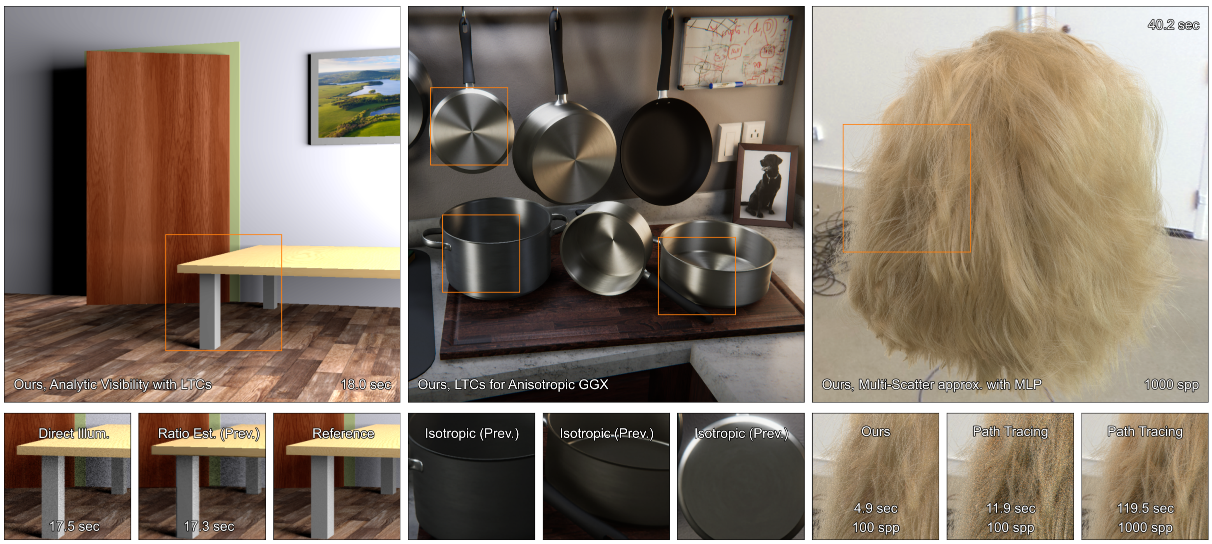
# Analytical & Neural approaches to Physically Based Rendering

Aakash KT
CVIT, International Institute of Information Technology, Hyderabad (IIIT-H)
India
aakash.kt@research.iiit.ac.in




(a) (b) (c)

**Figure 1: Summary of our contributions extending & accelerating Linearly Transformed Cosines (LTC) and multi-scattering in path tracing. (a) We first relax the *polygonal light only* and *no visibility* assumptions of a previous analytic method (LTC) for area light rendering. (b) Next, we relax the *isotropic BRDF only* assumption by supporting anisotropic BRDFs. (c) While both our preceding contributions focus on direct lighting, our third contribution accelerates the *recursion* (multi-scattering) in path tracing specifically for hair, using online trained neural networks.**

## ABSTRACT

Path tracing is ubiquitous for photorealistic rendering of various light transport phenomenon. At it's core, path tracing involves the stochastic evaluation of complex & recursive integrals leading to high computational complexity. Research efforts have thus focused on accelerating path tracing either by improving the stochastic sampling process to achieve better convergence or by using approximate analytical evaluations for a restricted set of these integrals. Another interesting set of research efforts focus on the integration of neural networks within the rendering pipeline, where these networks partially replace stochastic sampling and approximate it's converged result. The analytic and neural approaches are attractive from an acceleration point of view. Formulated properly & coupled with advances in hardware, these approaches can achieve much better convergence and eventually lead to real-time performance. Motivated by this, we make contributions to both avenues to accelerate path tracing. The first set of efforts aim to reduce the computational effort spent in stochastic direct lighting calculations from area light sources by instead evaluating it analytically. To this end, we introduce the analytic evaluation of visibility in a previously proposed analytic area light shading method. Second, we add support for anisotropic GGX to this method. This relaxes an important assumption enabling the analytic rendering of a wider set of light transport effects. Our final contribution is a neural approach that attempts to reduce yet another source of high computational load - the recursive evaluations. We demonstrate the versatility of our approach with an application to hair rendering, which exhibits one of the most challenging recursive evaluation cases. All our contributions improve on the state-of-the art and demonstrate photo-realism on par with reference path tracing.

## CCS CONCEPTS

• **Computing methodologies** → **Visibility**; **Ray tracing**; **Reflectance modeling**; *Volumetric models*; *Parametric curve and surface models.*


*Siggraph Asia Doctoral Consortium, December 12–15 2023, Sydney, Australia*

ACM ISBN 979-8-4007-0392-8/23/12…$15.00
https://doi.org/10.1145/3623053.3623372

## KEYWORDS

Direct Lighting, Linearly Transformed Cosines (LTC), GGX, Visibility, Hair Rendering, Neural Networks

**ACM Reference Format:**

Aakash KT. 2023. Analytical & Neural approaches to Physically Based Rendering. In *SIGGRAPH Asia 2023 Doctoral Consortium (SA Doctoral Consortium '23), December 12–15, 2023, Sydney, NSW, Australia.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3623053.3623372

## 1 INTRODUCTION

Achieving photorealism in rendered images requires intricate geometric objects, rich material models, and versatile lighting. Computer graphics has come a long way in rendering detailed 3D scenes with rich and complex material and lighting models. Modern physically based photorealistic rendering is achieved through Monte Carlo (MC) path tracing, which is widely adopted for visual effects in movies and in computer games.

At it's core, path tracing computes solutions to the *Light Transport Equation* (LTE) [Kajiya 1986; Veach 1998]::

$$L(x, \omega_v) = \int_{\Omega} \rho(x, \omega_v, \omega_l) L(t(x, \omega_l), -\omega_l) \, |\cos\theta_l| \, d\omega_l, \quad (1)$$

where on the left hand side, $L$ is the radiance from a 3D point $x$ in the view direction $\omega_v$. Inside the integrand, we have the Bidirectional Reflectance Distribution Function (BRDF) $\rho$, a cosine foreshortening factor on the angle $\theta_l$ between directions $\omega_l$ on the upper hemisphere $\Omega$ and the surface normal $n$ at $x$. We yet again have $L$ inside the integrand, which gives the radiance from the point $t(x, \omega_l)$ in the direction $-\omega_l$. We define $t$ as the *ray-casting operator* that gives the first point intersected by a ray from $x$ in the direction $\omega_l$.

Eq. 1 is stochastically evaluated with MC and converges to the correct result at a rate of $O(N^{-1/2})$. Artefacts due to MC sampling manifest as noise in the resulting rendered images. Practically, the noise in MC evaluation of Eq. 1 is more due it's recursive nature ($L$ appears on both sides).

Let's first consider *direct lighting*, which is one of the major sources noise in path tracing. We can rewrite Eq. 1 to only consider emitted radiance $L_e$ from all area light sources $A$ in the scene:

$$L(x, \omega_v) = \int_{A} \rho(x, \omega_v, \omega_l) L_e(t(x, \omega_l), -\omega_l) \, |\cos\theta_l| \, d\omega_l, \quad (2)$$

where the point returned by the ray-casting operator $t(x, \omega_l)$ will now lie on the light source which emits a radiance $L_e$. The above equation is no longer recursive, however it still exhibits noise from sampling. This has inspired research in analytic solutions to Eq. 2 which are attractive since the correct answer is computed at the outset without any noise. In Sect. 2.1, we discuss one such state-of-the-art analytic solution along with it's assumptions. In Sect. 2.2 & 2.3, we describe two of our contributions that relax these assumptions, enabling analytic rendering of a wider set of light transport effects.

Next, consider Eq. 1, for which deriving analytic solutions is considerably more challenging, primarily due to it's recursive nature. Much of computer graphics research has thus focused on better sampling strategies to improve it's convergence with MC. Recent efforts propose an interesting new direction - to use neural networks to approximate a part or whole of this recursion. The only constraint on these networks is that they need to be extremely small Multi Layer Perceptrons (MLPs). This is done for training & inference efficiency as well as generalizability with online training (as opposed to offline training on a dataset). In Sect. 3.1, we discuss such a recent method and show why it fails for high recursion depth. In Sect. 3.2, we describe our third contribution that solves this failure case using a similar neural approach. The resulting method is at best 70% faster than path tracing with a small amount of bias. We further formulate our method in such a way to provide control over both this bias & the speedup. The primary motivation & application of our method is to render hair, which is one of the prominent cases exhibiting high recursion depth.

This report is a summary of two of our recent peer-reviewed papers [KT et al. 2022, 2021] on relaxing assumptions for analytic direct lighting and a third peer-reviewed paper [KT et al. 2023] on using online trained MLPs to accelerate hair rendering.

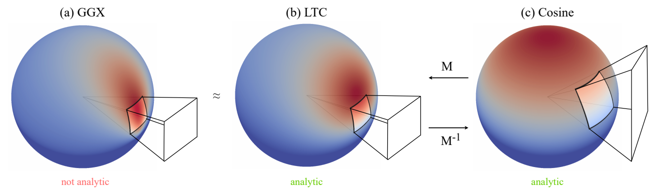



**Figure 2: Visualization of the BRDF response and the projection of a polygonal area light on a unit sphere centered around the shading point $x$. (a) A GGX lobe cannot be analytically integrated over the spherical domain covered by the area light. (b) An LTC represented by a matrix $M$ provides a good approximation to the GGX lobe, and the integral equals (c) the analytic integral of a cosine lobe over the light transformed by $M^{-1}$.**

## 2 ANALYTIC DIRECT LIGHTING FROM AREA LIGHTS

This section first discusses Linearly Transformed Cosines (LTC) [Heitz et al. 2016], which is a method to analytically compute the solution of the integral in Eq. 2 under certain assumptions. We then discuss two of our contributions that relax these assumptions.

### 2.1 Linearly Transformed Cosines

- **Assumption 0.** This assumption restricts $L_e$ to be spatially constant and diffuse (directionally constant). This means $L_e$ can now be taken out of the integral by accounting for the implicit *visibility function.* Note that Heitz et al. [Heitz et al. 2016] also describe a method to handle spatially varying area lights involving a second-level approximation.
- **Assumption 1.** The first assumption restricts $L_e$ to originate from polygonal area light sources. Denote $P$ as the spherical polygon subtended by the polygonal area light (Fig. 2). Together with assumption 0, we can rewrite Eq. 2 as:

$$L(x, \omega_v) = L_e \int_{P} \rho(x, \omega_v, \omega_l) V(x, t(x, \omega_l)) \, |\cos\theta_l| \, d\omega_l, \quad (3)$$

where $V$ defines the binary visibility of a point $t(x, \omega_l)$ on the light source from $x$ ($V = 1$ if visible, $V = 0$ otherwise).

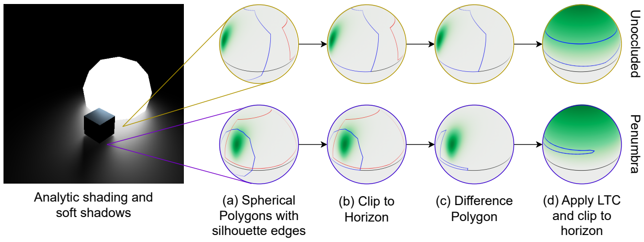



**Figure 3: High-level steps of our algorithm to compute shading and soft shadows from area lights. Unit spheres at an unoccluded shading point (yellow) and at a shading point in the penumbra (purple) are visualized with the corresponding spherical polygons of the light source (icosphere, marked as blue polygon) and occluder (cube, marked as red polygon).**

- **Assumption 2.** The next assumption is to ignore the visibility function $V$ in Eq. 3. In effect, we set $V = 1$ over the entire integration domain, irrespective of the visibility of the point $t(x, \omega_l)$.
- **Assumption 3.** The third assumption restricts the BRDF $\rho$ to be isotropic. This reduces the dimensionality of $\rho$ and paves the way for a simpler and memory efficient precomputation that is eventually used for analytic calculations.

With these assumptions, Heitz et al. [Heitz et al. 2016] show that given a linear transformation matrix $M$ which transforms the direction vectors of the clamped cosine distribution to the direction vectors of the BRDF $\rho$ (Fig. 2 (b), (c)), the resulting integral can be written as:

$$\begin{aligned} L(x, \omega_v) &= L_e \int_P \rho(x, \omega_v, \omega_l)\, |\cos\theta_l|\, d\omega_l \\ &\approx L_e \int_{P_o} \frac{1}{\pi} |\cos\theta_o|\, d\omega_o = L_e E(P_o) \end{aligned} \tag{4}$$

$P_o = M^{-1}P$ is the transformed spherical polygon, $\cos\theta_o = \omega_o \cdot n$ and $\omega_o = \frac{M^{-1}\omega_l}{||M^{-1}\omega_l||}$ are the transformed direction vectors. The irradiance integral $E$ can be analytically computed.

Note that the LTC matrix $M$ does not accurately approximate $\rho$. This is a source of bias and thus their method and all methods based on it are ultimately approximate.

## 2.2 Relaxing Assumptions 1 & 2

We extend the method of Heitz et al. [Heitz et al. 2016] to light sources of arbitrary 3D shapes, relaxing *assumption 1*. The key observation here is that for a convex 3D light source, the spherical polygon $P$ in Eq. 4 can be obtained using its silhouette edges as viewed from $x$. Non-convex lights are simple decomposed into a set of convex lights. These are then projected to the unit sphere to obtain a spherical polygon of the silhouette (Fig. 3(a) *blue* polygon). We then clip the spherical polygon to the horizon (Fig. 3(b)). Next, to obtain a polygon $P$ that represents only the visible region of the light source, we compute silhouette edges and corresponding spherical polygons for all potential occluders and clip them to the horizon (Fig. 3(a) *red* polygon). The clipped light and occluder polygons are then projected to a plane, where we perform a set difference between them. Finally, we reproject the resultant polygon to the unit sphere (Fig. 3(c)), apply LTC and clip the result to the horizon, to obtain $P_o$ for analytic evaluation of Eq. 4 (Fig. 3(d)). These steps are repeated for each light source in the scene. This procedure relaxes assumption 2 described above.

Our method accurately accounts for visibility within the analytic LTC framework, and produces better renderings in equal time as compared with stochastic methods (Fig. 1 (a), Fig. 3 left).

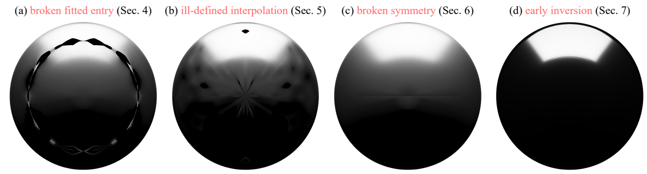



**Figure 4: Illustration of the problems to overcome for applying LTCs to anisotropic GGX.**

## 2.3 Relaxing Assumption 3

Heitz et al. [Heitz et al. 2016] use a fitting procedure to precompute LTC matrices for a given isotropic BRDF $\rho$ and store them in a 2D table. The difficulty is that this fitting approach cannot be simply extended to the more general anisotropic BRDFs. Indeed, in the isotropic case, the full dimensionality of LTCs is not used and this avoids several problems that arise in the anisotropic case.

The artefacts highlighted in Fig. 4 show that successfully bringing LTCs to anisotropic GGX requires *robust fitting*, *well-defined interpolation*, *valid symmetries* and *accurate storage*.

We achieve robust fitting by using the Sliced Wasserstein (SW) loss instead of a loss based on the magnitude of the BRDF. The SW loss is based on sample distributions which is the reason for it's robustness. A drawback of such a loss is that interpolation is not guaranteed, which we fix by *aligning* the LTC matrices by removing unnecessary rotations and flips. Both of these allow to dramatically reduce the precomputed table resolution. We exploit symmetries in GGX to further reduce the resolution to $8 \times 8 \times 8 \times 8$. Our method makes careful design choices resulting from new insights into the mathematical properties of LTCs. With this, we have relaxed assumption 3 described above.

The final outcome of our method is a 4D look-up table that yields a plausible and artifact-free LTC approximation to anisotropic GGX and is memory efficient. Fig. 1 (b) shows area-light shading using this table compared to the 2D table of Heitz et al. [Heitz et al. 2016]. Fitting LTCs to anisotropic GGX allows the analytic rendering of brushed metal appearance that was not possible before.

# 3 NEURAL NETWORKS FOR ACCELERATING RECURSION IN LTE

In this section, we first discuss Neural Radiance Caching (NRC) [Müller et al. 2021] which is the first method to use online trained MLPs to accelerate the recursion present in Eq. 1. Our contribution & method is discussed in Sect. 3.2.

## 3.1 Neural Radiance Caching

NRC uses a small MLP to fully approximate Eq. 1. During rendering, only short paths are traced and the radiance at the last vertex of

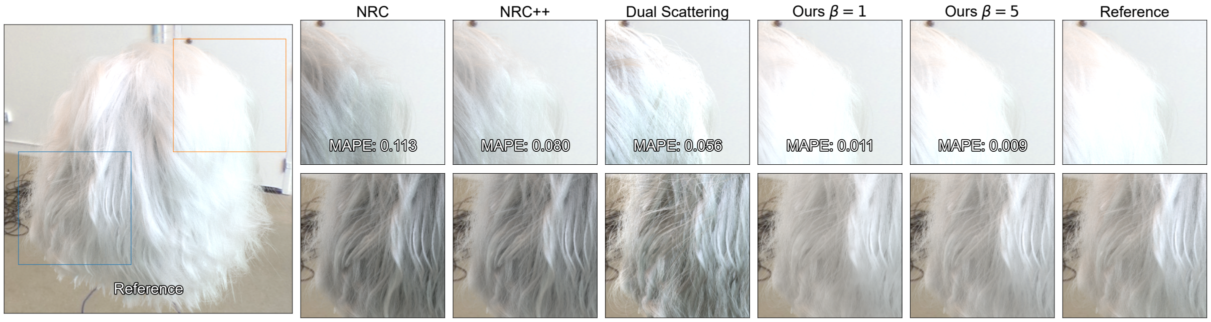



**Figure 5: We compare 5k spp renderings of our method for $\beta = 1, 5$ to 5k spp renderings of Neural Radiance Caching (NRC), dual scattering& a 10k spp path traced reference. We also compare to a version of NRC where all the training paths are unbiased (*NRC++*). All these methods fail to reproduce the soft look and saturation in hair.**

these short paths is approximated with the MLP. To achieve online training, they randomly choose a small percentage of the total paths and train on the radiance at each path vertex. This online training is extremely fast thanks to their use of fully-fused MLPs. Their method results in faster convergence than path tracing, at the cost of some bias from the network.

We are interested in rendering human hair, which exhibits very deep recursions due to long path lengths. Directly applying NRC to such cases fails due to the following reasons:

- NRC learns on the radiance at each path vertex, which is useful in surfaces (NRC's target application) with energy quickly degrading deeper in the path, and results in more training data for the same number of paths. However, for very deep recursion, this strategy leads to averaging in the network's output.
- NRC sets Eq. 1 as its target signal to be learnt by the MLP. However, if this signal is typically high frequency (which is the case in hair), which the small MLP is unable to represent well leading to artefacts in the final rendering.

Additionally, the resulting render from NRC will have bias from the network, which cannot be directly controlled. It is however desirable to have this control for certain applications.

### 3.2 Rendering Human Hair with MLPs

Reformulating Eq. 1 as a path integral:

$$L = \sum_{k=1}^{\infty} \int_{\Omega_k} f(\bar{\mathbf{x}}) \mathrm{d}\mu(\bar{\mathbf{x}}), \tag{5}$$

where $\Omega_k$ is the space of light paths $\bar{\mathbf{x}}$ and $f$ is the path contribution function. $\mathbf{x}_k$ and $\mathbf{x}_0$ are 3D points placed on a light source and the sensor respectively, and the differential measure $\mathrm{d}\mu(\bar{\mathbf{x}})$ models the area/volume integration for each vertex in the path.

We begin by setting a maximum depth $\beta << \infty$ in Eq. 5, giving the radiance $L'$ arriving at the pixel:

$$L' = \sum_{k=1}^{\beta} \int_{\Omega_k} f(\bar{\mathbf{x}}) \mathrm{d}\mu(\bar{\mathbf{x}}). \tag{6}$$

We estimate $\langle L' \rangle$ inplace of $\langle L \rangle$ during rendering. Modifying Eq. 5 in this way effectively sets a maximum depth for path termination, which introduces bias/error in $\langle L' \rangle$, given by:

$$E = L - L'. \tag{7}$$

We task an MLP $\sigma$ to learn $\langle E \rangle$ at the primary path vertex $\mathbf{x}_1$, given the view direction $\omega = \frac{\mathbf{x}_0 - \mathbf{x}_1}{||\mathbf{x}_0 - \mathbf{x}_1||}$ and the hair tangent $\mathbf{t}_1$ at $\mathbf{x}_1$:

$$\sigma(\mathbf{x}_1, \omega, \mathbf{t}_1) \approx \langle E \rangle. \tag{8}$$

Our formulation solves the limitations of NRC mentioned in the previous section while also naturally allowing control over the renderer's bias & speedup via the parameter $\beta$. Fig. 5 compares renderings of our approach for $\beta = 1, 5$ with NRC and dual scattering. Our method quantitatively & qualitatively outperforms both methods and is has better convergence than path tracing (Fig. 1 (c)).

## 4 CONCLUSION

We presented approaches for accelerating physically based rendering using analytic & neural methods. First, we relaxed three assumptions made by LTCs, which is a previous analytic method, enabling the rendering of a wider set of light transport effects. Second, we presented a neural approach to approximate high recursion depth in path integrals with a specific application to hair rendering. All our proposed methods outperform the state-of-the-art in analytic & neural domains and we hope will inspire future research.

## REFERENCES

Eric Heitz, Jonathan Dupuy, Stephen Hill, and David Neubelt. 2016. Real-Time Polygonal-Light Shading with Linearly Transformed Cosines. *ACM Trans. Graph.* 35, 4, Article 41 (July 2016), 8 pages. https://doi.org/10.1145/2897824.2925895

James T. Kajiya. 1986. The rendering equation. In *Computer Graphics*. 143–150.

Aakash KT, Eric Heitz, Jonathan Dupuy, and P. J. Narayanan. 2022. Bringing Linearly Transformed Cosines to Anisotropic GGX. *Proc. ACM Comput. Graph. Interact. Tech.* 5, 1, Article 12 (May 2022), 18 pages. https://doi.org/10.1145/3522612

Aakash KT, Adrian Jarabo, Carlos Aliaga, Matt Jen-Yuan Chiang, Olivier Maury, Christophe Hery, P. J. Narayanan, and Giljoo Nam. 2023. Accelerating Hair Rendering by Learning High-Order Scattered Radiance. *Computer Graphics Forum* (2023). https://doi.org/10.1111/cgf.14895

Aakash KT, Parikshit Sakurikar, and P. J. Narayanan. 2021. Fast Analytic Soft Shadows from Area Lights. In *Eurographics Symposium on Rendering - DL-only Track*, Adrien Bousseau and Morgan McGuire (Eds.). The Eurographics Association. https://doi.org/10.2312/sr.20211295

Thomas Müller, Fabrice Rousselle, Jan Novák, and Alexander Keller. 2021. Real-time neural radiance caching for path tracing. *ACM Trans. Graph.* 40, 4 (2021), 1–16.

Eric Veach. 1998. *Robust Monte Carlo methods for light transport simulation.* Stanford University.