

# Irradiance Rigs

Hong Yuan

UMass Amherst, Disney Interactive Studios

Derek Nowrouzezahrai

University of Toronto, Disney Research Zürich

Peter-Pike Sloan

Disney Interactive Studios

**Abstract.** When using precomputed lighting on the static elements of a scene, it is important to light dynamic elements, such as animating characters, in a manner that is consistent with the static elements. One technique used in games is to project the external lighting into spherical harmonics (SH) from the center of each dynamic character, every frame, and then to use this SH vector to relight the character. For distant lights or small characters this approximation works well, however if a light source is close to the character, or if the character is large (e.g., a dinosaur or large vehicle), this approach can breakdown. We present a technique that uses more sophisticated local lighting representations which explicitly model spatial variation and error driven criteria to determine whether light sources should be evaluated analytically or in a custom rig. This can increase the efficiency of deferred rendering techniques by minimizing the fill rate required for analytic light sources. Least squares techniques can be used to efficiently compute the coefficients of a spatial rig model based on the actual geometry of each dynamic mesh.

## 1. Introduction

Realistically lighting dynamic characters in video games is a challenging problem. As the use of static precomputed lighting has become more common in games, it is important to ensure that the dynamic elements of the scene, such as animating characters, are lit in a manner that conveys their placement in the environment.

Irradiance Volumes [Greger et al. 98] were designed specifically for this problem. The spatially varying irradiance in a volume is explicitly computed and interpolated at run time [McTaggart 04] as dynamic objects move through the volume. For dynamic characters, a related alternative is to sample the static lighting at a character's feet and then synthesize a light probe from this sample [Chen 08]. While these techniques work in several circumstances, there remain key problems: memory and computation restrictions often restrict the number of samples used to evaluate and approximate the lighting, and deciding how to place and interpolate the light probes can be time consuming. Furthermore, mixing dynamic local light sources with these light probes can also be challenging.

Gears of War 2 [Smedberg and Wright 09] generated light maps from few direct light sources and many artist-placed fill lights used to approximate indirect lighting. Since evaluating hundreds of light sources analytically is impractical on current console hardware, all the lights are projected onto the spherical harmonic (SH) basis, from the center of each character every frame, decoupling lighting complexity at run time. This implicit representation of the Irradiance Volume has been used by several titles [McQuade 04, Smedberg and Wright 09], and while it does not share the problems of standard Irradiance Volumes, it does introduce two key problems. Firstly, light sources close to a dynamic object, like a flare, cannot be captured accurately since the captured spatial variation of the incident light does not match the actual spatial variation over the surface of the object. Secondly, large objects, like the Tyrannosaurs Rex used in the recent game Turok, can have significant lighting variation over the object, even for mid-to-distant light source. Lastly, both problems can be present at the same time.

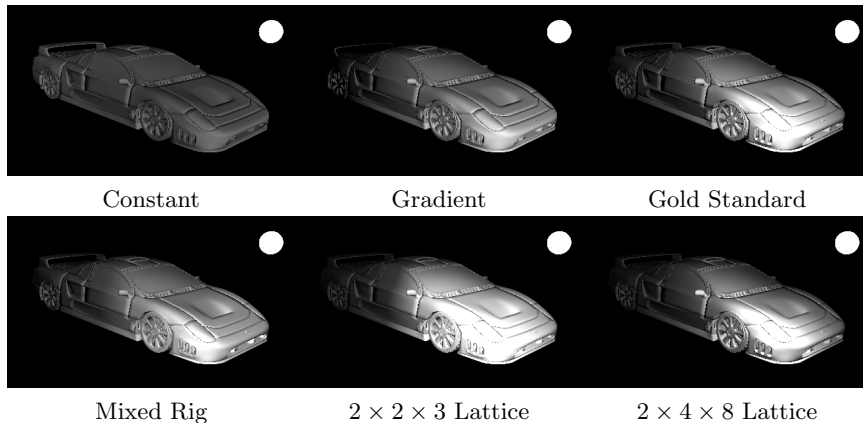
To address the first problem, we use error driven criteria to determine when a spherical light source should be evaluated in the local lighting rig or whether it should be evaluated analytically. For the second problem, we use local irradiance models that model a sufficient amount of spatial variation: SH gradients [Annen et al. 04] and regular lattices of SH coefficients. Unlike our work, previous techniques do not use any error criteria to decide whether a light source should be evaluated analytically or in an SH rig. For example, Forsyth [Forsyth 03] arbitrarily renders two point light sources analytically and the rest using SH.

Deferred rendering has garnered significant attention in the game develop-

ment community recently [Andersson 09, Engel 09], particularly due to its ability to manage large numbers of light sources. Our technique is focused on scenarios where the environment is already lit and only the characters need to be lit by several light sources. It would be possible to render the analytic and transition light sources in our paper using deferred rendering techniques, and an interesting avenue of future work would be to extend our technique to the scene lighting problem.

## 2. Irradiance Rigs

We adopt the use of SH [Sloan 08] from previous work on dynamic character lighting to encode the spherical incident irradiance [Ramamoorthi and Hanrahan 01] at a point in space. We are interested in rigs that reconstruct SH irradiance coefficients at a point in space. Given a point  $p$  in a bounded volume around a dynamic object, the spatially-varying irradiance coefficients  $\mathbf{l}_p$  can be computed using an abstract model function  $M(p, \mathbf{c})$ , where  $\mathbf{c}$  is a vector of model parameters for a given object, that is evaluated every frame.



**Figure 1.** Different lighting rigs applied to a car model.

### 2.1. Various Lighting Rigs

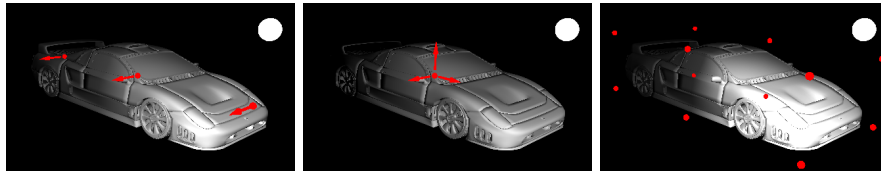
The simplest type of rig for a dynamic object is a spatially-constant SH function. Projecting point or spherical light sources into SH is simple [Sloan 08], and this rig’s only parameter  $\mathbf{c}$  is the SH projection of the light sources as observed from the center of the object,  $\mathbf{l}_c$ . Concretely, this rig model is  $\mathbf{l}_p = M(p, \mathbf{c}) = M(p, \mathbf{l}_c) = \mathbf{l}_c$ .

The simplest *spatially varying* rig augments the aforementioned “centroid-lighting” rig with a linear model of how lighting changes in space with its SH gradient,  $\mathbf{g}_x$ ,  $\mathbf{g}_y$ ,  $\mathbf{g}_z$  (The computation of SH gradient is detailed in Appendix A2). The model for this rig is  $\mathbf{l}_p = M(p, \mathbf{c}) = M(p, \{\mathbf{l}_c, \mathbf{g}_x, \mathbf{g}_y, \mathbf{g}_z\}) = \mathbf{l}_c + p_x \mathbf{g}_x + p_y \mathbf{g}_y + p_z \mathbf{g}_z$ . The gradient can be evaluated efficiently for spherical lights and precomputed numerically for other types of sources, such as disks.

Another common rig fixes a lattice around an object, where lighting is sampled at each lattice point  $p_i$  and tri-linearly interpolated to reconstruct irradiance over the surface of the object. Given an arbitrary reconstruction kernel  $b(p, p_i)$ , this model is  $\mathbf{l}_p = M(p, \mathbf{c}) = \sum_i b_i(p - p_i) \mathbf{l}_i$ , where  $\mathbf{l}_i$  is the SH irradiance at point  $p_i$ . This can be evaluated efficiently in hardware with volume textures.

For objects with different spatial scales along each axis, it can often be advantageous to employ a rig with different models for each dimension. The car’s surface irradiance in Figures 1 and 2 varies at different rates across its length, height and width. We apply a three-sample lattice model for the length, a gradient model for the height, and constant model for the width. This *mixed rig* model’s parameter vector is composed of only six SH vectors, yet generates results comparable to an  $8 \times 4 \times 2 = 64$ -sample lattice rig and outperforms a 12-sample rig. We compare to traditional (e.g., constant [Smedberg and Wright 09]) rigs and a ground-truth analytic evaluation (see Figure 1). Figure 2 visualizes some of the rig setups from Figure 1.

We have experimented with several other approaches, most notably, coupling radial basis functions with linear polynomials, combining functions and gradients in a lattice, as well as a form of hermite interpolation; however, all of these approaches performed poorly from a quality-per-unit-performance metric.



**Figure 2.** Sampling setups for mixed, gradient, and  $2 \times 2 \times 3$  lattice rigs.

## 2.2. Least-Squares to Approximate Rig Parameters

While the aforementioned lighting rigs provide varying degrees of flexibility, they all assume an arbitrary dynamic object. We will outline a straightforward, yet powerful approach for tailoring any lighting rig to the specific mesh

of a rigid dynamic object.

We distribute  $N$  sample points,  $\{s_1, \dots, s_n\}$ , over the surface of the dynamic object’s mesh using Lloyd’s repulsion algorithm [Gershon and Gray 92] then, given an abstract rig function  $M(p, \mathbf{c})$ , we wish to determine a setting of the rig-specific lighting parameters,  $\mathbf{c}$ , that most closely reproduce the exact lighting at the sample points (see Figure 3).

In the general case of an arbitrary rig, we can compute

$$\operatorname{argmin}_{\mathbf{c}} \sum_{i=1}^N (M(s_i, \mathbf{c}) - \mathbf{l}_i)^2, \quad (1)$$

where  $\mathbf{l}_i$  are the SH lighting coefficients at position  $s_i$ .

In several cases  $M$  can be modeled exactly as a linear operator,  $\overline{M}$ , that maps the rig-specific lighting parameters into spatially-varying illumination coefficients. In these cases, the elements of  $\overline{M}$  are linear in  $p$ , and element  $\overline{M}_{i,j}$  represents the influence of the  $j^{\text{th}}$  rig-specific lighting parameter on the  $i^{\text{th}}$  sample point. In these cases, we can solve for the optimal parameters with linear least-squares:

$$\overline{M} \cdot \mathcal{C} = \{\mathbf{l}_1, \dots, \mathbf{l}_n\} \longrightarrow \mathcal{C} = \underbrace{(\overline{M}^T \overline{M})^{-1} \overline{M}^T}_{L} \cdot \mathcal{L},$$

where the matrix  $L$  maps sampled lighting to optimal rig-specific light parameters, the matrix  $\mathcal{C}$  is comprised of the SH coefficients of each model parameter in  $\mathbf{c}$  laid out as rows, and similarly for  $\mathcal{L} = \{\mathbf{l}_1, \dots, \mathbf{l}_n\}$ .

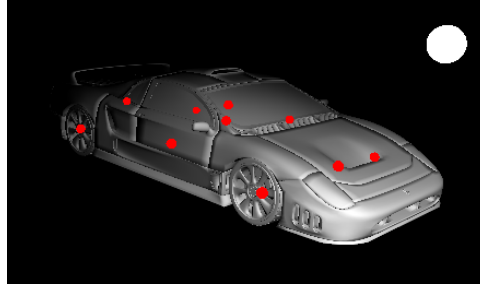
This linear least squares problem need only be solved once, yielding a matrix  $L$  that generates model parameters  $\mathbf{c}$  when multiplied by the SH coefficients at all the sample locations,  $\mathcal{L}$ . It is worth noting that the least squares technique only requires the evaluation of lighting at discrete sample points and analytic gradients are not necessary. This makes it most suitable for rigid objects.

The error driven criteria used to transition (see Figure 4) between analytic lighting evaluation and evaluation in the rig will be discussed next.

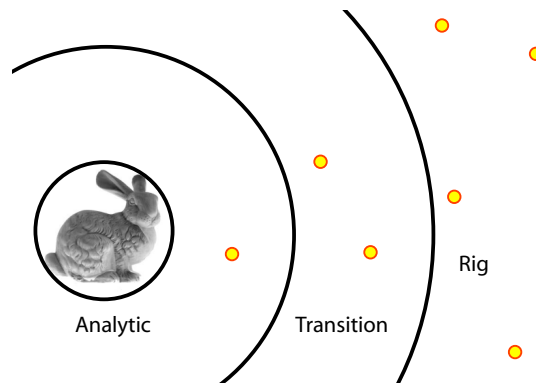
### 3. Transitioning In and Out of the Rig

#### 3.1. Error Driven Criterion

We focus on arbitrary spherical light sources. The analytical solution to compute their SH contribution is detailed in Appendix A1. To determine when to transition between analytic light source evaluation and rig evaluation, we tabulate the error in reflected radiance (as a function of normalized distance



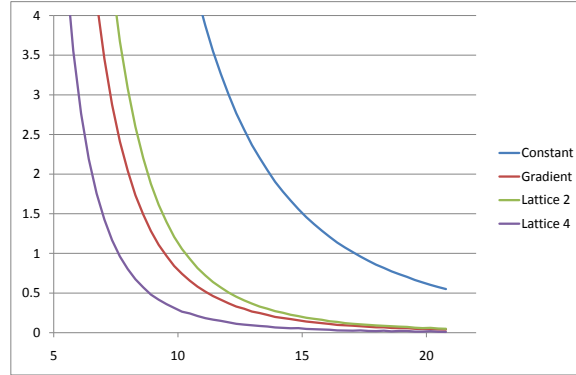
**Figure 3.** Least Squares Sample Locations for a Gradient Rig.



**Figure 4.** All lights have the same size and intensity, and the demarcation of regions depends strictly on the distance from object.

relative to the object's radius) between brute-force and rig evaluation for a point on the bounding sphere of a character closest to the light source to the center of the character. From the analytic equation of the irradiance from a spherical light source, we derive a normalization factor  $N(r)$  to map the error from a sphere light with an arbitrary radius to the unitized sphere used in the experimental setup. We define  $N(r)$  as  $N(r) = 4D^2/r^2$ , where  $D$  is a certain distance of a light source from the object center such that the reflected radiance at a point of unitized sphere and with normal pointing to the light source is 1. This function amounts to a simple analytic error evaluation,  $(\epsilon \times N(r))/c$ , where  $\epsilon$  is the error threshold,  $c$  is the light intensity. Figure 5 shows these error curves for several types of rigs, where horizontal axis is distance in terms of radius of the object and vertical axis is error measured in grey levels on an 8-bit display. For example, any curve's intersection with the  $y = 1$  line corresponds to a single grey level of error.

Given this target error value, we apply binary search to bracket the interval of interest, and linear interpolation is used to determine the transition distance

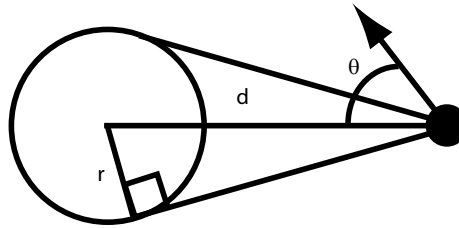


**Figure 5.** A plot of pixel error on an 8-bit display as a function of normalized distance to a dynamic character mesh.

threshold.

### 3.2. Analytic Evaluation

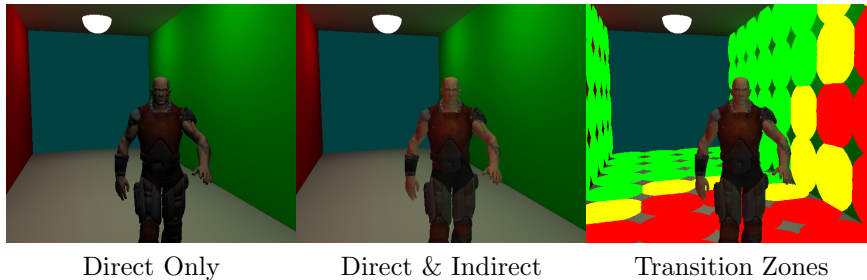
If a light source is near an object, the rig will not perform accurately. In these cases, the lights should instead be evaluated analytically. In this paper we focus on spherical light sources.



**Figure 6.** Geometry for computing irradiance from a sphere.

The analytic expression for the irradiance due to a spherical light source is complex when the light source is clipped by the horizon [Snyder 96], however it can easily be tabulated in a texture. Given the geometry in Figure 6, we parameterize the texture by  $\cos(\theta)$  and the sine of the half-angle for the cone that bounds the light,  $\frac{r}{d}$ , both of which can be easily computed in a shader. We found a  $128 \times 128$  texture to be accurate enough for our examples.

#### 4. Results



**Figure 7.** The effects of irradiance rig lighting. Right: **analytic**, **transitional**, and **rig evaluated** lighting zones.

The performance results in the figures are computed on a laptop with an nVidia Quadro 3600M at  $640 \times 480$ . For a representative demo, we have a scene with four direct light sources that can change, and lightmaps that are a function of these intensities to model a single light bounce. The character is illuminated by these four spherical light sources, as well as eight small, dynamic, but bright spherical light sources, and 192 disk light sources on the walls and floor that are also functions of the intensities of the primary light sources (to simulate indirect lighting). The right wall is green and the left wall is red to approximate color bleeding effects. Figure 7 illustrates the difference between just using the four direct light sources and using both the direct and indirect lights; the indirect lights add to the realism and placement of the character in the virtual scene, especially as the character animates and moves inside the room.

The scene in Figure 7 runs at 449 FPS with an irradiance rig with gradient and 62 FPS using only brute-force analytic light sources on a desktop nVidia QuadroFX 4800 GPU. A simple test scene with nine characters and 64 spherical lights runs at 350 FPS on the QuadroFX and 124 FPS on a lower-end laptop GPU, both at a resolution of  $800 \times 600$ .

#### 5. Discussion

We present Irradiance Rigs, an approach that uses error driven criteria to determine whether lights should be computed analytically or with a customized rig. This enables efficient rendering of characters with large numbers of light sources while maintaining accuracy for nearby light sources. We include efficient methods to evaluate spherical light sources and gradients of spherical light sources using SH in the appendices, and an efficient technique to evaluate



them using a carefully parameterized texture map.

It might be worth investigating using techniques similar to skinning for deformable characters, however this would only make sense for a moderately dense rig and would not be appropriate for some of the simpler rigs we consider in this work.

The high level idea of partitioning light sources into a “near field” that is rendered traditionally and a “far field” that is reconstructed using an inexpensive model could also be applied to deferred rendering of scenes, where it could significantly reduce the fill rate required.

**Acknowledgments.** This work was motivated by discussions with developers at Propaganda Studios. We would like to acknowledge Peter Shirley and Ladislav Kavan for constructive comments and discussion. The soldier model is from the DirectX SDK.

## References

- [Andersson 09] Johan Andersson. “Parallel Graphics in Frostbite - Current & Future.” In *SIGGRAPH '09: ACM SIGGRAPH 2009 classes*. New York, NY, USA: ACM, 2009.
- [Annen et al. 04] Thomas Annen, Jan Kautz, Fredo Durand, and Hans-Peter Seidel. “Spherical Harmonic Gradients for Mid-Range Illumination.” In *Proceedings of Eurographics Symposium on Rendering 2004*, edited by A. Keller and H. W. Jensen, pp. 331–336, 2004.
- [Chen 08] Hao Chen. “Lighting and Materials of Halo3.” In *Game Developers Conference*, 2008.
- [Engel 09] Wolfgang Engel. “The Light Pre-Pass Renderer: Renderer Design for Efficient Support of Multiple Lights.” In *SIGGRAPH '09: ACM SIGGRAPH 2009 classes*. New York, NY, USA: ACM, 2009.
- [Forsyth 03] Tom Forsyth. “Spherical Harmonics in Actual Games.” In *Game Developers Conference Europe*, 2003.
- [Gersho and Gray 92] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer, 1992.
- [Greger et al. 98] Gene Greger, Peter Shirley, Philip M. Hubbard, and Donald P. Greenberg. “The Irradiance Volume.” *IEEE Computer Graphics & Applications* 18:2 (1998), 32–43. Available online (<http://visinfo.zib.de/EVlib/Show?EVL-1998-168>).
- [McQuade 04] Loren McQuade. personal communication, 2004.
- [McTaggart 04] Gary McTaggart. “Half-Life 2 Source Shading.” In *Game Developers Conference*, 2004.

- [Ramamoorthi and Hanrahan 01] Ravi Ramamoorthi and Pat Hanrahan. “An efficient representation for irradiance environment maps.” In *SIGGRAPH 2001 Conference Proceedings, August 12–17, 2001, Los Angeles, CA*, edited by ACM, pp. 497–500. pub-ACM:adr: ACM Press, 2001.
- [Sloan 08] Peter-Pike Sloan. “Stupid Spherical Harmonics (SH) Tricks.” In *GDC '08*, 2008.
- [Smedberg and Wright 09] Niklas Smedberg and Daniel Wright. “Rendering Techniques in Gears of War 2.” In *GDC '09*, 2009.
- [Snyder 96] John Snyder. “Area Light Sources for Real-Time Graphics.” Technical Report MSR-TR-96-11, 1996.

## A1 Spherical light sources

If the projection is parameterized in terms of the radius  $r$  and distance  $d$  to the light source, one can derive a more efficient representation than in [Sloan 08]. The Zonal Harmonic (ZH) coefficients, including the per-band scaling coefficients, are simply:

$$\mathbf{z} = \left[ 2\pi \left( 1 - \sqrt{1 - \frac{r^2}{d^2}} \right), \frac{\pi r^2}{d^2}, \frac{\pi r^2 \sqrt{1 - \frac{r^2}{d^2}}}{d^2}, \frac{\pi r^2}{d^2} - \frac{5\pi r^4}{4d^4}, \right. \\ \left. \frac{\pi r^2 (4d^2 - 7r^2) \sqrt{1 - \frac{r^2}{d^2}}}{4d^4}, \frac{\pi r^2 (8d^4 - 28d^2 r^2 + 21r^4)}{8d^6} \right]. \quad (2)$$

Notice that no sine or cosine functions need to be evaluated, instead there is a single square root as well as many common sub-expressions.

## A2 SH Gradients for spherical lights

The gradient of the SH projection of a spherical light source, in the direction of the light source, can be computed analytically. For irradiance computation, we need only retain up to the the quadratic band, and the resulting directional derivative is also a ZH function. Given the light’s radius  $r$  and distance to the object  $d$ , the per-band coefficients are:

$$\mathbf{g} = \left[ \frac{2\pi r^2}{d^3 \sqrt{1 - r^2/d^2}}, \frac{2\pi r^2}{d^3}, \frac{2d^2 \pi r^2 - 3\pi r^4}{d^5 \sqrt{1 - r^2/d^2}}, \frac{\pi (2d^2 r^2 - 5r^4)}{d^5}, \right. \\ \left. \frac{\pi (8d^4 r^2 - 40d^2 r^4 + 35r^6)}{4d^7 \sqrt{1 - \frac{r^2}{d^2}}}, \frac{\pi (8d^4 r^2 - 56d^2 r^4 + 63r^6)}{4d^7} \right]. \quad (3)$$

The perpendicular directional derivatives could be computed semi-analytically, but it is faster to compute them using simple forward differences. Given these 3

directional derivatives,  $\mathbf{F}_{x'}, \mathbf{F}_{y'}, \mathbf{F}_{z'}$ , and corresponding directions,  $X', Y', Z'$ , solving for the gradient is simple:

$$\begin{vmatrix} X' \\ Y' \\ Z' \end{vmatrix} \begin{vmatrix} \mathbf{G}_x \\ \mathbf{G}_y \\ \mathbf{G}_z \end{vmatrix} = \begin{vmatrix} \mathbf{F}_{x'} \\ \mathbf{F}_{y'} \\ \mathbf{F}_{z'} \end{vmatrix} \Rightarrow \begin{vmatrix} \mathbf{G}_x \\ \mathbf{G}_y \\ \mathbf{G}_z \end{vmatrix} = \begin{vmatrix} X' & Y' & Z' \end{vmatrix} \begin{vmatrix} \mathbf{F}_{x'} \\ \mathbf{F}_{y'} \\ \mathbf{F}_{z'} \end{vmatrix},$$

where the directions (now stored as columns) are orthogonal by construction.

### Web Information:

Hong Yuan, UMass Amherst, Disney Interactive Studios  
(Hong.Yuan@disney.com)

Derek Nowrouzezahrai, University of Toronto, Disney Research Zürich  
(derek@disneyresearch.com)

Peter-Pike Sloan, Disney Interactive Studios  
(Peter-Pike.Sloan@disney.com)

Received [DATE]; accepted [DATE].