



**HAL**  
open science

# Efficient spherical harmonic shading for separable BRDF

Pierre Mézières, Mathias Paulin

► **To cite this version:**

Pierre Mézières, Mathias Paulin. Efficient spherical harmonic shading for separable BRDF. 14th ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia (SIGGRAPH ASIA 2021), Special interest group of the Association for Computing Machinery in computer graphics and interactive techniques, Dec 2021, Tokyo, Japan. pp.1-4, 10.1145/3478512.3488597 . hal-03358603

**HAL Id: hal-03358603**

**<https://hal.science/hal-03358603>**

Submitted on 29 Sep 2021

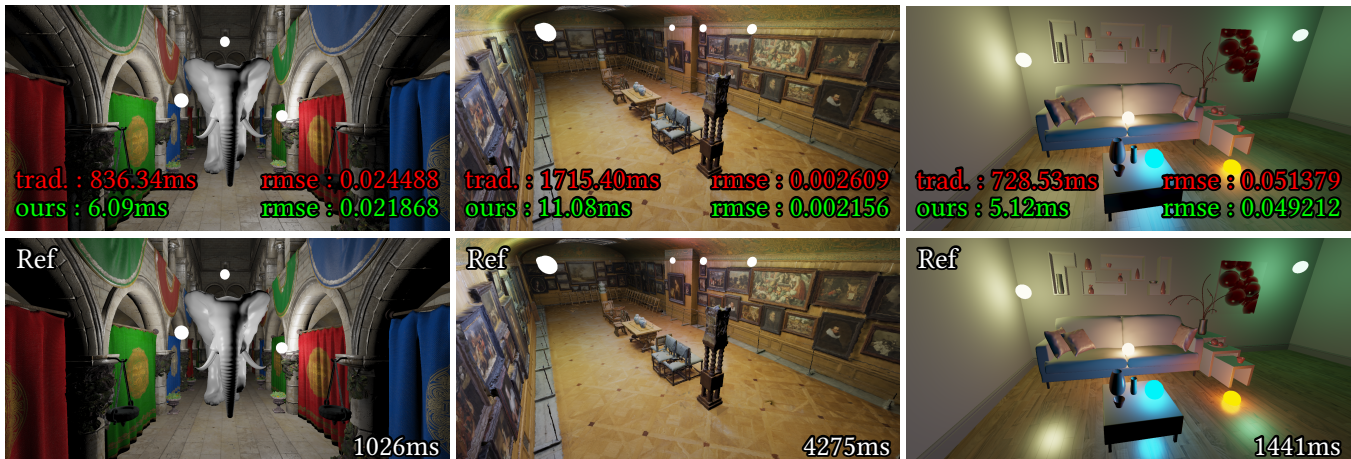
**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Efficient spherical harmonic shading for separable BRDF

Pierre Mézières  
IRIT, Université de Toulouse  
Toulouse, France  
pierre.mezieres@irit.fr

Mathias Paulin  
IRIT, Université de Toulouse  
Toulouse, France  
mathias.paulin@irit.fr



**Figure 1:** Computing per-pixel shading using Spherical Harmonics (SH) is a costly task that requires to compute a triple SH product when evaluating specular contribution. In this work we derive a new formulation of this product for separable BRDFs. Our approach reduces computation timings by two orders of magnitude and improves the quality of the final image (as measured by RMSE). Top: images obtained with our approach using 9 SH bands. Bottom: Reference images generated by sampling 10.000 rays for each light.

## ABSTRACT

Spherical Harmonics (SH) are commonly and widely used in computer graphics in order to speed up the evaluation of the rendering equation. With separable BRDF, the diffuse and specular contributions are traditionally computed separately. Our first contribution is to demonstrate that there is a simple relationship between both computations, but one-way, *i.e.* from specular to diffuse. We show how to deduce the diffuse contribution from the specular contribution, using a single multiplication. This replaces the use of tens of multiplications for some cases up to complex rotations for other cases. Our second contribution is an efficient way to compute the SH product between an arbitrary function and a clamped cosine, much less expensive than the traditional SH triple product.

Authors' addresses: Pierre Mézières, IRIT, Université de Toulouse, Toulouse, France, pierre.mezieres@irit.fr; Mathias Paulin, IRIT, Université de Toulouse, Toulouse, France, mathias.paulin@irit.fr.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Association for Computing Machinery.  
0730-0301/2021/9-ART \$15.00  
<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## CCS CONCEPTS

• Computing methodologies → Rendering; Rasterization; Real-time simulation.

## ACM Reference Format:

Pierre Mézières and Mathias Paulin. 2021. Efficient spherical harmonic shading for separable BRDF. *ACM Trans. Graph.* 1, 1 (September 2021), 4 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 1 INTRODUCTION

Spherical Harmonics (SH) are a well-known basis of functions used to solve efficiently the rendering equation. For instance, *Precomputed Radiance Transfer* (PRT) [Sloan et al. 2002] pre-computes and stores the lighting using SH at the vertices of a mesh and interpolate the SH over the triangles to compute per-pixel shading. 3D scenes are nowadays made of very refined meshes, and thus require long pre-computation time and large memory space to store and interpolate the SH coefficients.

In addition, SH have limited ability to represent high frequencies with limited memory requirements. As such, they first have been used to store and compute diffuse lighting contribution when using separable BRDFs. Recent approaches have been proposed to use SH for the specular contribution, but at the cost of increasing the rendering time. The main objective of this work is to reformulate SH operations to speed up the shading computation.

We observed that diffuse and specular contributions are always treated separately in the literature, but in practice there is a clear

computational similarity between them. In this work we demonstrate the gain obtained by taking advantage of this similarity, and expose an efficient algorithm to compute the SH product between any function and the clamped cosine one.

At first, we quickly recall the SH properties we need for this paper (Sec. 2). Then, we present how to compute the diffuse contribution from the specular one (Sec. 3) and the way we can compute efficiently the SH product between an arbitrary function and a clamped cosine (Sec. 4). We conclude by an evaluation of our proposal and explore the different use cases for both methods (Sec. 5).

## 2 BACKGROUND

**Spherical Harmonics.** Real spherical harmonics are defined by

$$Y_l^m(\theta, \phi) = \begin{cases} \sqrt{2}K_l^{|m|} P_l^{|m|}(\cos \theta) \sin(|m|\phi) & m < 0 \\ K_l^0 P_l^0(\cos \theta) & m = 0 \\ \sqrt{2}K_l^m P_l^m(\cos \theta) \cos(m\phi) & m > 0 \end{cases} \quad (1)$$

where  $K$  is a normalization factor given by

$$K_l^m = \sqrt{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}}. \quad (2)$$

Projection to SH obey the following equation

$$\mathbf{F}_l^m = \int_{\Omega} F(\omega) Y_l^m(\omega) d\omega. \quad (3)$$

**Double and triple product.** In literature, we find two classical operations on SH referenced as the double and triple product. Double product computes a spherical convolution between two functions, and is defined as

$$\begin{aligned} D &= \int_{\Omega} L(\omega) F(\omega) d\omega \\ &= \int_{\Omega} \left( \sum_{l_1, m_1} \mathbf{L}_{l_1}^{m_1} Y_{l_1}^{m_1}(\omega) \right) \left( \sum_{l_2, m_2} \mathbf{F}_{l_2}^{m_2} Y_{l_2}^{m_2}(\omega) \right) d\omega \\ &= \sum_{l_1, m_1} \sum_{l_2, m_2} \mathbf{L}_{l_1}^{m_1} \mathbf{F}_{l_2}^{m_2} \int_{\Omega} Y_{l_1}^{m_1}(\omega) Y_{l_2}^{m_2}(\omega) d\omega \\ &= \sum_{l, m} \mathbf{L}_l^m \mathbf{F}_l^m. \end{aligned} \quad (4)$$

The triple product computes the SH coefficients of the multiplication of two spherical functions:

$$\begin{aligned} \mathbf{T}_l^m &= \int_{\Omega} \left( \sum_{l_1, m_1} \mathbf{L}_{l_1}^{m_1} Y_{l_1}^{m_1}(\omega) \right) \left( \sum_{l_2, m_2} \mathbf{F}_{l_2}^{m_2} Y_{l_2}^{m_2}(\omega) \right) Y_l^m(\omega) d\omega \\ &= \sum_{l_1, m_1} \sum_{l_2, m_2} \mathbf{L}_{l_1}^{m_1} \mathbf{F}_{l_2}^{m_2} \int_{\Omega} Y_{l_1}^{m_1}(\omega) Y_{l_2}^{m_2}(\omega) Y_l^m(\omega) d\omega \end{aligned} \quad (5)$$

Usually, the integral in Eq.5 is precomputed to generate an efficient code that apply the SH product.

**Estimating diffuse and specular contributions.** Let's consider a typical rendering case with a light function  $L$ , a visibility  $V$  (cosine weighted according to surface normal) and a separable BRDF  $M$ . The diffuse contribution is computed by using the double product between  $L$  and  $V$  and then multiplying by the albedo. The specular contribution is given by computing the triple product between  $L$  and  $V$  and then apply the double product between this

result and the BRDF specular lobe. This process requires to compute double and triple products on the same two functions, which makes numerous computations redundant.

## 3 ESTIMATING DIFFUSE FROM SPECULAR

In this section, we show how to deduce the diffuse from specular contribution by computing the double product from the triple product with a single multiplication. Let's call  $O$ , the unit function over the sphere:

$$O(\omega) = \mathbb{1} = \sum_{l, m} \mathbf{O}_l^m Y_l^m(\omega). \quad (6)$$

Due to SH properties, only coefficient  $\mathbf{O}_0^0$  is non null and equal to

$$\mathbf{O}_0^0 = \int_{\Omega} Y_0^0 d\omega = \sqrt{\frac{1}{4\pi}} 4\pi = 2\sqrt{\pi} \approx 3.5449077. \quad (7)$$

Intuitively, if we multiply two spherical functions, *i.e.* apply the triple product, we can obtain the convolution between those two by applying the double product between the result of the triple product and  $O$ . Looking at the first SH coefficient obtained by triple product  $\mathbf{T}_0^0$ , we place the SH  $Y_0^0$  out of the integral as it's constant.

$$\mathbf{T}_0^0 = Y_0^0 \sum_{l_1, m_1} \sum_{l_2, m_2} \mathbf{L}_{l_1}^{m_1} \mathbf{F}_{l_2}^{m_2} \int_{\Omega} Y_{l_1}^{m_1}(\omega) Y_{l_2}^{m_2}(\omega) d\omega = Y_0^0 D. \quad (8)$$

By reversing, we obtain

$$\mathbf{T}_0^0 2\sqrt{\pi} = \mathbf{T}_0^0 \mathbf{O}_0^0 = D. \quad (9)$$

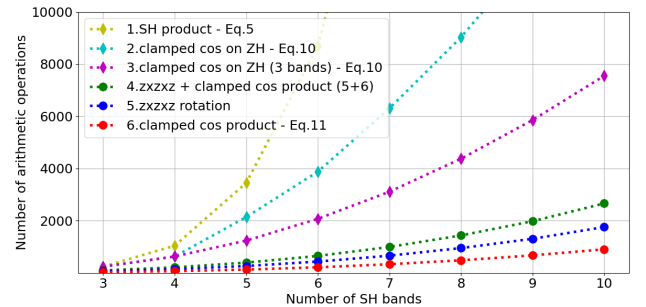
This signifies that when computing the triple product for two functions, the computation of the double product between the same functions is done with a single multiplication ( $D \approx 3.5449077 \mathbf{T}_0^0$ ) which corresponds to the double product between  $\mathbf{T}$  and  $\mathbf{O}$ .

## 4 EFFICIENT CLAMPED COSINE PRODUCT

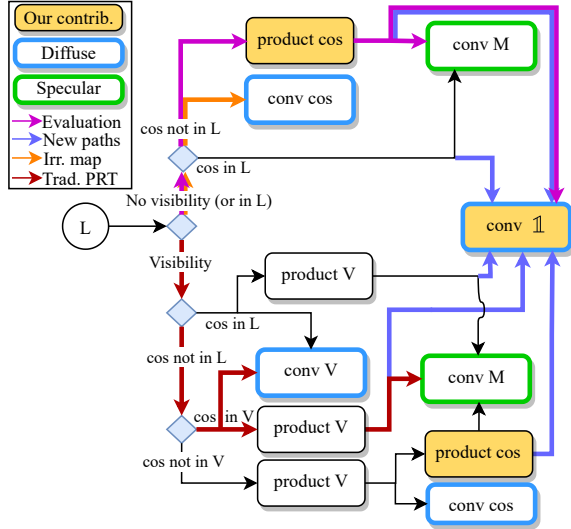
We present a method that drastically decreases the number of arithmetic operations needed to compute the SH triple product of a function  $L$  and a clamped cosine.

As the cosine is a circularly symmetric function, only the zonal harmonics (ZH) ( $m = 0$ ) coefficients are non null. By replacing the function  $F$  in the triple product equation (Eq. 5) by the equation of rotation of ZH functions [Sloan et al. 2005], we obtain:

$$\mathbf{T}_l^m = \sum_{l_1, m_1} \sum_{l_2, m_2} \mathbf{L}_{l_1}^{m_1} Y_{l_2}^{m_2}(n) \left( \sqrt{\frac{4\pi}{2l_2+1}} \mathbf{F}_{l_2}^0 C_{l_1 l_2 l}^{m_1 m_2 m} \right) \quad (10)$$



**Figure 2:** Number of arithmetic operations needed to compute different SH product up to a certain band.



**Figure 3:** Flowchart of typical SH rendering with separable BRDF. We annotated four paths: 1. paths we used for our evaluation, 2. new paths from our first contribution, 3. Irradiance map from [Ramamoorthi and Hanrahan 2001], 4. Traditional PRT.

where  $n$  is the normal surface and  $C_{l_1 l_2 l}^{m_1 m_2 m}$  are the Clebsch-Gordan coefficients corresponding to the integral in Eq.5.

This solution is more efficient than the classical SH triple product (Fig. 2). Moreover, as most of the cosine energy is caught on 3 SH bands, we can limit the cosine projection to this number of bands ( $l_2 < 3$ ) hence improving its evaluation. Nevertheless, there is a more efficient solution that does not imply a low frequency approximation of the cosine function. We can instead express the cosine in the world coordinate system and precompute it directly in the integral:

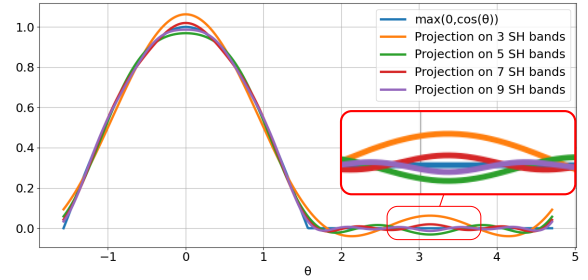
$$T_l^m = \sum_{l_1, m_1} L_{l_1}^{m_1} \left( \int Y_{l_1}^{m_1}(\omega) \max(0, \omega \cdot z) Y_l^m(\omega) d\omega \right) \quad (11)$$

When the light function is expressed in the same coordinate system, this solution provides fewer arithmetic operations. If light is not in the same coordinate system, adding a rotation step, as the  $zxzx$  method [Kautz et al. 2002], adds few operations and remains more efficient than other methods (Fig. 2). As noted for triple product, the parts of the equations into brackets (Eq.10 and Eq.11) are precomputed to generate the code that will apply the SH product.

## 5 EVALUATION AND USE CASES

As pointed out in the introduction, the goal of our methods is to compute the shading with SH using as few arithmetic operations and data storage as possible, and thus to reduce the computation time for rendering.

We compiled in Fig. 3 the different algorithms one can implement to compute shading using SH, considering four functions that appear in the rendering equation: light ( $L$ ), visibility ( $V$ ), brdf ( $M$ ) and the clamped cosine ( $\cos$ ). Paths that do not use visibility are particularly adapted to render fully dynamic scenes because shading do not depend on mesh-specific precomputations, unlike when visibility is pre-computed at each vertex. Using traditional PRT,



**Figure 4:** Clamped cosine projected on different number of SH bands. Even though the majority of the cosine energy is caught with few bands, it produces non-negligible ringing artefacts as it is clamped.

clamped cosine is packed with visibility in a precomputation step for each vertex. Similar but less common, cosine can be packed with light instead of visibility. This path can be used with dynamic visibility that do not take cosine into account, as it is included in the light function. Finally, path at the bottom, where  $L$ ,  $V$  and cosine are projected separately on SH is the more general. This path can be taken if everything is dynamic, light and visibility, for instance when using many polygonal lights [Wu et al. 2020] and with dynamic visibility [Ren et al. 2006]. In such case cosine cannot be packed with either function. Our contributions should be used for all of these algorithms to reduce their computational cost.

Eq. for SH product	Fig. 2*	rmse ( $\times 10^2$ )		
		Fig. 1a	Fig. 1b	Fig. 1c
Eq.5	1	2.4488	0.2609	5.1379
Eq.10	2	2.3652	0.2609	4.9844
Eq.10 ( $l_j < 3$ )	3	2.8814	0.5933	5.2027
Eq.11 (w. or wth. rotation)	6/4	2.1968	0.2156	4.9212

\* Corresponding line in Fig. 2

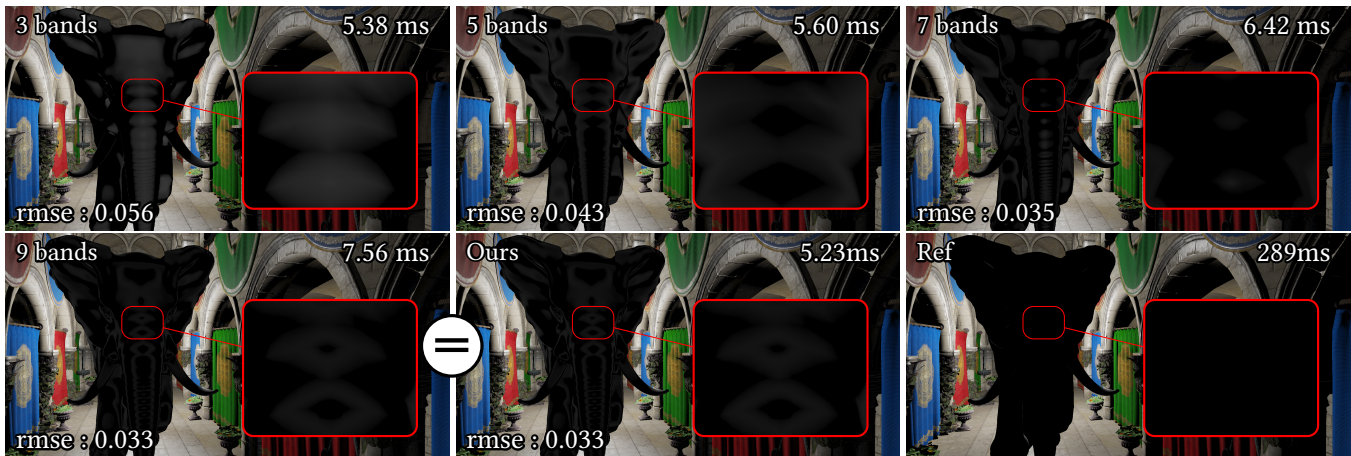
**Table 1:** RMSE ( $\times 10^2$ ) for the SH product. Our method always generates the lowest rmse because the cosine is not directly approximated by SH (Eq. 11). The important difference in magnitude between figures is due to the nature of materials used, mainly diffuse (Fig. 1b) or glossy (Fig. 1c). Specular is computed up to 9 SH bands while diffuse is computed with our method (Eq. 9).

Eq. for SH product	Fig. 2*	band limit			
		3	5	7	9
Eq.5	1	1.65	6.92	183.54	836.34
Eq.10	2	1.65	6.42	128.22	590.13
Eq.10 ( $l_j < 3$ )	3	1.63	3.78	41.54	289.88
Eq.11	6	1.46	1.78	2.92	6.09
Eq.11 (w. rotation)	4	1.59	2.19	5.08	15.78

\* Corresponding line in Fig. 2

**Table 2:** Timings (ms) with the different solution to apply the SH product with a clamped cosine. Specular is computed up to the corresponding band while diffuse is computed using our proposal (Eq. 9).





**Figure 5:** Ringing due to clamped cosine (Fig. 4). Scene contains only one light behind the elephant. Specular contribution is always computed on 9 SH bands. Diffuse contribution is computed with various number of SH bands, or evaluated from specular computation (our method). Negative color of shading result are clamped to zero, it explains why inset for 7 bands is darker than 9 while rmse is still higher. Ringing problem occurs too when using visibility with SH because of the frequency cut introduced by the clamping of the southern hemisphere of the unit sphere.

For all our experiments, we use the magenta path with spherical lights sources where lighting is computed at each pixel. We evaluate our results on a RTX 2080 with all renderings done at resolution  $1920 \times 968$ . To compare, we compute the traditional SH product (Eq. 5) using the same path for specular and Irr. map path for diffuse. To render dynamic scenes, cosine is precomputed on ZH and rotated on the fly [Sloan et al. 2005] before computing the product.

**Estimating diffuse from specular** Our method to compute double product from triple product allows to avoid the convolution needed to compute diffuse contribution, thus replacing  $N^2$  multiplications, where  $N$  is the number of SH bands, by a single multiplication. This method creates a new derivation on each path (Fig. 3). Moreover, diffuse is classically computed on 3 SH bands as the majority of the cosine energy is caught. This restriction produces non-negligible ringing artifacts, visible when light is only on one object’s side (Fig. 5), and is clearly explained by looking at the behaviour of clamped cosine projection on SH (Fig. 4). Our method removes these artifacts by computing diffuse lighting on the same number of bands as specular at no extra cost (Fig. 5).

**Efficient clamped cosine product** The efficient SH product with cosine is well adapted to render dynamic scenes. It fits into paths where the SH product with cosine is a specific step. We have shown that with our method the number of arithmetic operations is greatly reduced (Fig. 2), as well as computation time (Table 2). All our results were computed using Eq.11 with all SH coefficients expressed in the same frame. We also demonstrated (Table 2, last row) that when coefficients are not expressed in the same frame, thus requiring an additional SH rotation, our proposal still greatly improves the computation time. This last case is useful, e.g., when computing SH gradients as in [Wu et al. 2020]. Moreover, since the cosine is not approximated with SH (Eq. 11), the rmse is lower than the other methods (Table 1).

## 6 CONCLUSION AND FUTURE WORK

We have presented a general method to avoid redundant computations when computing both diffuse and specular contributions in

SH shading. We shown that diffuse contribution can be easily computed with a single multiplication from the specular computation. We also shown that the SH product between light, or an arbitrary function, and a clamped cosine can be done with only a few arithmetic operations. Our proposals then improve the computation time up to two orders of magnitude when using 9 SH bands.

It could be envisioned to apply the same principle to other functions that we did for cosine, *i.e.* replace cosine by any other function in Eq.11. Nevertheless, while improving the computation time for the triple product, the final number of required operations will be hard to predict.

Algorithms that fit into our classification (Fig. 3) should also benefit from our approaches by using the blue paths appropriately.

## ACKNOWLEDGMENTS

Sponza and Gallery taken from the McGuire Computer Graphics Archive. Living room taken from BlendSwap. Elephant taken from Sumner and Popovic.

## REFERENCES

- Jan Kautz, Peter-Pike Sloan, and John Snyder. 2002. Fast, arbitrary BRDF shading for low-frequency lighting using spherical harmonics. In *Eurographics Work. Render.*, Vol. 2. 291–296.
- Ravi Ramamoorthi and Pat Hanrahan. 2001. An efficient representation for irradiance environment maps. In *Proc. 28th Annu. Conf. Comput. Graph. Interact. Tech. SIGGRAPH 2001*. 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 SIGGRAPH 2006 Papers, SIGGRAPH '06*, 977–986. <https://doi.org/10.1145/1179352.1141982>
- Peter Pike Sloan, Jan Kautz, and John Snyder. 2002. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Proc. 29th Annu. Conf. Comput. Graph. Interact. Tech. SIGGRAPH '02*. 527–536. <https://doi.org/10.1145/566570.566612>
- Peter Pike Sloan, Ben Luna, and John Snyder. 2005. Local, deformable precomputed radiance transfer. In *ACM Trans. Graph.*, Vol. 24. ACM New York, NY, USA, 1216–1224. <https://doi.org/10.1145/1073204.1073335>
- Lifan Wu, Guangyan Cai, Shuang Zhao, and Ravi Ramamoorthi. 2020. Analytic spherical harmonic gradients for real-time rendering with many polygonal area lights. *ACM Trans. Graph.* 39, 4 (2020). <https://doi.org/10.1145/3386569.3392373>