
Irradiance Gradients in the Presence of Participating Media and Occlusions

“Write a wise saying and your name will live forever.”

—Unknown

THE irradiance caching [Ward et al., 1988] algorithm described in Chapter 3 is one of the most successful acceleration strategies for solving the rendering equation using Monte Carlo ray tracing. In the previous chapter we developed a complementary radiance caching algorithm to efficiently solve the *volume* rendering equation *within* participating media by accounting for surface-volume and volume-volume interactions of light. These two approaches can easily be coupled to simulate all light transport by caching both on surfaces and within the medium. However, irradiance caching and volumetric radiance caching both rely on accurate gradient computations to improve the accuracy of interpolation. Participating media, unfortunately, affects not only the irradiance but also the irradiance gradient on surfaces embedded within media.

6.1 Contributions

In Table 6.1 we summarize the most important previous techniques for computing illumination gradients for irradiance caching. Unfortunately, none of these methods take into account the full radiative transport equation, which leads to increased interpolation artifacts in the presence of participating media. In this chapter we present extensions to these techniques to reduce these problems. We develop novel gradients that consider absorption, emission, and scattering in volumetric media, and the effect of surfaces occluding media. In particular, we

present the following contributions:

1. We derive gradients to handle indirect illumination from surfaces in the presence of **absorbing** media and occlusions. This generalization also allows us to handle non-Lambertian reflectors.
2. We derive gradients of irradiance in the presence of **scattering** or **emissive** media and account for surface-medium occlusion changes.

These two contributions enable, for the first time, the accurate computation of illumination gradients in the context of the full radiative transport equation. Without tracing any additional rays, the gradients described in this chapter are more accurate and contain significantly less noise than gradients computed using expensive numerical techniques such as finite differencing. The resulting gradient computations are easy to implement and straightforward to incorporate into an irradiance caching framework. We show that the new gradients produce higher-quality interpolation than previous techniques, which do not take into account full radiative transport.

Table 6.1: A comparison of the capabilities of illumination gradient techniques. The gradients derived in this chapter take into account the full radiative transport equation, including the effects of absorbing, emissive, and scattering participating media (PM). We support cache points on glossy surfaces (GCP), and we consider effects of glossy indirect reflectors (GIR), visibility changes (V), indirect illumination (II), and curved objects (CO).

Method	PM	GCP	GIR	V	II	CO
Ward and Heckbert [1992]				✓	✓	✓
Arvo [1994]				✓	✓	
Křivánek et al. [2005b]		✓			✓	✓
Křivánek et al. [2005a]		✓		✓	✓	✓
Ramamoorthi et al. [2007]		✓	✓	✓		✓
Chapter 5	✓	✓	✓		✓	✓
This Chapter	✓	✓	✓	✓	✓	✓

6.2 Overview

As we saw in Chapter 2, irradiance E at a surface location \mathbf{x} can be written as the cosine-weighted integral of incident radiance,

$$E(\mathbf{x}) = \int_{\Omega} L(\mathbf{x} \leftarrow \vec{\omega}) (\hat{\mathbf{n}} \cdot \vec{\omega}) d\vec{\omega}. \quad (6.1)$$

We are interested in computing the gradient $\nabla E(\mathbf{x})$ with respect to a translation of \mathbf{x} .

In the presence of participating media, light transport obeys the *radiative transfer equation* from Equation 4.27. In this case we can interpret the incident radiance $L(\mathbf{x} \leftarrow \vec{\omega})$ as the sum of two separate contributions: the radiance L_s coming from surfaces that reflect illumination, and the radiance L_m contributed by the participating medium, i.e.,

$$L(\mathbf{x} \leftarrow \vec{\omega}) = \underbrace{T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s) L(\mathbf{x}_s \rightarrow -\vec{\omega})}_{L_s(\mathbf{x} \leftarrow \vec{\omega})} + \underbrace{\int_0^s T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t) \sigma_s(\mathbf{x}_t) L_i(\mathbf{x}_t \rightarrow -\vec{\omega}) dt}_{L_m(\mathbf{x} \leftarrow \vec{\omega})}, \quad (6.2)$$

where $\mathbf{x}_t = \mathbf{x} + t\vec{\omega}$ with $t \in (0, s)$, and s is the distance through the medium to the nearest surface at $\mathbf{x}_s = \mathbf{x} + s\vec{\omega}$. The outgoing radiance $L(\mathbf{x}_s \rightarrow -\vec{\omega})$ is computed by integrating the lighting and BRDF at \mathbf{x}_s , and $T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s)$ is the transmittance of the medium between \mathbf{x} and \mathbf{x}_s .

To improve the clarity of our derivations, we split up the total irradiance in Equation 6.1 into irradiance from surfaces E_s and irradiance from the medium E_m such that

$$E_s(\mathbf{x}) = \int_{\Omega} L_s(\mathbf{x} \leftarrow \vec{\omega}) (\hat{\mathbf{n}} \cdot \vec{\omega}) d\vec{\omega}, \quad (6.3)$$

$$E_m(\mathbf{x}) = \int_{\Omega} L_m(\mathbf{x} \leftarrow \vec{\omega}) (\hat{\mathbf{n}} \cdot \vec{\omega}) d\vec{\omega}, \quad (6.4)$$

and $E(\mathbf{x}) = E_s(\mathbf{x}) + E_m(\mathbf{x})$. The total gradient is simply $\nabla E = \nabla E_s + \nabla E_m$. We derive gradients for the irradiance from surfaces in Section 6.3 and for irradiance from participating media in Section 6.4. We illustrate this process in Figure 6.1.

Note that in the following sections we only consider irradiance gradients for brevity, but all of these computations could easily be projected onto spherical harmonics to obtain full radiance gradients, as done in the previous chapter. Also, for clarity, we do not include media emission in

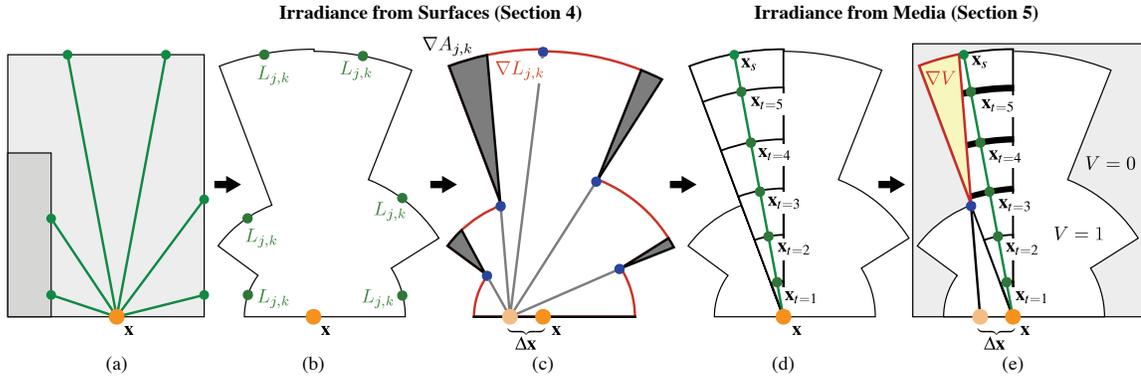


Figure 6.1: Irradiance caching approximates irradiance by tracing rays in a hemisphere (a) and calculating the radiance and distance to each hitpoint (b). Ward and Heckbert express irradiance gradients (c) by computing $\nabla A_{j,k}$ (grey) of each cell under a translation of \mathbf{x} , where the rate of motion of a cell boundary is determined by the distance (blue) to the closer surface. Our method accounts for absorbing media by additionally considering $\nabla L_{j,k}$ (red). We use ray marching (d) by sampling the in-scattered radiance at discrete points \mathbf{x}_t . For each discrete “shell” we compute both a radiance gradient and a visibility gradient (e) to account for surfaces occluding the medium. Conceptually, ∇V reduces the contribution of radiance from shells (highlighted) beyond the occluder (blue).

our derivations. This can be trivially included since the emission and in-scattered terms are nearly identical. We also restrict our derivations to translational gradients. We account for gradients on curved surfaces by including a rotational gradient computed as described by Ward and Heckbert [1992]. Unlike the translational gradient, participating media does not influence the computation of the rotational gradient.

6.3 Irradiance Gradients for Surfaces

In this section we present a generalization of Ward and Heckbert [1992] and Křivánek et al. [2005a] irradiance gradients. Our generalization relaxes the restriction that radiance arriving at \mathbf{x} from an indirect light source at \mathbf{x}' is constant under translation of \mathbf{x} . This generalization allows us to correctly handle scenes where the radiance may change due either to absorption by participating media or glossy reflectors.

6.3.1 Irradiance Gradients

In the irradiance gradient computations of Ward and Heckbert [1992] and Křivánek et al. [2005a], the irradiance integral from Equation 6.3 is estimated using the stratified Monte Carlo

estimator introduced in Chapter 3,

$$E_s(\mathbf{x}) \approx \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} A_{j,k} L_s(\mathbf{x} \leftarrow \vec{\omega}_{j,k}) (\vec{\mathbf{n}} \cdot \vec{\omega}_{j,k}), \quad (6.5)$$

where we defined the relevant quantities in Table 3.1 and illustrated the stratified geometry in Figure 3.4.

The contribution of each sample is the product of the radiance through the cell, $L_s(\mathbf{x} \leftarrow \vec{\omega}_{j,k})$, and the area of the cell, $A_{j,k}$, and this contribution is weighted by the cosine term $(\vec{\mathbf{n}} \cdot \vec{\omega}_{j,k})$. The irradiance gradient considers how these terms change when translating the center of projection by an infinitesimal amount. By differentiating Equation 6.5 we get:

$$\begin{aligned} \nabla E_s(\mathbf{x}) &\approx \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} \nabla (A_{j,k} L_s(\mathbf{x} \leftarrow \vec{\omega}_{j,k}) (\vec{\mathbf{n}} \cdot \vec{\omega}_{j,k})) \\ &= \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} (\nabla A_{j,k} L_{j,k}^s + A_{j,k} \nabla L_{j,k}^s) (\vec{\mathbf{n}} \cdot \vec{\omega}_{j,k}), \end{aligned} \quad (6.6)$$

where we have used $L_{j,k}^s = L_s(\mathbf{x} \leftarrow \vec{\omega}_{j,k})$ for brevity. Translation in the tangent plane may induce both a change in the area of a cell $\nabla A_{j,k}$ and a change in the radiance seen through a cell $\nabla L_{j,k}^s$. Note that the cosine term does not change under translation, so the $\nabla(\vec{\mathbf{n}} \cdot \vec{\omega}_{j,k})$ term drops out.

The total surface irradiance gradient can be expressed as a sum of two terms,

$$\nabla E_s(\mathbf{x}) \approx \nabla_A E_s(\mathbf{x}) + \nabla_L E_s(\mathbf{x}), \quad (6.7)$$

where $\nabla_A E_s(\mathbf{x})$ includes the gradient terms containing changing cell area $\nabla A_{j,k}$, and $\nabla_L E_s(\mathbf{x})$ incorporates the terms with changing cell radiance $\nabla L_{j,k}^s$.

Previous work on irradiance and radiance gradients only considers the rate of change of each cell area, $\nabla A_{j,k}$, but assumes that $\nabla L_{j,k}^s = 0$. In Section 3.6.2 we provided detailed derivations of the cell area gradient contribution. In this chapter we consider the effects of $\nabla_L E_s(\mathbf{x})$, for which we derive expressions in the following section.

6.3.2 Gradient of Cell Radiance

By allowing the cell radiances to change with translation, we can include attenuation due to participating media as well as glossy reflectors in the gradients. We additionally compute a gradient $\nabla_L E_s$ that incorporates changes in cell radiance,

$$\nabla_L E_s(\mathbf{x}) \approx \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} A_{j,k} \nabla L_{j,k}^s(\mathbf{n} \cdot \vec{\omega}_{j,k}), \quad (6.8)$$

where the area of each cell is

$$A_{j,k} = \int_{\phi_{k_-}}^{\phi_{k_+}} \int_{\theta_{j_-}}^{\theta_{j_+}} \sin \theta d\theta d\phi \implies (\cos \theta_{j_-} - \cos \theta_{j_+})(\phi_{k_+} - \phi_{k_-}). \quad (6.9)$$

For a cosine-weighted distribution, this gradient term reduces to the average of the cell gradients,

$$\nabla_L E_s(\mathbf{x}) \approx \frac{\pi}{MN} \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} \nabla L_{j,k}^s. \quad (6.10)$$

Using the definition of surface radiance from Equation 6.2, the gradient $\nabla L_{j,k}^s$ is

$$\nabla L_{j,k}^s = \nabla T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s) L(\mathbf{x}_s, -\vec{\omega}_{j,k}) + T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s) \nabla L(\mathbf{x}_s, -\vec{\omega}_{j,k}). \quad (6.11)$$

We compute the gradient of the transmittance term using the techniques described in the previous chapter. Though we do not demonstrate this in our results, glossy reflectors can easily be handled in our framework by computing the radiance gradient using the gradient of the BRDF at \mathbf{x}_s .

The final irradiance gradient is simply the sum of Ward and Heckbert's original gradient $\nabla_A E_s(\mathbf{x})$ and our additional cell radiance gradient term $\nabla_L E_s(\mathbf{x})$ as expressed in Equation 6.7. In Figure 6.2 we demonstrate the impact of adding the gradients of cell radiance. The resulting irradiance gradients are more accurate, and the quality of interpolation in irradiance caching is significantly improved.

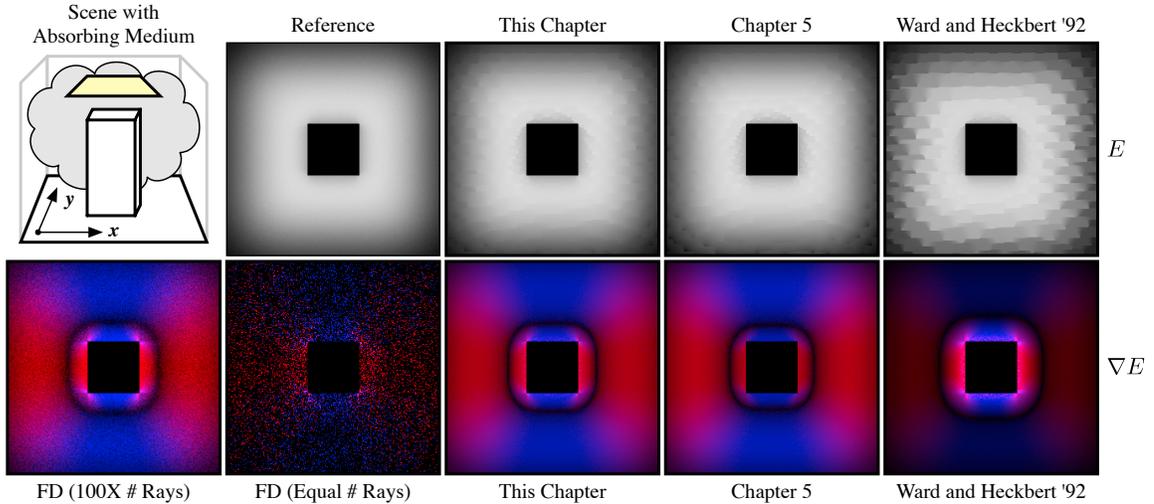


Figure 6.2: We compare irradiance caching and extrapolation using different gradient computation techniques in a scene with an **absorbing** medium. The top row shows irradiance *extrapolated* over the ground plane from a few cache points. The improved gradients from this chapter consider changes of visibility and absorption by the medium. Our method from Chapter 5 ignores visibility and Ward and Heckbert [1992] ignore absorption by the medium, which leads to increased artifacts. In the bottom row we visualize the irradiance gradients computed *per-pixel*. The red and blue color coding shows absolute values of the x and y components of the gradient, respectively. We computed gradients for each technique using 1.8K gather rays per pixel and also a “ground truth” gradient using finite differences (FD). Since finite difference gradients are extremely noisy, our reference uses 100 times more rays (180K rays/pixel). Our method produces accurate gradients with few gather rays and converges to the reference solution using finite differencing.

6.4 Irradiance Gradients for Media

In this section we consider the contribution of scattering media to the irradiance gradient. Our derivation is based on a reformulation of the radiance from the medium, L_m , in Equation 6.2. Instead of integrating to the closest visible surface, we introduce a visibility function and integrate to infinity. We rewrite the media irradiance, Equation 6.4, as

$$E_m(\mathbf{x}) = \int_{\Omega} \int_0^{\infty} T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t) \sigma_s(\mathbf{x}_t) L_i(\mathbf{x}_t \rightarrow -\vec{\omega}) V(\mathbf{x} \leftrightarrow \mathbf{x}_t) (\vec{\mathbf{n}} \cdot \vec{\omega}) dt d\vec{\omega}, \quad (6.12)$$

where we ignore the emission terms. The visibility function V returns one if the two arguments are mutually visible, and zero otherwise. If we collect all factors except visibility in a new term $L_V(\mathbf{x}, \mathbf{x}_t, \vec{\omega}) = T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t) \sigma_s(\mathbf{x}_t) L_i(\mathbf{x}_t, -\vec{\omega}) (\vec{\mathbf{n}} \cdot \vec{\omega})$, which we call *unshadowed radiance*, the media

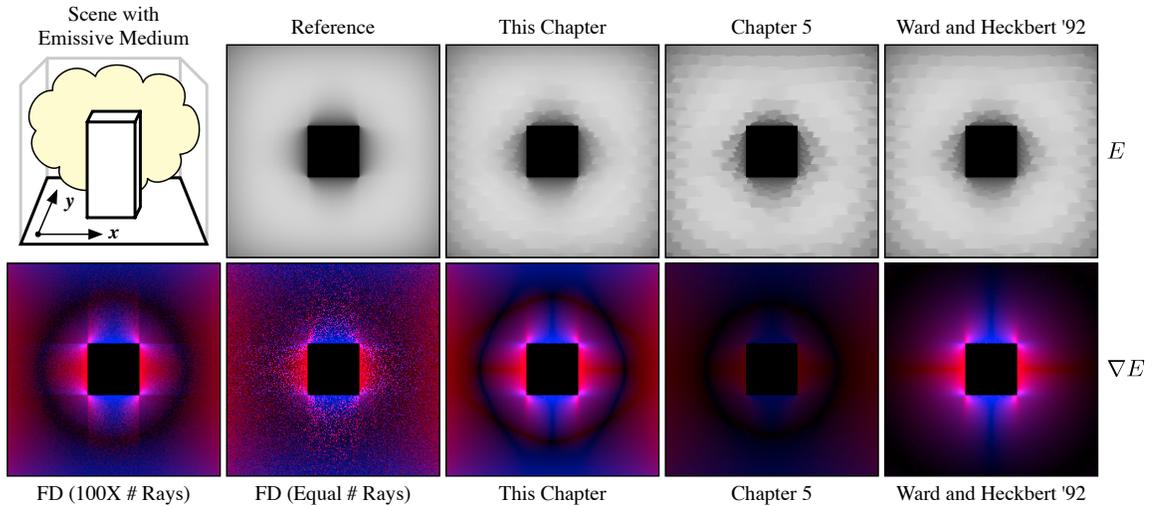


Figure 6.3: This modification of the scene from Figure 6.2 contains an **emissive** medium. The improved gradient formulation obtains accurate results by taking into account the medium and surface-medium occlusions, whereas previous methods ignore these effects and suffer from artifacts.

irradiance reduces to a product of unshadowed radiance and visibility,

$$E_m(\mathbf{x}) = \int_{\Omega} \int_0^{\infty} L_V(\mathbf{x}, \mathbf{x}_t, \vec{\omega}) V(\mathbf{x} \leftrightarrow \mathbf{x}_t) dt d\vec{\omega}. \quad (6.13)$$

Differentiation using the product rule leads to the gradient of media irradiance,

$$\nabla E_m(\mathbf{x}) = \int_{\Omega} \int_0^{\infty} \nabla L_V V + L_V \nabla V dt d\vec{\omega}, \quad (6.14)$$

where we have omitted the function arguments for brevity.

In the previous chapter, we derived gradients of multiple scattering within participating media in a similar manner. In those derivations, however, we assumed visibility was constant, and the gradient of V was ignored. In effect, we only considered the first term of Equation 6.14. In order to obtain the full gradient, we now also compute the gradient of visibility, as described in the following section.

6.4.1 Visibility Gradient

In order to account for the visibility gradient, we need to compute the following quantity:

$$\nabla_V E_m(\mathbf{x}) = \int_0^\infty \int_\Omega L_V \nabla V d\vec{\omega} dt, \quad (6.15)$$

$$= \int_0^\infty \tilde{V}_\Omega(\mathbf{x}, t) dt, \quad (6.16)$$

where we use ∇_V to denote that this is a gradient only of the visibility component, and we swap the order of integration to make the derivation more convenient. Additionally, we introduce the shorthand \tilde{V}_Ω for the hemispherical integral of $L_V \nabla V$.

If we consider a fixed t , \tilde{V}_Ω computes the gradient of the weighted visibility integral over the hemisphere. This is similar to Ward and Heckbert, except they compute the gradient of a weighted *radiance* integral over the hemisphere. Therefore, by substituting the radiance function with visibility, and replacing the weighting functions, we can compute this weighted visibility gradient using Ward and Heckbert's formulation. Intuitively, we are computing an irradiance gradient in a scene where the radiance function encodes V as black occluders in front of a distant environment map, L_V . Applying these modifications to Equation 3.32 results in

$$\tilde{V}_\Omega \approx \sum_{k=0}^{N-1} \left(\hat{u}_k \sum_{j=1}^{M-1} \nabla_{\hat{u}_k} A_{j,k} (V_{j,k} - V_{j-1,k}) L_V(j, k) + \hat{v}_k \sum_{j=0}^{M-1} \nabla_{\hat{v}_k} A_{j,k} (V_{j,k} - V_{j,k-1}) L_V(j, k) \right),$$

where the directional derivatives $\nabla_{\hat{u}_k} A_{j,k}$ and $\nabla_{\hat{v}_k} A_{j,k}$ are computed using Equations 3.42 and 3.46 and utilize the distance to occluders in their denominators. $V_{j,k}$ is a binary function indicating for each direction $\vec{\omega}_{j,k}$ whether the distance s to the nearest surface is less than or greater than t :

$$V_{j,k}(t, s) = \begin{cases} 0 & \text{if } t \geq s, \\ 1 & \text{if } t < s. \end{cases} \quad (6.17)$$

In order to integrate over t we perform ray marching, which discretizes the medium into "shells" as illustrated in Figure 6.1(d,e). Conceptually, we consider radiance from the medium as coming from expanding shells of radius t about the evaluation point \mathbf{x} and compute the visibility

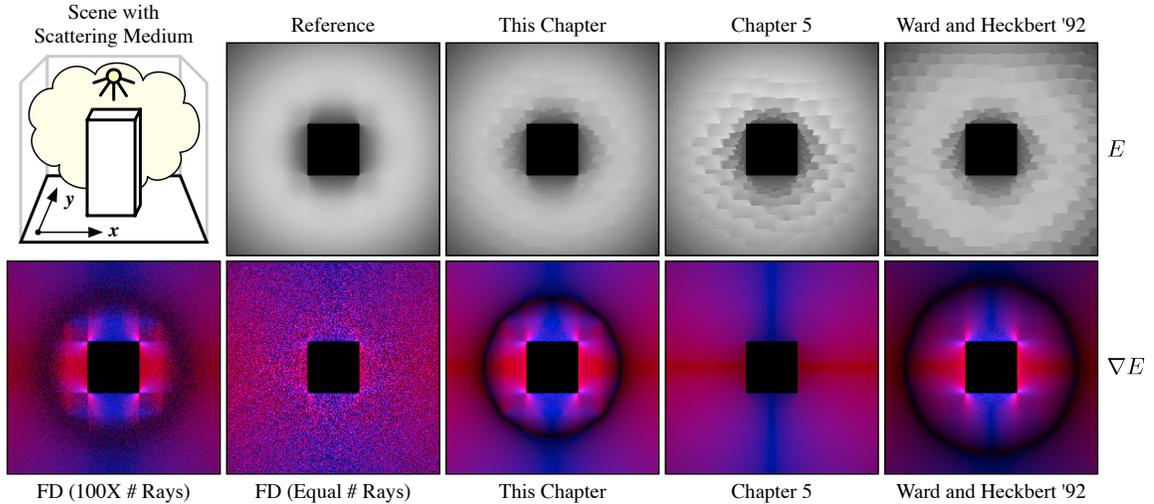


Figure 6.4: This modification of the scene from Figure 6.2 contains a spotlight and a **scattering** medium. The improved gradient formulation obtains accurate results by taking into account the medium and surface-medium occlusions, whereas previous methods ignore these effects.

gradient in Equation 6.16 as

$$\nabla_V E_m(\mathbf{x}) \approx \sum_{t=0}^{S-1} \tilde{V}_\Omega \Delta t. \quad (6.18)$$

In practice, however, ray marching is performed independently for each hemispherical direction, which swaps the order of the summations back again.

In summary, for a fixed distance t , we compute $L_V, \nabla L_V$ as in Chapter 5, and we compute a Ward and Heckbert-style gradient for $L_V \nabla V$ by evaluating the unshadowed radiance along visibility boundaries.

6.5 Implementation

Our new irradiance gradients can easily be added to a Monte Carlo renderer, which uses irradiance caching and supports participating media. The irradiance caching algorithm stays the same; just the gradient computation needs to be modified to account for the medium. Our procedure is illustrated in Figure 6.1. We provide pseudo-code for computing E_s and E_m and the gradients ∇E_s and ∇E_m in Algorithm 6.1 and describe the procedure in the following sections.

Algorithm 6.1: COMPUTEGRADIENT($\mathbf{x}, \vec{\mathbf{n}}$)

Data: \mathbf{x} is the position and $\vec{\mathbf{n}}$ is the normal at the evaluation location.
Result: $E_s, E_m, \nabla E_s$, and ∇E_m

```

1 begin
2   foreach cell (j, k) do
3     Determine distance s by tracing ray towards  $\vec{\omega}_{j,k}$ ;
4     Compute  $L(\mathbf{x}_s, -\vec{\omega}_{j,k})$  and  $\nabla L(\mathbf{x}_s, -\vec{\omega}_{j,k})$ ;
5      $Ls(j, k) = L(\mathbf{x}_s, -\vec{\omega}_{j,k})$ ;
6      $gL_s(j, k) = \nabla L(\mathbf{x}_s, -\vec{\omega}_{j,k})$ ;
7      $A \leftarrow \frac{\pi}{M \cdot N}$ ;
8     foreach cell (j, k) do
9        $T_r \leftarrow 1$ ;
10       $\nabla T_r \leftarrow 0$ ;
11      foreach ray-marching step through the medium do
12        Update  $T_r$  and  $\nabla T_r$ ;
13        Compute  $L_V$ , and  $\nabla L_V$  as in Chapter 5;
14        Compute  $L_V \nabla V$  using Equation 6.17;
15         $L_m(j, k) += L_V A \Delta t$ ;
16         $gL_m(j, k) += (L_V \nabla V + \nabla L_V A) \Delta t$ ;
17       $gL_s(j, k) = gL_s(j, k) T_r + Ls(j, k) \nabla T_r$ ;
18       $Ls(j, k) *= T_r$ ;
19    return  $E_s = A \sum_{j,k} Ls(j, k)$ ;
20    return  $\nabla E_s = \text{Equation 3.47} + A \sum_{j,k} gL_s(j, k)$ ;
21    return  $E_m = \sum_{j,k} L_m(j, k)$ ;
22    return  $\nabla E_m = \sum_{j,k} gL_m(j, k)$ ;
23 end
```

Initialization. The irradiance and gradient computation starts by creating a $M \times N$ array, where each element stores a HemiSample. This array is used to store the hemispherical samples needed to compute the irradiance and the irradiance gradient. Each HemiSample stores the distance s , the surface radiance and gradient L_s and gL_s , and the media radiance and gradient L_m and gL_m . All these values are initialized to zero.

Sampling Surfaces. The algorithm starts by looping over each cell (j, k) and tracing a ray in the $\vec{\omega}_{j,k}$ direction, generated using a cosine-weighted distribution [Ward and Heckbert, 1992]. We save the hit distance to the nearest surface, s , within the HemiSample and store the radiance $L(\mathbf{x}_s, -\vec{\omega}_{j,k})$ in L_s . If the intersected surface at \mathbf{x}_s is glossy, we also compute a gradient $\nabla L(\mathbf{x}_s, -\vec{\omega}_{j,k})$ and store this in gL_s . At the end of this stage we have a full hemispherical discretization of the

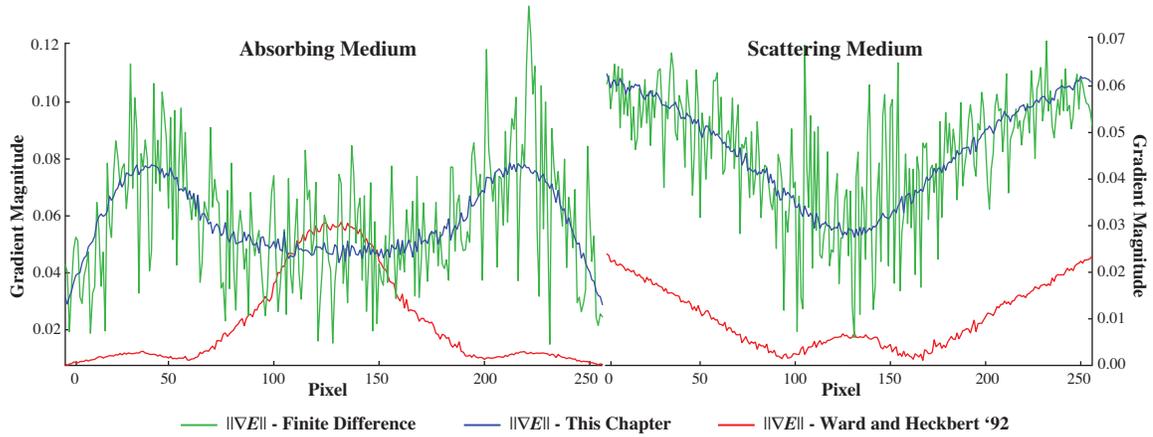


Figure 6.5: Visualizations of the gradient magnitude, (left) along a scanline in Figure 6.2, and (right) along a scanline in Figure 6.4. Our new gradient formulation correctly matches the profile of the finite difference gradient but with significantly less noise. Since it ignores participating media, the original Ward and Heckbert formulation does not produce the correct gradient in either of these situations.

scene, with a radiance, gradient, and distance to the nearest surface in each cell. The cell radiances and gradients do not yet take into account participating media.

Sampling Media. In order to account for the media, we perform ray marching individually for each cell (j, k) . As mentioned in Section 6.4, the integration over the hemisphere is the outer loop. For each cell, at each step in the medium we compute $L_V(\mathbf{x}_t)$ and $\nabla L_V(\mathbf{x}_t)$ using the techniques developed in Chapter 5. We also compute $L_V(\mathbf{x}_t)\nabla V(\mathbf{x}_t)$ by evaluating Equation 6.17 for the (up to) four boundaries of the current cell: (j_-, k_-) , (j_-, k_+) , (j_+, k_-) , and (j_+, k_+) . We then multiply the terms by the cell area $A_{j,k}$ and accumulate their contributions into L_m and gL_m .

At each step through the medium we maintain the current value of the transmittance T_r and its gradient ∇T_r . Once ray marching terminates, these values correspond to the transmittance $T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s)$ and transmittance gradient $\nabla T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s)$ from the evaluation location \mathbf{x} to the surface at \mathbf{x}_s . We use these transmittance values to augment the L_s and gL_s values computed in the previous stage in order to properly account for absorption by applying Equations 6.2 and 6.11.

Integrating. At the end of this process we integrate the irradiance and its gradient by summing over all the `HemiSamples`. Additionally, we add Equation 3.47 to the surface irradiance gradient to account for the changing cell areas.

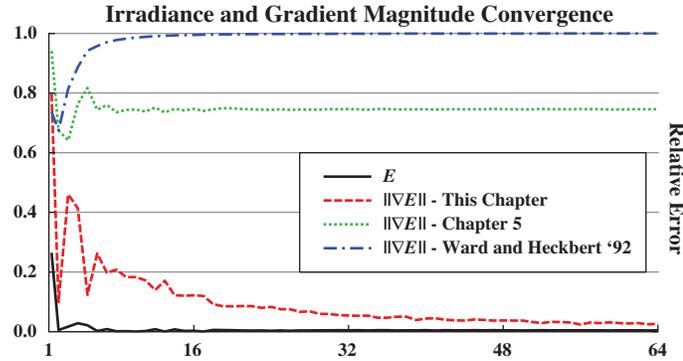


Figure 6.6: Relative errors for computing the irradiance and irradiance gradient for a single pixel in the scene from Figure 6.3. The x-axis plots M , the number of divisions in elevation angle, and N is set to $2M$. Previous gradient methods converge to incorrect values, leading to high error even with many samples.

6.6 Results

We implemented our irradiance gradients within a global illumination renderer written in C++. We compare our gradient method to Ward and Heckbert [1992] for a number of scenes. We also compare to a modification of Chapter 5’s gradients, where we compute irradiance gradients on surfaces by integrating over the hemisphere (instead of the whole sphere) and consider the cosine-weighted BRDF (instead of the phase function). In all of our results comparing irradiance caching, we only change the gradient method used. All other parameters, including the number of evaluation rays and the resulting cache point locations, are kept exactly the same. All timings are for an Intel Core 2 Duo 2.4 GHz machine using one core.

In Figures 6.2, 6.3, and 6.4 we demonstrate the importance of considering effects from the medium for variations of a scene containing absorbing, emitting, and scattering media, respectively. We directly visualize the x, y gradient components in these figures and compare them to “reference” solutions computed using finite differences. These renderings were computed at a resolution of 256×256 using $30 \times 60 = 1,800$ rays per evaluation. Our gradients match the reference solutions more faithfully than previous techniques and contain less noise than finite difference gradients computed using 180,000 evaluation rays per pixel. We demonstrate the accuracy of our gradients in Figures 6.5 and 6.6. Figure 6.5 plots the gradient magnitude along a scanline in the scenes from Figures 6.2 and 6.4. In Figure 6.6 we visualize the convergence rate for the irradiance and gradient computed at a single pixel in the scene from Figure 6.3. Previous

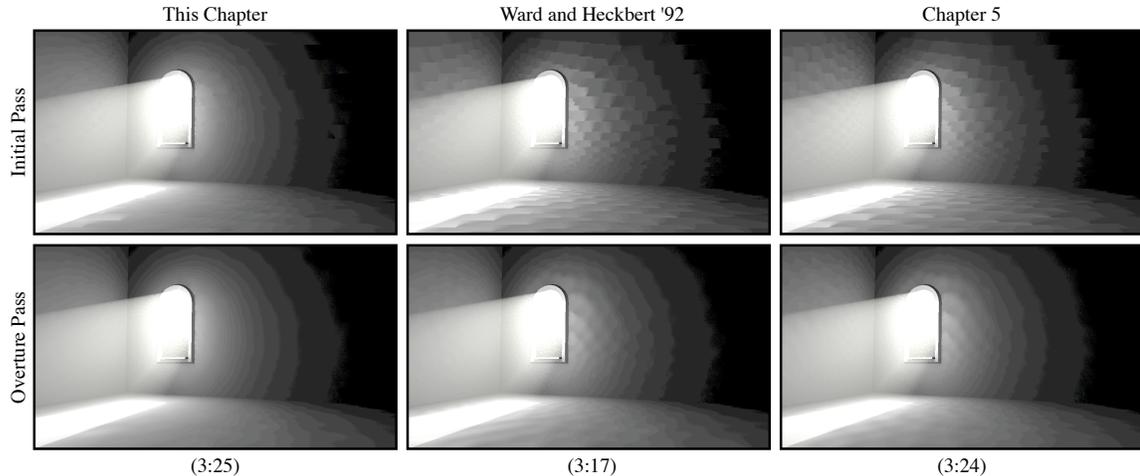


Figure 6.7: All the illumination on the walls of this room has first scattered off the floor or the bright beam of light through the window. Our improved gradient computation significantly reduces extrapolation artifacts (top). Using an “overture” pass, where the scene is re-rendered once all cache points are computed, previous techniques still suffer from interpolation artifacts (bottom), while our method produces smooth reconstruction.

techniques do not correctly capture the true gradient. The technique described in this chapter matches the finite difference gradient and quickly converges to the correct solution.

Figure 6.7 features a room with a strong volumetric light beam entering an open window. The walls and most of the floor in this scene are indirectly illuminated by this single beam of light. Using Ward and Heckbert’s gradient computation, this scene renders in 3:17 minutes at a horizontal resolution of 1K. Since this gradient method ignores the media, all light is incorrectly assumed to come from surfaces during gradient computation. This results in significant artifacts. Performing an “overture” re-rendering of the image is a common technique for improving the quality of irradiance cache renderings. This allows for interpolation, instead of extrapolation, to be used for the entire image. Even though the overture pass improves the quality of the rendering by reducing sudden discontinuities, the inaccurate gradients result in distracting “ripples” on the walls and floor. The gradients from Chapter 5 do account for the medium but fail to handle visibility changes, which also leads to significant artifacts. The gradients from this chapter, on the other hand, can correctly handle this scene and produce noticeably smoother reconstruction of the indirect illumination. The overhead of the improved gradients is fairly negligible, and we are able to render the same image in 3:25 minutes. The overture pass takes a fraction of the total render time for each method, and completes in under four seconds.

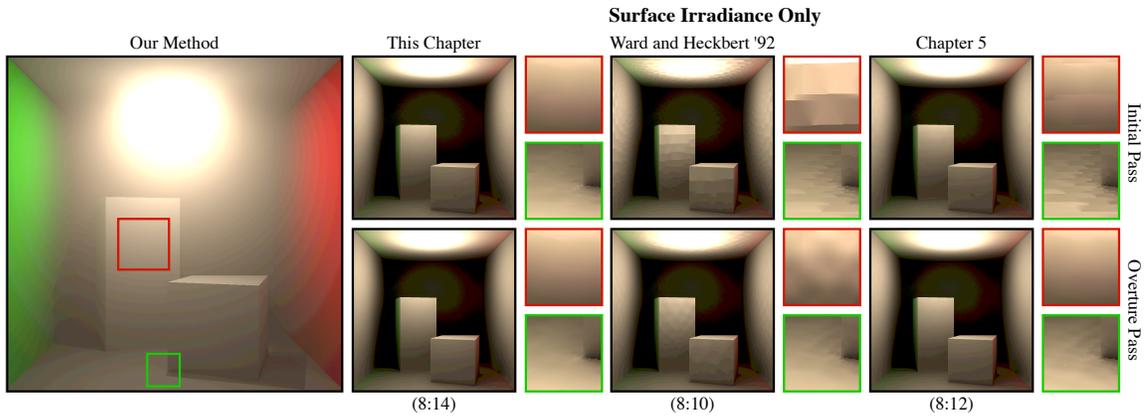


Figure 6.8: The classic Cornell box scene with a scattering medium. We compare the quality of irradiance caching and provide two zoomed-in regions for each result. Ward’s gradient formulation does not consider the medium, which results in inaccurate gradients and significant extrapolation artifacts. Our participating media gradients from Chapter 5 do a much better job, but still suffer from artifacts due to occlusion changes. The gradients from this chapter obtain the smoothest results by taking into account the media and occlusions.

Figure 6.8 contains a modification of the classic Cornell box rendered at a resolution of 1K. Ward’s method produces distracting artifacts even after an overture pass. Gradients computed using the techniques from Chapter 5 do a much better job in this scene, but still contain artifacts in areas with occlusion changes. The gradients derived in this chapter produce smooth results even in the initial pass. The overture pass takes less than five seconds for each method.

The disco light scene in Figure 6.9 presents a particular challenge for conventional gradient methods, since all lighting on surfaces is indirect lighting from the scattering medium. Due to this, Ward and Heckbert’s gradients suffer from significant artifacts both on the surface of the sphere and the ground plane. The gradients from Chapter 5, since they ignore visibility changes, also suffer from artifacts. This is particularly noticeable on the ground near the base of the disco light, where visibility changes are most prominent. The improved gradients are able to obtain a smooth reconstruction of the irradiance on the first pass and we can render this scene at 1K resolution in 10:33 minutes. This is only a slight overhead on top of the 10:30 minute render time using Ward and Heckbert’s gradients. The overture pass takes less than seven seconds to compute for each method.

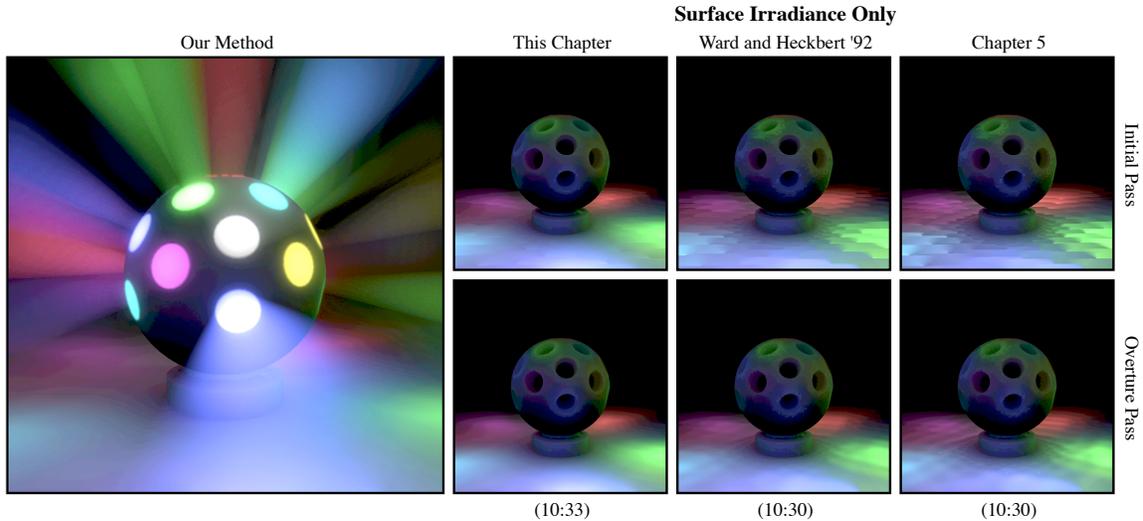


Figure 6.9: This disco light scene contains 21 light sources, but all lighting on surfaces is indirect. Ward’s gradient formulation assumes all lighting (even from the media) arrives from surfaces, which results in inaccurate gradients and significant extrapolation artifacts. The participating media gradients from Chapter 5, however, do not take the visibility term into account. The gradients developed in this chapter obtain the smoothest results by taking into account the media and occlusions.

6.7 Summary and Discussion

Our work on computing accurate irradiance gradients in scenes containing participating media exposes a number of limitations of current techniques and, in the process, suggests several exciting possibilities for future work.

Error Metric. In this chapter we were only concerned with improving the quality of interpolation by computing more accurate gradients. However, another significant contributor to the efficiency and quality of the irradiance caching method is the error metric used to compute valid radii of cache points. The split-sphere model, which drives the error metric, is geometry-driven and completely ignores lighting and all effects from participating media. This can lead to suboptimal cache point distributions. A more general error metric is desirable, but it is not immediately clear how to extend the split-sphere model to incorporate these effects. In Chapter 5 we derived an error metric specifically for participating media, and Křivánek et al. [2006] introduced a neighbor clamping technique. Creating a robust error metric that takes into account the local geometry as well as lighting from surfaces and media is a difficult problem which warrants further work.

Radiance Gradients. Though we presented all of our derivations within the context of irradiance, it is trivial to apply our approach to radiance caching on surfaces. Křivánek et al. [2005a] showed that, to compute radiance gradients, the cosine terms in Equations 6.5 and 6.6 can be replaced by a set of basis functions. This projection enables efficiently storing the full radiance function as a vector of coefficients and the radiance gradient as a corresponding set of gradient coefficients.

Radiance Gradients in Participating Media. In Chapter 5 we compute radiance gradients of single scattering from lights, single scattering from surfaces, and multiple scattering, but ignore visibility in all of these computations. This chapter provides the first step in computing visibility-aware radiance gradients for caching *within* participating media.

Computing single scattering from surfaces is very similar to the surface irradiance presented in Section 6.3. Ward and Heckbert, however, only consider gradients in 2D along the tangent plane. Our surface irradiance gradients could be extended to work in participating media by additionally deriving expressions for the gradients of cell area with respect to motion along a third axis.

Similarly, the definition of media irradiance in Equation 6.12 is nearly identical to the computation of multiple scattering. Extending Ward and Heckbert’s stratified gradients to consider motion along all three dimensions would also enable the use of a visibility gradient within the multiple scattering gradient.

Gradients of single scattering from light sources cannot be handled using Ward and Heckbert’s stratified gradients since these effects are typically not computed using hemispherical integration. For point light sources, however, visibility gradients can be efficiently approximated without tracing additional rays by using shadow maps and performing finite differencing. We, in fact, implemented this extension and use it when computing the gradient of single-scattered radiance embedded within the ∇L_V term of Equation 6.14.

Visibility Gradient. Ward and Heckbert’s stratified gradient formulation is intuitive and works well in practice; however, it is difficult to quantify its mathematical correctness since the gradient is performed after discretization. More recently, Ramamoorthi et al. [2007] presented an elegant

new visibility gradient formulation. Their approach is more mathematically rigorous since it presents an analytic expression for the gradients and only then performs discretization of this analytic expression. In Section 6.4.1 we modify Ward and Heckbert's gradients to estimate visibility gradients; however, an obvious alternative to this approach would be to directly use the gradients presented by Ramamoorthi et al. This approach would allow for a completely analytic expression for the media irradiance gradient. We explored this avenue, but Ramamoorthi's visibility integration suffers from a weak singularity. Though this singularity can be avoided by using adaptive sampling over the hemisphere, it makes it more cumbersome to integrate into the stratified ray marching process needed within our context. Nevertheless, we still believe this is a promising alternative and plan to investigate this approach further.

6.8 Conclusion

In this chapter we presented a method for accurately computing irradiance gradients on surfaces in the presence of participating media and occlusions. We applied our gradient derivations to the irradiance caching method and demonstrated that incorporating participating media and visibility information within the gradient is important for high quality irradiance interpolation in scenes containing these effects.

6.9 Acknowledgements

The material in this chapter is, in part, a reproduction of the material published in Wojciech Jarosz, Matthias Zwicker, and Henrik Wann Jensen. "Irradiance Gradients in the Presence of Participating Media." In *Computer Graphics Forum (Proceedings of EGSR 2008)*, 27(4), 2008. The dissertation author was the primary investigator and author of this paper. This work was supported in part by NSF grant CPA 0701992 and the UCSD FWGrid Project, NSF Research Infrastructure Grant Number EIA-0303622.