\Omega^+} L_\mathrm{i}(\boldsymbol{p},\boldsymbol{\omega}_\mathrm{i})\{1-[1-V(\boldsymbol{p},\boldsymbol{\omega}_\mathrm{i})]\}\cos\theta_\mathrm{i}\mathrm{d}\boldsymbol{\omega}_\mathrm{i} \\ &= \int_{\Omega^+} L_\mathrm{i}(\boldsymbol{p},\boldsymbol{\omega}_\mathrm{i})\cos\theta_\mathrm{i}\mathrm{d}\boldsymbol{\omega}_\mathrm{i} - \int_{\Omega^+} L_\mathrm{i}(\boldsymbol{p},\boldsymbol{\omega}_\mathrm{i})[1-V(\boldsymbol{p},\boldsymbol{\omega}_\mathrm{i})]\cos\theta_\mathrm{i}\mathrm{d}\boldsymbol{\omega}_\mathrm{i} \\ &= E^{\mathrm{unshadowed}}(\boldsymbol{p}) - E^{\mathrm{occluded}}(\boldsymbol{p}), \end{aligned} \tag{3}$$

in which the incident irradiance $E(\boldsymbol{p})$ is split into two terms. The first term $E^{\mathrm{unshadowed}}(\boldsymbol{p})$ is the incident irradiance of precomputed/unshadowed environmental lighting, and the second term $E^{\mathrm{occluded}}(\boldsymbol{p})$ is the *occluded irradiance* representing the incident irradiance occluded by dynamic objects, which can be acquired by accumulating the cosine-weighted occluded incident radiances over the hemisphere. The irradiance occluded by dynamic objects is subtracted from the original precomputed/unshadowed incident irradiance to obtain the incident irradiance that contains the shadows cast by dynamic objects.

To compute the occluded irradiance, only the rays that can intersect dynamic objects are required to be traced. Therefore, the number of rays to be traced can be significantly reduced when the scene contains both static and dynamic objects. The rays that may intersect dynamic objects can be approximately determined by conical ray culling as described in the next subsection.

3.3 Conical Ray Culling

To efficiently find all the rays that may intersect dynamic objects, we can check the rays whether their directions are included by the circular cones towards all dynamic objects, which is called *direction culling* in this work. As depicted in Figure 3, the circular cone $C = (\boldsymbol{a}, \boldsymbol{d}, h, \psi)$ can be derived from the bounding sphere $S = (\boldsymbol{c}, r)$ of a dynamic object as follows

$$\begin{cases} \boldsymbol{a} = \boldsymbol{p} \\ \boldsymbol{d} = \frac{\boldsymbol{p}-\boldsymbol{c}}{\|\boldsymbol{p}-\boldsymbol{c}\|} \\ h = \|\boldsymbol{p}-\boldsymbol{c}\| + r \\ \psi = \arcsin(\frac{r}{\|\boldsymbol{p}-\boldsymbol{c}\|}) \end{cases}, \tag{4}$$

where $\boldsymbol{a}$ denotes the position of the apex of the cone, h is the height of the cone that is defined as the sum of the radius r of the bounding sphere and the distance between the surface point $\boldsymbol{p}$ and

the center of the bounding sphere $\boldsymbol{c}$, $\boldsymbol{d}$ is the direction of the axis of the cone, and ψ is the half angle of the cone.

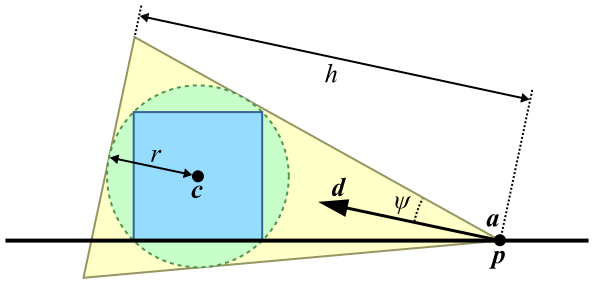


Fig. 3. Definition of the circular cone (yellow) by the bounding sphere (green) of a dynamic object (blue).

Apart from culling the rays whose directions are outside the cone, the length of the ray can be restricted to the height of the cone to accelerate ray tracing when static geometries are behind the dynamic objects from the perspective of the surface point, which is called *length culling*. It must be noted that if a ray is included by more than one cone, the length of the ray should be equal to the maximum height of all the cones.

As depicted in Figure 4, the rays whose directions are not inside any cone are culled by direction culling, and the ray segments beyond the cones are excluded by length culling. Therefore, the number of rays to be traced can be reduced to speed up the computation of soft shadows cast by dynamic objects. Besides, the static geometries behind the cones from the perspective of $\boldsymbol{p}$ can be neglected during ray tracing.

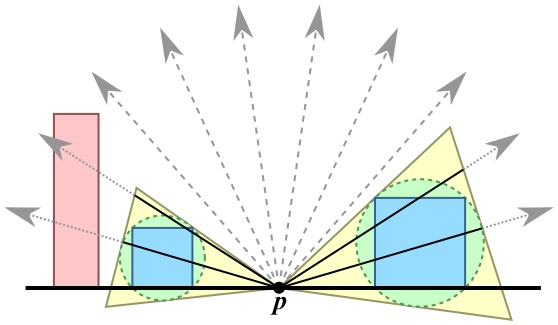


Fig. 4. Principle of conical ray culling. The rays whose directions are outside the cones (gray dashed) are culled by direction culling, which reduces the number of rays to be traced. The ray segments beyond the cones (gray dotted) are excluded by length culling, which prevents the rays from intersecting the static geometries (red) behind the cones from the perspective of $\boldsymbol{p}$.

Since the rays are distributed in the upper hemisphere along the surface normal $\boldsymbol{n}$, the bounding spheres that fully lie behind the surface should be ignored as depicted in Figure 5, which means the bounding sphere $S = (\boldsymbol{c}, r)$ should satisfy $(\boldsymbol{p} - \boldsymbol{c}) \cdot \boldsymbol{n} < r$. It should also be noted that no rays are culled when a scene point is inside one of the bounding spheres.

To further speed up ray tracing, the number of cones can be reduced by merging the cones for each surface point. Given two cones $C_1 = (\boldsymbol{a_1}, \boldsymbol{d_1}, h_1, \psi_1)$ and $C_2 = (\boldsymbol{a_2}, \boldsymbol{d_2}, h_2, \psi_2)$, these two cones

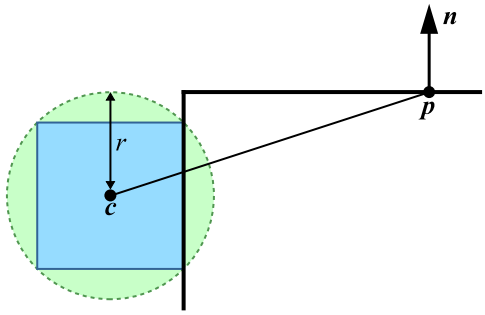


Fig. 5. Bounding sphere behind the surface should be ignored because the rays are distributed in the upper hemisphere along the surface normal $\boldsymbol{n}$.

can be merged if $\arccos(\boldsymbol{d_1} \cdot \boldsymbol{d_2}) + \psi_1 < \psi_2$ as illustrated in Figure 6. The merged cone $C = (\boldsymbol{a}, \boldsymbol{d}, h, \psi)$ can be obtained by

$$\begin{cases} \boldsymbol{a} = \boldsymbol{a}_1 = \boldsymbol{a}_2 \\ \boldsymbol{d} = \boldsymbol{d}_2 \\ h = \max\{h_1, h_2\} \\ \psi = \psi_2 \end{cases}, \tag{5}$$

which indicates that $(\boldsymbol{d}, \psi)$ of the merged cone are those whose angle is larger, and h of the merged cone is the larger height across these two cones.

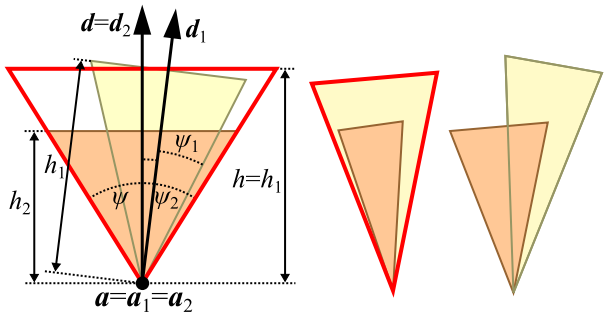


Fig. 6. Three cases of merging two cones. The two cones are marked in yellow and orange, respectively. The borders of the merged cones are marked in red. Left and middle: two cones that can be merged. Right: two cones that cannot be merged.

4 IMPLEMENTATION

We implement our proposed method using *Vulkan* with the extensions that enable hardware-accelerated ray tracing. The proposed method consists of four passes. The first pass generates the G-buffer which contains the positions, the normals, the diffuse albedos, and the incident irradiances of the precomputed environmental lighting for static objects or the unshadowed environmental lighting for dynamic objects; the second pass constructs and merges the cones for conical ray culling, and computes the irradiances occluded by dynamic objects using ray tracing; the third pass blurs the occluded irradiances by a cross-bilateral filter; and the last pass composites the irradiances of the unshadowed environmental lighting with the blurred occluded irradiances, multiplies the

composite irradiances by the diffuse albedos to obtain the outgoing radiances, and renders the environment map. Blurring the occluded irradiances instead of the composite irradiances prevents the noise-free unshadowed environmental lighting from blurring.

The environmental light source can be either a global environment map represented by a SH vector or an irradiance volume represented by 3D textures storing SH vectors in the grid cells. When irradiance volume is used, trilinear interpolation of the SH vectors is performed to acquire an interpolated SH vector as the environmental light source for each surface point of the scene. When the environment map is rotated, the irradiance volume is updated by reconstructing the irradiances by PRT at the sample points over the sphere and reprojecting the radiances of all sample points to the SH basis to obtain the irradiance for each grid cell.

For static objects, the incident irradiance without the shadows cast by dynamic objects at the surface point $\boldsymbol{p}$ is computed by diffuse PRT [Sloan et al. 2002] as follows

$$E^{\mathrm{PRT}}(\boldsymbol{p}) = \sum_{l=0}^{n-1} \sum_{m=-l}^{l} L_l^m t_l^m, \tag{6}$$

where $E(\boldsymbol{p})$ is the irradiance at surface point $\boldsymbol{p}$, L_l^m is the SH coefficient of the environmental light source, and t_l^m is the transfer coefficient. For dynamic objects, the unshadowed incident irradiance at the surface point $\boldsymbol{p}$ can be computed by the surface normal $\boldsymbol{n}$ and the SH coefficients of the environmental light source as described in [Ramamoorthi and Hanrahan 2001]

$$E^{\mathrm{unshadowed}}(\boldsymbol{p}) = \sum_{l=0}^{n-1} \sum_{m=-l}^{l} L_l^m \hat{A}_l Y_l^m(\boldsymbol{n}), \tag{7}$$

where $Y_l^m(\boldsymbol{n})$ is the SH basis function.

Vulkan ray tracing has two levels of acceleration structures: bottom-level acceleration structure (BLAS) and top-level acceleration structure (TLAS). We build the BLASes of all the models in the scene during the initialization stage and rebuild the BLASes of the animated models every frame. The TLAS is updated when the BLASes are built for the first time or the BLASes are rebuilt. Static objects are included in the acceleration structures because the static objects may lie between the origins of the rays and the dynamic objects and block the rays.

At the beginning of the second pass, the cones are constructed from the bounding spheres of dynamic objects, and then the cones are merged to reduce the number of cones. Algorithm 1 shows the pseudo-code to construct and merge the cones for each fragment in the fragment shader of the second pass. Once the cones are constructed and merged by Algorithm 1, ray tracing is performed in the fragment shader of the second pass. We trace the rays in the fragment shader using the Vulkan ray query extension VK_KHR_ray_query that allows ray tracing in the traditional shader stages. The hemispherical ray samples with cosine-weighted distribution are generated by the low-discrepancy Hammersley sequence [Niederreiter 1992]. The ray samples are rotated by random angles along the surface normal to avoid correlation artifacts. To retrieve the random angle of each fragment, a texture equal in size to the viewport is generated, which consists of random numbers in the range $[0, 2\pi]$. Each ray is checked to see if it is included by each cone of the current fragment. Ray tracing is not required to be performed if a ray is not included by any cone. If a ray is included by multiple cones, the ray length is set to the maximum height of all the cones.

When a ray intersects the scene geometry, the category of the hit point is determined by the rayQueryGetIntersectionInstanceCustomIndexEXT function. If a ray does not intersect the scene geometry or the hit point belongs to one of the static objects, the occluded incident radiance is zero. If the hit point belongs to dynamic objects, the occluded incident radiance that contributes to the occluded irradiance E^{occluded} is obtained by reconstructing the environment map in the direction of

Algorithm 1: Construct and merge the cones

Input: fragment position $\boldsymbol{p}$, fragment normal $\boldsymbol{n}$, bounding spheres of dynamic objects $\{S\}$
Output: merged cones $\{C\}$

foreach *bounding sphere* $S = (\boldsymbol{c}, r) \in \{S\}$ **do**
  $\boldsymbol{a} \leftarrow \boldsymbol{p}$; // Eq. (4)
  $\boldsymbol{d} \leftarrow \frac{\boldsymbol{p}-\boldsymbol{c}}{\|\boldsymbol{p}-\boldsymbol{c}\|}$; // Eq. (4)
  if $\|\boldsymbol{p}-\boldsymbol{c}\| < r$ **or** $(\boldsymbol{p}-\boldsymbol{c}) \cdot \boldsymbol{n} < r$ **then**
    $h \leftarrow \|\boldsymbol{p}-\boldsymbol{c}\| + r$; // Eq. (4)
    if $\|\boldsymbol{p}-\boldsymbol{c}\| < r$ **then**
      $\psi \leftarrow \frac{\pi}{2}$;
      $\boldsymbol{d} \leftarrow \boldsymbol{n}$;
    else
      $\psi \leftarrow \arcsin(\frac{r}{\|\boldsymbol{p}-\boldsymbol{c}\|})$; // Eq. (4)
    end
    inside = false;
    foreach *cone* $C_i = (\boldsymbol{a}_i, \boldsymbol{d}_i, h_i, \psi_i) \in \{C\}$ **do**
      if $\psi_i = \frac{\pi}{2}$ **or** $\arccos(\boldsymbol{d_1} \cdot \boldsymbol{d_2}) + \psi < \psi_i$ **then**
        inside $\leftarrow$ true;
        $h_i \leftarrow \max\{h_i, h\}$; // Eq. (5)
        break;
      end
      if $\psi = \frac{\pi}{2}$ **or** $\arccos(\boldsymbol{d_1} \cdot \boldsymbol{d_2}) + \psi_i < \psi$ **then**
        inside $\leftarrow$ true;
        $\boldsymbol{d}_i \leftarrow \boldsymbol{d}$; // Eq. (5)
        $h_i \leftarrow \max\{h_i, h\}$; // Eq. (5)
        $\psi_i \leftarrow \psi$; // Eq. (5)
        break;
      end
    end
    if *inside* = *false* **then**
      add $C = (\boldsymbol{a}, \boldsymbol{d}, h, \psi)$ to $\{C\}$;
    end
  end
end

the ray $\boldsymbol{\omega}$ by the SH coefficients as follows

$$L_{\mathrm{i}}^{\mathrm{occluded}}(\boldsymbol{\omega}) = \sum_{l=0}^{n-1} \sum_{m=-l}^{l} L_l^m Y_l^m(\boldsymbol{\omega}). \tag{8}$$

All the occluded incident radiances of the hemispherical ray samples are accumulated and then multiplied by π and divided by the number of ray samples to obtain the occluded irradiance. The pseudo-code of conical ray culling and ray tracing for each fragment in the fragment shader of the second pass is exhibited in Algorithm 2.

Algorithm 2: Conical ray culling and ray tracing

Input: fragment position $\boldsymbol{p}$, fragment normal $\boldsymbol{n}$, N hemispherical ray samples $\{R\}$, cones $\{C\}$

Output: occluded irradiance $E^{\text{occluded}}(\boldsymbol{p})$

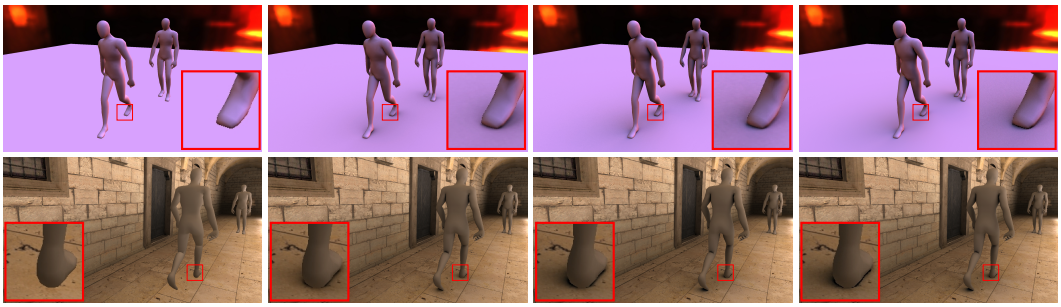
$E^{\text{occluded}}(\boldsymbol{p}) \leftarrow 0$;
foreach *ray* $R = (\boldsymbol{o}, \boldsymbol{\omega}) \in \{R\}$ **do**
  rotate R along $\boldsymbol{n}$ by a random angle;
  $inside \leftarrow$ false;
  $maxRayLength \leftarrow 0$;
  foreach *cone* $C = (\boldsymbol{a}, \boldsymbol{d}, h, \psi) \in \{C\}$ **do**
    if $\arccos(\boldsymbol{\omega} \cdot \boldsymbol{d}) < \psi$ **then**
      $inside \leftarrow$ true;
      $maxRayLength \leftarrow \max\{maxRayLength, h\}$;
    end
  end
  if $inside = true$ **then**
    trace R;
    if *R intersects dynamic objects* **then**
      $E^{\text{occluded}}(\boldsymbol{p}) \leftarrow E^{\text{occluded}}(\boldsymbol{p}) + \frac{\pi}{N} L_{\text{i}}^{\text{occluded}}(\boldsymbol{\omega})$;
    end
  end
end

5 RESULTS

We test our proposed method in two scenes. The first scene consists of a plane (2 triangles), which is regarded as the static object, and two animated characters (6.4 k triangles in total) on the plane as the dynamic objects, which are obtained from the resources of the SHEXP paper [Ren et al. 2006]. The whole scene is illuminated by a global Grace Cathedral environment map represented by a 3-band SH vector (9 coefficients for each color channel). The second scene contains the Crytek Sponza Atrium model (81.7 k triangles) downloaded from Morgan McGuire's Computer Graphics Archive [McGuire 2017] as the static object and the same two animated characters on the ground of the Crytek Sponza Atrium as in the first scene. The Crytek Sponza Atrium is illuminated by 3-band SH diffuse PRT with interreflections under the Uffizi Gallery environment map. The radiance transfer vectors of PRT are stored in a 2D texture array with a resolution of 2048×2048. The curtains and the small objects in the Crytek Sponza Atrium model are manually removed to allow the corridor to be illuminated by the environment map and simplify the PRT precomputation procedure. The dynamic objects are illuminated by an irradiance volume that stores 3-band SH vectors, which are represented by multiple 3D textures with a resolution of 53×29×23. Each texel of a 3D texture stores three SH coefficients, which represent the RGB color channels. The number of ray samples is set to 128 and the kernel size of the cross-bilateral filter is set to 5×5. All results are rendered at a resolution of 1920×1080 on a computer equipped with an Intel Core i9-10980XE CPU, 128 GB RAM, and an NVIDIA GeForce RTX 3080 GPU. All timings are averaged over 100 frames.

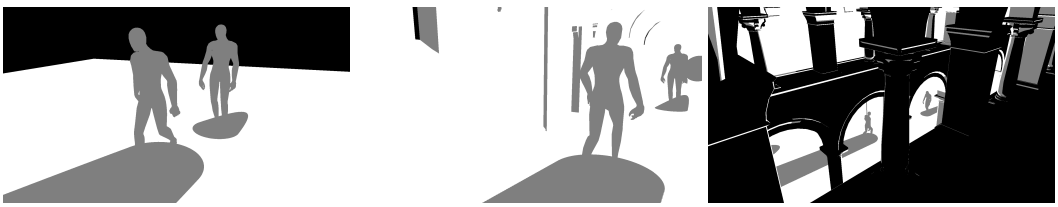
As displayed in Figure 7, we compare our proposed method (Figure 7c) to unshadowed environmental lighting (Figure 7a), environmental lighting with RTAO (Figure 7b), and unfiltered ray-traced shadowed environmental lighting using 512 ray samples per pixel as the reference (Figure 7d).

Compared to unshadowed environmental lighting, the proposed method can significantly enhance the realism by providing visual cues about the relative positions of objects, so the characters do not seem to float in the air. Compared to environmental lighting with RTAO, the proposed method can capture the directionality of the environment map. The rendering results of the proposed method are almost the same as those of unfiltered 512-spp ray-traced environmental lighting, except for some insignificant overblurring caused by the cross-bilateral filter around the corners, such as the surroundings of the feet. The number of cones after merging is visualized in Figure 8, in which black denotes no cone, gray denotes one cone, and white denotes two cones.



(a) Unshadowed (b) RTAO (c) Proposed (d) Reference

Fig. 7. Comparison of the rendering results using different methods. Regions in the red squares are zoomed in. (a) Unshadowed environmental lighting. (b) Environmental lighting with 128-spp RTAO. (c) Our proposed method (128-spp ray-traced shadowed environmental lighting). (d) Reference (unfiltered 512-spp ray-traced shadowed environmental lighting). Top row: scene 1 (two characters on a plane illuminated by a global Grace Cathedral environment map, 6.4 k triangles in total). Bottom row: scene 2 (two characters and the Crytek Sponza Atrium model under the Uffizi Gallery environment map, 88.1 k triangles in total).



(a) Scene 1 (b) Scene 2 (c) Another perspective of scene 2

Fig. 8. Visualization of the number of cones after merging (black denotes no cone, gray denotes one cone, and white denotes two cones). (a) Scene 1. (b) Scene 2. (c) Another perspective of scene 2.

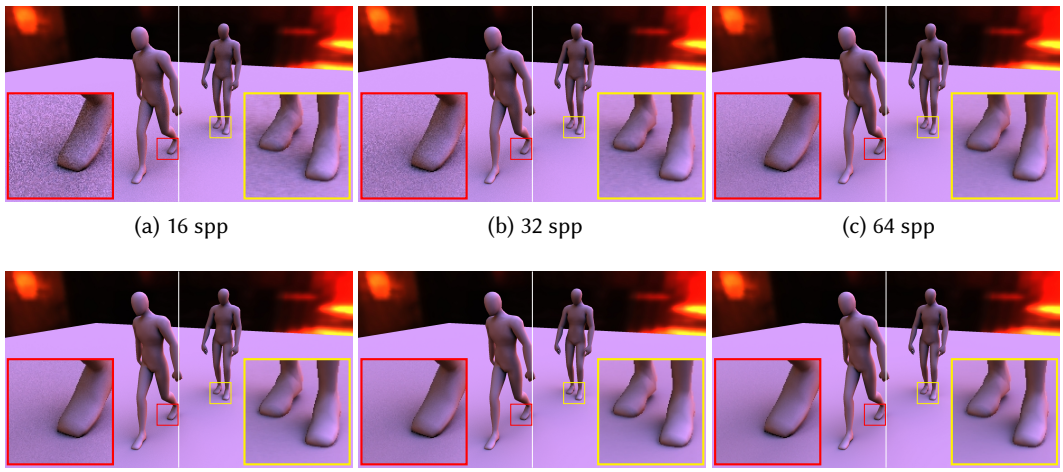
The timings of individual passes w/o and w/ our proposed conical ray culling are displayed in Table 1, which demonstrates that the ray tracing pass spends most of the total frame time. In the first scene, direction culling can reduce the cost of the ray tracing pass from 12.89 ms to 8.23 ms, and length culling can slightly reduce the cost from 8.23 ms to 7.79 ms. The total speedup of the first scene is 1.65×. In the second scene, the same cost can be reduced by direction culling from 49.29 ms to 21.01 ms, and further reduced by length culling to 15.75 ms. The total speedup of the second scene is 3.13×, which is much higher than that of the first scene because the Crytek Sponza Atrium

model is more complex than the plane (81.7 k triangles versus 2 triangles). The statistics reveal that 60 fps cannot be achieved without conical ray culling in the second scene, which demonstrates the necessity of conical ray culling.

Table 1. Timings (in ms) of individual rendering passes w/o and w/ conical ray culling.

		G-Buffer	Ray tracing	Blurring	Composition	Total
Scene 1	w/o conical ray culling	0.10	12.89	0.14	0.05	13.19
	w/ direction culling		8.23			8.53
	w/ direction & length culling		7.79			8.09
Scene 2	w/o conical ray culling	0.25	49.29	0.18	0.06	49.79
	w/ direction culling		21.01			21.51
	w/ direction & length culling		15.75			16.25

We also compare the results of the first scene using 16, 32, 64, 128, 256, and 512 ray samples per pixel (spp), respectively. The rendering results are shown in Figure 9, and the timings of the ray tracing pass are exhibited in Table 2. It can be seen that the noise is unnoticeable when spp reaches 128. Therefore, we set 128 as the default number of ray samples in the previous results.



(a) 16 spp (b) 32 spp (c) 64 spp

(d) 128 spp (e) 256 spp (f) 512 spp

Fig. 9. Comparison of the rendering results using different numbers of ray samples per pixel. Regions in the red and yellow squares are zoomed in. The left half of each image is unfiltered, and the right half is filtered. (a) 16 spp. (b) 32 spp. (c) 64 spp. (d) 128 spp. (e) 256 spp. (f) 512 spp.

Table 2. Timings (in ms) of the ray tracing pass using different samples per pixel.

spp	16	32	64	128	256	512
Timings	0.99	1.95	3.89	7.79	15.70	35.52

6 LIMITATIONS

While conical ray culling can speed up ray tracing, the speedup ratio decreases when the number of dynamic objects is large and the cones cannot be substantially merged because each ray has to be tested against all the cones to check whether it should be traced. For example, ten characters are placed in the two scenes shown in the results section, as shown in Figure 10. Conical ray culling can only slightly reduce the cost of the ray tracing pass from 16.08 ms to 15.85 ms in the first scene. And in the second scene, the ray tracing pass is sped up by conical ray culling from 55.41 ms to 27.31 ms. The speedup ratio is only 2×, which is not as high as that in the same scene with only two characters. The cones can be clustered together to reduce the number of cones, which may raise another issue that the cones constructed by distant objects that have small apex angles contribute little to the occlusion but affect the clustering process. Therefore, cones should be culled according to the sample density before clustering. Another potential performance issue is that the parallelism may decrease when the number of cones is large because the cones are constructed for each fragment and stored in the registers of the GPU. In addition, performance may dramatically drop when dynamic objects occupy the viewport because all rays are required to be traced if a scene point is inside one of the bounding spheres of dynamic objects. Multiresolution techniques can be introduced to speed up in this situation.

Fig. 10. Speedup of the ray tracing pass by conical ray culling decreases when the number of dynamic objects is large. Left: 16.08 ms to 15.85 ms. Right: 55.41 ms to 27.31 ms.

Apart from the performance issues, we assume low-frequency environmental lighting and diffuse surfaces in this work. Besides, indirect illumination is not considered, and only environmental light sources are supported in this work. We will take indirect illumination into account and add support for sun light and other local lights such as point lights and spot lights in the future.

7 CONCLUSION

In this work, we present a method to render soft shadows of environmental lighting in real-time based on hardware-accelerated ray tracing. We assume that the scene includes both static and dynamic objects. To composite the soft shadows cast by dynamic objects with the precomputed/unshadowed lighting of static objects, the incident irradiance occluded by dynamic objects, which represents the soft shadows of environmental lighting, is subtracted from the precomputed/unshadowed incident irradiance to obtain the composite incident irradiance. Conical ray culling is proposed based on the observation that only the rays that can intersect dynamic objects are required to be traced to evaluate the occluded irradiance. The rays whose directions are outside the cones defined by the surface point and the bounding spheres of dynamic objects are excluded, and the ray segments beyond the cones are excluded as well. Therefore, fewer and shorter rays

are required to be traced, which significantly improves the efficiency of rendering soft shadows of environmental lighting cast by dynamic objects.

ACKNOWLEDGMENTS

We thank Marko Dabrovic, Frank Meinl, and Morgan McGuire for the Crytek Sponza Atrium model used in our test [McGuire 2017]. This work was supported by the National Key R&D Program of China (2019YFC1521102), the Key Research and Development Program of Shaanxi (2019GY-215), the Major Research and Development Project of Qinghai (2020-SF-143), and the Scientific Research Program Funded by Shaanxi Provincial Education Department (21JK0931).

REFERENCES

Louis Bavoil and Miguel Sainz. 2009. Multi-Layer Dual-Resolution Screen-Space Ambient Occlusion. In *ACM SIGGRAPH 2009 Talks* (New Orleans, Louisiana) *(SIGGRAPH '09)*. Association for Computing Machinery, New York, NY, USA, Article 45, 1 pages. https://doi.org/10.1145/1597990.1598035

Louis Bavoil, Miguel Sainz, and Rouslan Dimitrov. 2008. Image-Space Horizon-Based Ambient Occlusion. In *ACM SIGGRAPH 2008 Talks* (Los Angeles, California) *(SIGGRAPH '08)*. Association for Computing Machinery, New York, NY, USA, Article 22, 1 pages. https://doi.org/10.1145/1401032.1401061

Michael Bunnell. 2005. Dynamic Ambient Occlusion and Indirect Lighting. In *GPU Gems 2*. Addison-Wesley, 223–233.

Chakravarty R. Alla Chaitanya, Anton S. Kaplanyan, Christoph Schied, Marco Salvi, Aaron Lefohn, Derek Nowrouzezahrai, and Timo Aila. 2017. Interactive Reconstruction of Monte Carlo Image Sequences Using a Recurrent Denoising Autoencoder. *ACM Trans. Graph.* 36, 4, Article 98 (July 2017), 12 pages. https://doi.org/10.1145/3072959.3073601

Aude Giraud and Derek Nowrouzezahrai. 2015. Practical Shading of Height Fields and Meshes using Spherical Harmonic Exponentiation. In *Eurographics Symposium on Rendering - Experimental Ideas & Implementations*, Jaakko Lehtinen and Derek Nowrouzezahrai (Eds.). The Eurographics Association, 1–8. https://doi.org/10.2312/sre.20151161

Paul Guerrero, Stefan Jeschke, and Michael Wimmer. 2008. Real-Time Indirect Illumination and Soft Shadows in Dynamic Scenes Using Spherical Lights. *Comput. Graph. Forum* 27, 8 (2008), 2154–2168. https://doi.org/10.1111/j.1467-8659.2008.01296.x

Jon Hasselgren, Jacob Munkberg, Marco Salvi, Anjul Patney, and Aaron Lefohn. 2020. Neural Temporal Adaptive Sampling and Denoising. *Comput. Graph. Forum* 39, 2 (2020), 147–155. https://doi.org/10.1111/cgf.13919

Jared Hoberock and Yuntao Jia. 2007. High-Quality Ambient Occlusion. In *GPU Gems 3*. 257–274.

Michal Iwanicki. 2013. Lighting Technology of the Last of Us. In *ACM SIGGRAPH 2013 Talks* (Anaheim, California) *(SIGGRAPH '13)*. Association for Computing Machinery, New York, NY, USA, Article 20, 1 pages. https://doi.org/10.1145/2504459.2504484

Jan Kautz, Jaakko Lehtinen, and Timo Aila. 2004. Hemispherical Rasterization for Self-Shadowing of Dynamic Objects. In *Eurographics Workshop on Rendering*, Alexander Keller and Henrik Wann Jensen (Eds.). The Eurographics Association. https://doi.org/10.2312/EGWR/EGSR04/179-184

Benjamin Keinert, Jana Martschinke, and Marc Stamminger. 2018. Learning Real-Time Ambient Occlusion from Distance Representations. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (Montreal, Quebec, Canada) *(I3D '18)*. Association for Computing Machinery, New York, NY, USA, Article 3, 9 pages. https://doi.org/10.1145/3190834.3190847

Adam G. Kirk and Okan Arikan. 2007. Real-Time Ambient Occlusion for Dynamic Character Skins. In *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games* (Seattle, Washington) *(I3D '07)*. Association for Computing Machinery, New York, NY, USA, 47–52. https://doi.org/10.1145/1230100.1230108

Janne Kontkanen and Timo Aila. 2006. Ambient Occlusion for Animated Characters. In *Proceedings of the 17th Eurographics Conference on Rendering Techniques* (Nicosia, Cyprus) *(EGSR '06)*. The Eurographics Association, Goslar, DEU, 343–348. https://doi.org/10.2312/EGWR/EGSR06/343-348

Janne Kontkanen and Samuli Laine. 2005. Ambient Occlusion Fields. In *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games* (Washington, District of Columbia) *(I3D '05)*. Association for Computing Machinery, New York, NY, USA, 41–48. https://doi.org/10.1145/1053427.1053434

Binh Huy Le, Henrik Halen, Carlos Gonzalez-Ochoa, and JP Lewis. 2019. High-Quality Object-Space Dynamic Ambient Occlusion for Characters Using Bi-Level Regression. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (Montreal, Quebec, Canada) *(I3D '19)*. Association for Computing Machinery, New York, NY, USA, Article 6, 10 pages. https://doi.org/10.1145/3306131.3317029

Yue Li, Pablo Wiedemann, and Kenny Mitchell. 2019. Deep Precomputed Radiance Transfer for Deformable Objects. *Proc. ACM Comput. Graph. Interact. Tech.* 2, 1, Article 3 (June 2019), 16 pages. https://doi.org/10.1145/3320284

Michael Mara, Morgan McGuire, Derek Nowrouzezahrai, and David Luebke. 2016. Deep G-Buffers for Stable Global Illumination Approximation. In *Proceedings of High Performance Graphics* (Dublin, Ireland) *(HPG '16)*. The Eurographics Association, Goslar, DEU, 87–98. https://doi.org/10.2312/hpg.20161195

Morgan McGuire. 2017. *Computer Graphics Archive.* https://casual-effects.com/data

Martin Mittring. 2007. Finding Next Gen: CryEngine 2. In *ACM SIGGRAPH 2007 Courses* (San Diego, California) *(SIGGRAPH '07)*. Association for Computing Machinery, New York, NY, USA, 97–121. https://doi.org/10.1145/1281500.1281671

Harald Niederreiter. 1992. *Random Number Generation and Quasi-Monte Carlo Methods.* Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9781611970081

Derek Nowrouzezahrai, Patricio Simari, Evangelos Kalogerakis, Karan Singh, and Eugene Fiume. 2007. Compact and Efficient Generation of Radiance Transfer for Dynamically Articulated Characters. In *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia* (Perth, Australia) *(GRAPHITE '07)*. Association for Computing Machinery, New York, NY, USA, 147–154. https://doi.org/10.1145/1321261.1321288

Ravi Ramamoorthi and Pat Hanrahan. 2001. An Efficient Representation for Irradiance Environment Maps. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. Association for Computing Machinery, New York, NY, USA, 497–500. https://doi.org/10.1145/383259.383317

Zhong Ren, Rui Wang, John Snyder, Kun Zhou, Xinguo Liu, Bo Sun, Peter-Pike Sloan, Hujun Bao, Qunsheng Peng, and Baining Guo. 2006. Real-Time Soft Shadows in Dynamic Scenes Using Spherical Harmonic Exponentiation. *ACM Trans. Graph.* 25, 3 (July 2006), 977–986. https://doi.org/10.1145/1141911.1141982

Tobias Ritschel, Thorsten Grosch, and Hans-Peter Seidel. 2009. Approximating Dynamic Global Illumination in Image Space. In *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games* (Boston, Massachusetts) *(I3D '09)*. Association for Computing Machinery, New York, NY, USA, 75–82. https://doi.org/10.1145/1507149.1507161

Christoph Schied, Anton Kaplanyan, Chris Wyman, Anjul Patney, Chakravarty R. Alla Chaitanya, John Burgess, Shiqiu Liu, Carsten Dachsbacher, Aaron Lefohn, and Marco Salvi. 2017. Spatiotemporal Variance-Guided Filtering: Real-Time Reconstruction for Path-Traced Global Illumination. In *Proceedings of High Performance Graphics* (Los Angeles, California) *(HPG '17)*. Association for Computing Machinery, New York, NY, USA, Article 2, 12 pages. https://doi.org/10.1145/3105762.3105770

Perumaal Shanmugam and Okan Arikan. 2007. Hardware Accelerated Ambient Occlusion Techniques on GPUs. In *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games* (Seattle, Washington) *(I3D '07)*. Association for Computing Machinery, New York, NY, USA, 73–80. https://doi.org/10.1145/1230100.1230113

Peter-Pike Sloan, Naga K. Govindaraju, Derek Nowrouzezahrai, and John Snyder. 2007. Image-Based Proxy Accumulation for Real-Time Soft Global Illumination. In *15th Pacific Conference on Computer Graphics and Applications (PG'07)*. IEEE Computer Society, Los Alamitos, CA, USA, 97–105. https://doi.org/10.1109/PG.2007.28

Peter-Pike Sloan, Jan Kautz, and John Snyder. 2002. Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques* (San Antonio, Texas) *(SIGGRAPH '02)*. Association for Computing Machinery, New York, NY, USA, 527–536. https://doi.org/10.1145/566570.566612

Kostas Vardis, Georgios Papaioannou, and Athanasios Gaitatzes. 2013. Multi-View Ambient Occlusion with Importance Sampling. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (Orlando, Florida) *(I3D '13)*. Association for Computing Machinery, New York, NY, USA, 111–118. https://doi.org/10.1145/2448196.2448214

Jop Vermeer, Leonardo Scandolo, and Elmar Eisemann. 2021. Stochastic-Depth Ambient Occlusion. *Proc. ACM Comput. Graph. Interact. Tech.* 4, 1, Article 3 (April 2021), 15 pages. https://doi.org/10.1145/3451268

Hanggao Xin, Zhiqian Zhou, Di An, Ling-Qi Yan, Kun Xu, Shi-Min Hu, and Shing-Tung Yau. 2021. Fast and Accurate Spherical Harmonics Products. *ACM Trans. Graph.* 40, 6, Article 280 (Dec. 2021), 14 pages. https://doi.org/10.1145/3478513.3480563

Kun Zhou, Yaohua Hu, Stephen Lin, Baining Guo, and Heung-Yeung Shum. 2005. Precomputed Shadow Fields for Dynamic Scenes. *ACM Trans. Graph.* 24, 3 (July 2005), 1196–1201. https://doi.org/10.1145/1073204.1073332