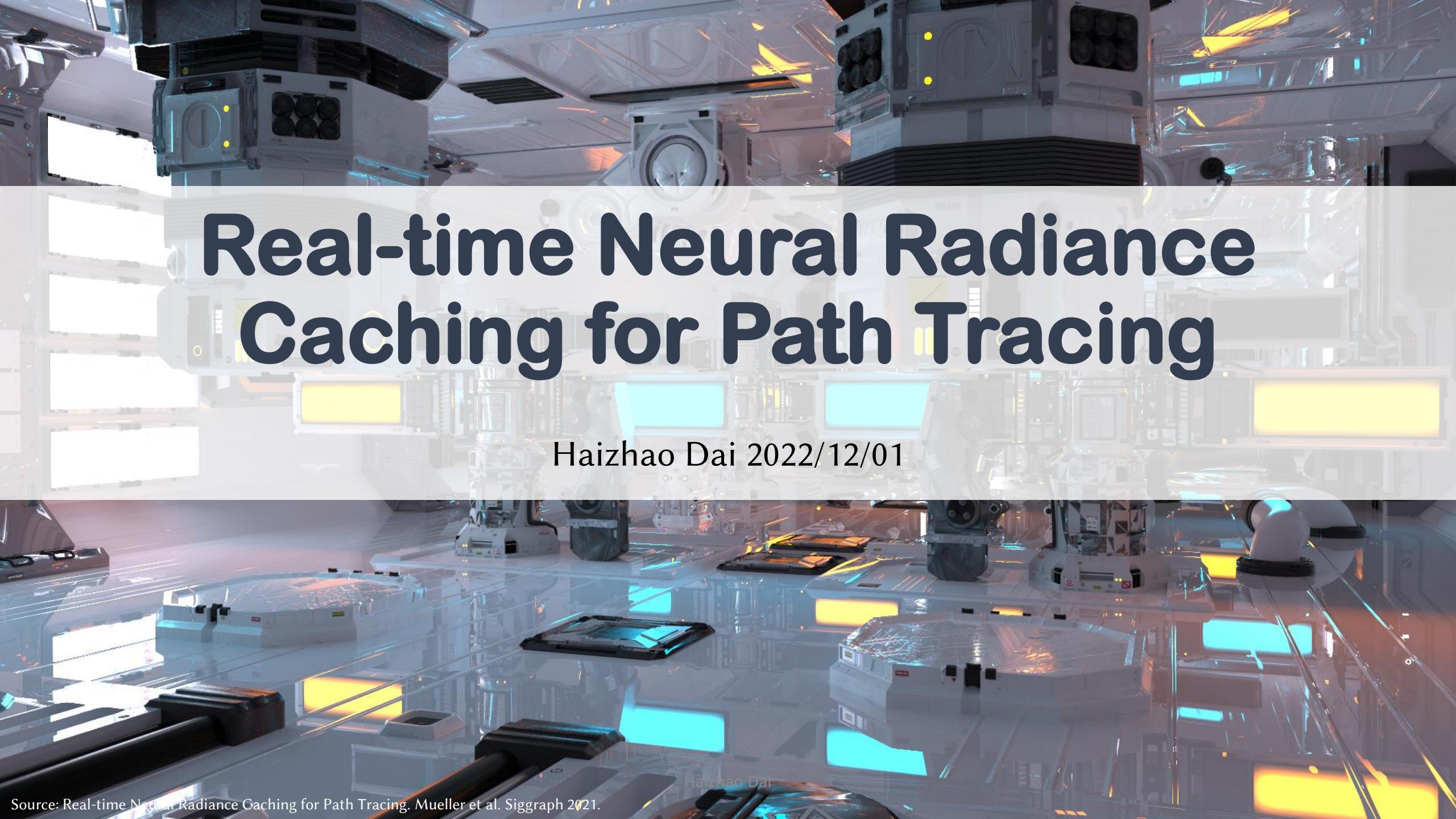


Real-time Neural Radiance Caching for Path Tracing

Haizhao Dai 2022/12/01

Source: Real-time Neural Radiance Caching for Path Tracing. Mueller et al. Siggraph 2021.

Overview

- Path Tracing
- Real-time Ray Tracing
- ReSTIR & Radiance Caching
- Neural Radiance Caching + Fully Fused Neural Networks

Path Tracing (Kindly remind: CG HW4)

- Rendering equation describes the light transport:
  - $L_o(p_1 \to p_0) = L_e(p_1 \to p_0) + \int_{\mathcal{H}^2} f(p_2 \to p_1 \to p_0) L_i(p_2 \to p_1) |\langle \mathbf{n}_{p_1}, \omega_i \rangle| \mathrm{d}\omega_i$

- Consider light transporting from another point p_2:
  - $L_i(p_2 \to p_1) = L_e(p_2 \to p_1) + \int_{\mathcal{H}^2} f(p_3 \to p_2 \to p_1) L_i(p_3 \to p_2) |\langle \mathbf{n}_{p_2}, \omega_{ii} \rangle| \mathrm{d}\omega_{ii}$

- Merge together, we complete the path tracing from p_2 to p_1 and to camera:
  - $L_o(p_1 \to p_0) = L_e(p_1 \to p_0)$ <-Illumination from light source

Direct Illumination-> $+ \int_{\mathcal{H}^2} f(p_2 \to p_1 \to p_0) L_e(p_2 \to p_1) |\langle \mathbf{n}_{p_1}, \omega_i \rangle| \mathrm{d}\omega_i$

Indirect Illumination-> $+ \int_{\mathcal{H}^2} f(p_2 \to p_1 \to p_0) \int_{\mathcal{H}^2} f(p_3 \to p_2 \to p_1) L_i(p_3 \to p_2) |\langle \mathbf{n}_{p_2}, \omega_{ii} \rangle| \mathrm{d}\omega_{ii} |\langle \mathbf{n}_{p_1}, \omega_i \rangle| \mathrm{d}\omega_i$

Path Tracing (Kindly remind: CG HW4)

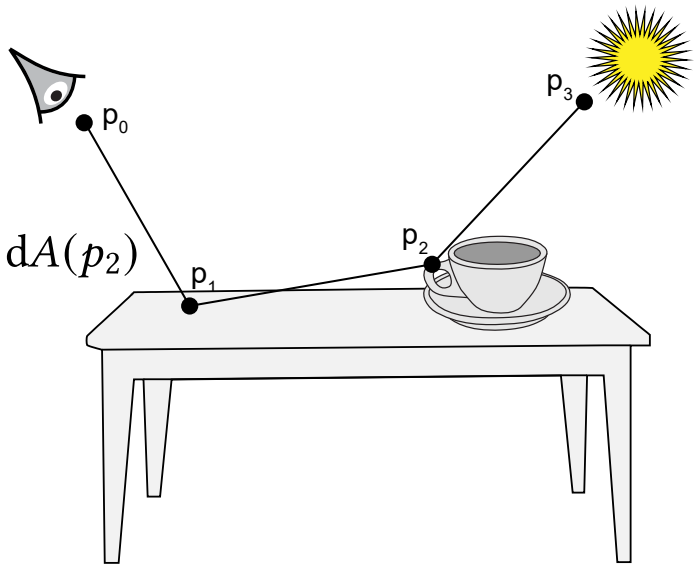
- Directly sample from light sources (Next Event Estimation):

$$L(p_1 \to p_0) = L_e(p_1 \to p_0) + \int_A f(p_2 \to p_1 \to p_0) L_e(p_2 \to p_1) G(p_2 \leftrightarrow p_1) \mathrm{d}A(p_2) + \int_A f(p_2 \to p_1 \to p_0) \int_A f(p_3 \to p_2 \to p_1) L_e(p_3 \to p_2) G(p_3 \leftrightarrow p_2) G(p_2 \leftrightarrow p_1) \mathrm{d}A(p_3) \mathrm{d}A(p_2)$$

- This process can go on forever...

$$L(p_1 \to p_0) = \sum_{i=1}^{\infty} P(\bar{p}_n) \quad \text{where}$$

$$P(\bar{p}_n) = \underbrace{\int \cdots \int_A}_{n-1} L_e(p_n \to p_{n-1}) \prod_{i=1}^{n-1} f(p_n \to p_{n-1} \to p_{n-2}) G(p_n \leftrightarrow p_{n-1}) \mathrm{d}A(p_n) \cdots \mathrm{d}A(p_2)$$



Source: Physically Based Rendering: From Theory To Implementation Third Edition. Pharr et al. 2018.

Real-time Ray Tracing

- What is the problem of the path tracing?
  - Extremely slow for ray-object intersection, huge number of sampling.
  - 1920 * 1080 * 1024 * 64 = 136 B / 398 M for vanilla NeRF.

8 spp

1024 spp

Source: Physically Based Rendering: From Theory To Implementation Third Edition. Pharr et al. 2018.

Real-time Ray Tracing

- Difficulties:
  - Everything should done in ~33 ms.
  - CUDA/GPU/rendering API/parallel computing.
  - Dynamic scene.
  - How to find the important lights in millions of light sources?
- Possible Solutions:
  - A data structure reducing the number of intersection – Displacement Mapping BVH.
  - Find the optimal sampling strategy for each term of rendering equation – ReSTIR.
  - Deep learning: Video frame interpolation/Super sampling – DLSS.
  - Neural network + Reduce the number of path to trace – NRC.

Why NV is so good at CG???

ReSTIR

- ReSTIR is short for Reservoir-based SpatioTemporal Importance Resampling.
  - Reservoir Sampling.
  - Multi-importance Sampling + Resampled Importance Sampling.
  - Spatial/Temporal/Visibility Reuse.

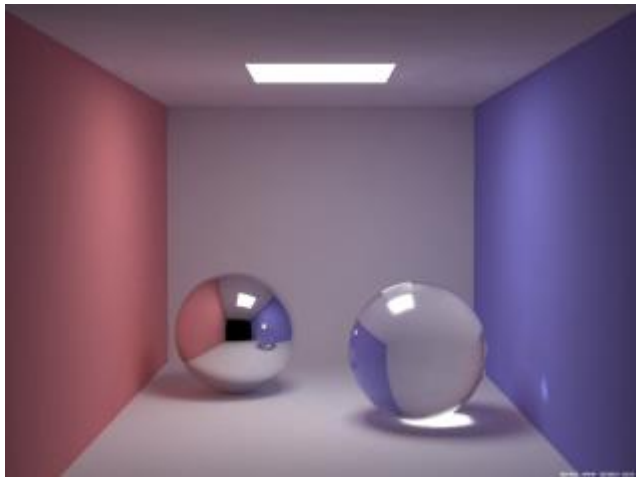
Only aimed for direct illumination!

Striped-comparisons in 100 ms rendering: Light-BVH, Biased ReSTIR, Unbiased ReSTIR, Reference.

Source: Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. Bitterli et al. Siggraph 2020.

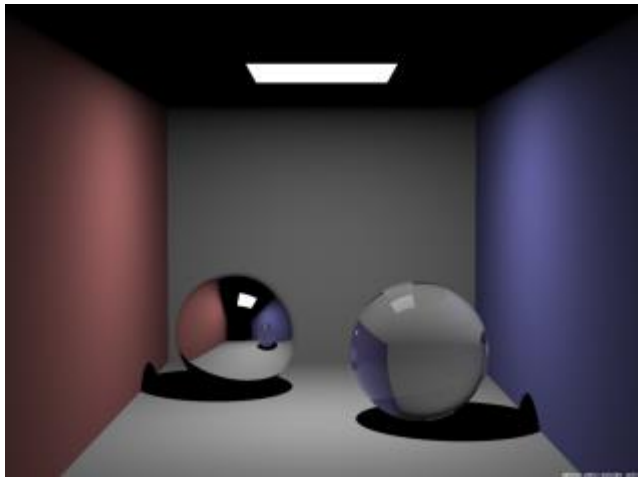
Irradiance Caching

- Irradiance at diffuse surface varies smoothly in the scene.



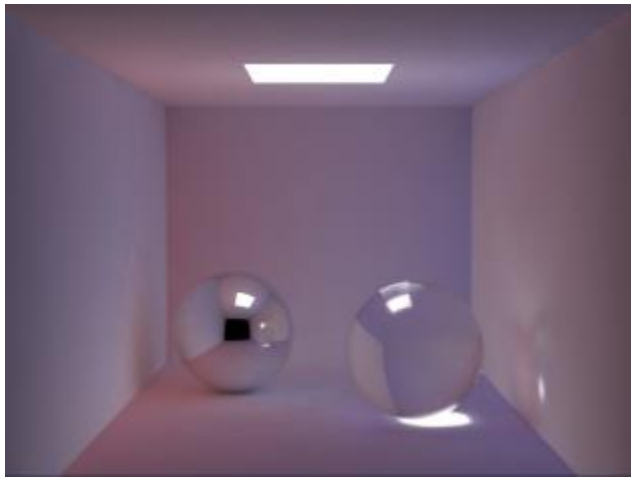
Global Illumination

=



Direct Illumination

+



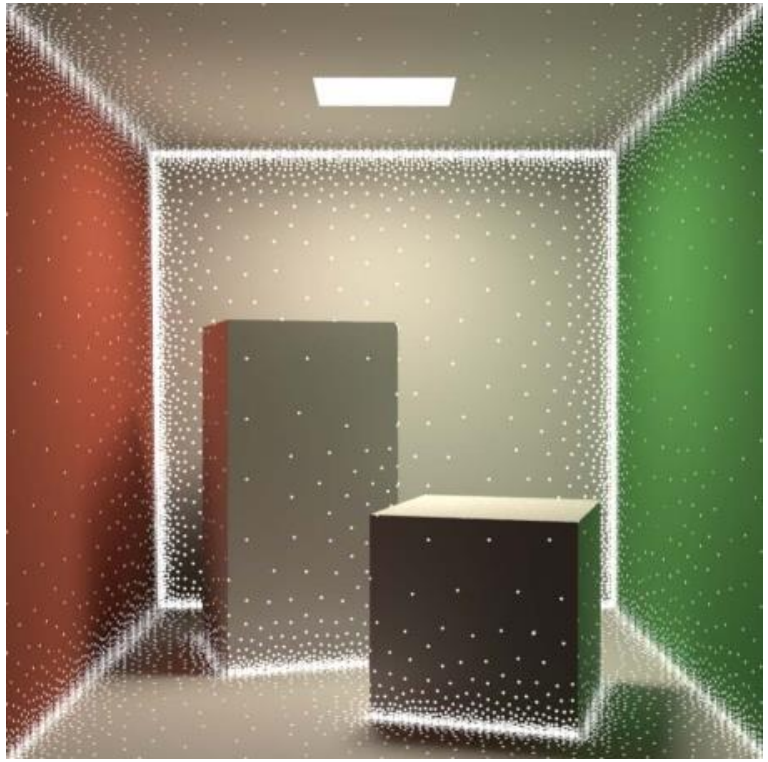
Indirect Illumination

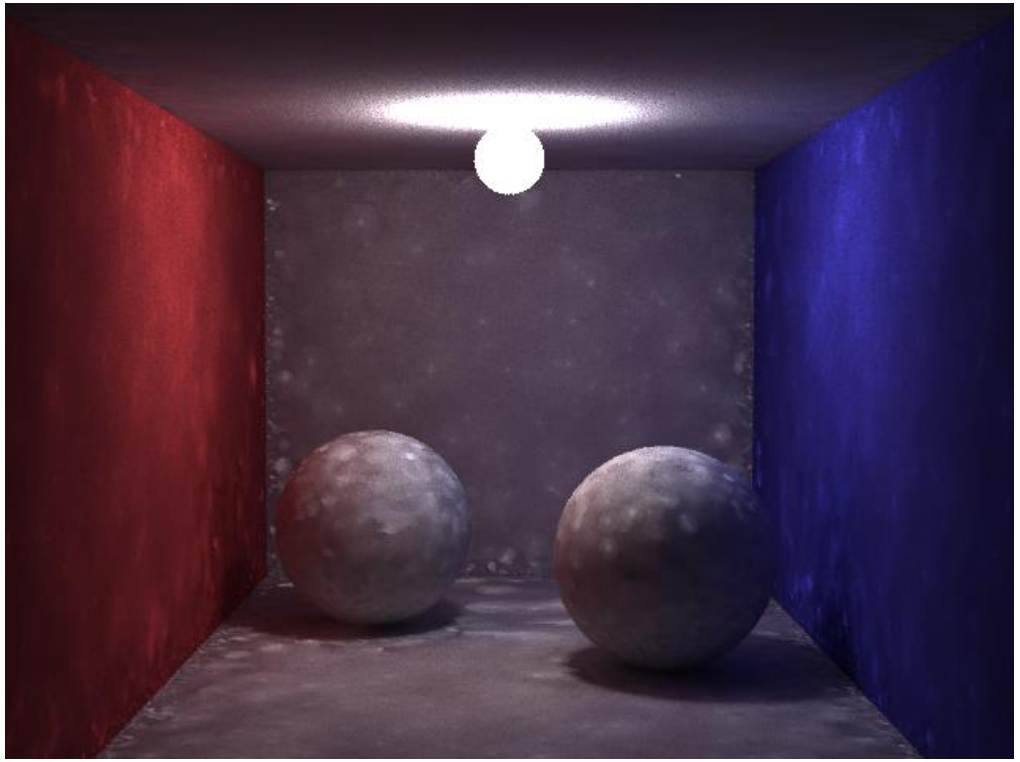
Source: https://www.cs.rpi.edu/~cutler/classes/advancedgraphics/S09/lectures/19_irradiance_caching_photon_mapping.pdf

Irradiance Caching

- Store the irradiance in a data structure.
- In pre-computation/During rendering.

Only aimed for indirect illumination!



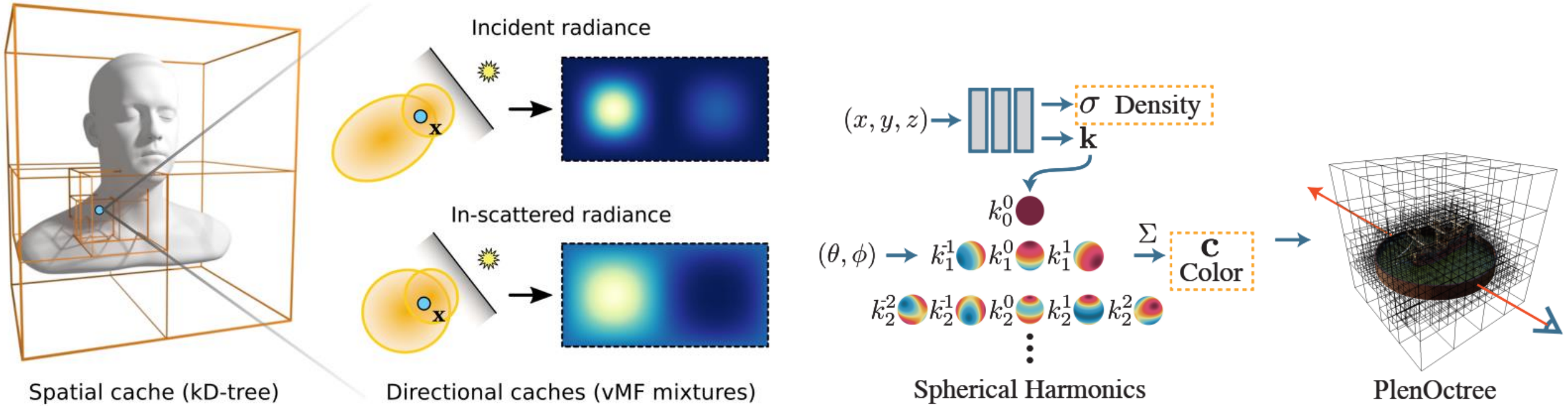


Result, raw HDR (no gamma correction).

Radiance Caching

- How about glossy surface?
- Radiative quantities feature significant **spatial**, **directional**, and **temporal** correlations.
- Using directional function: Spherical Harmonics/Spherical Gaussian.
- Photon Mapping → Hash Encoding!

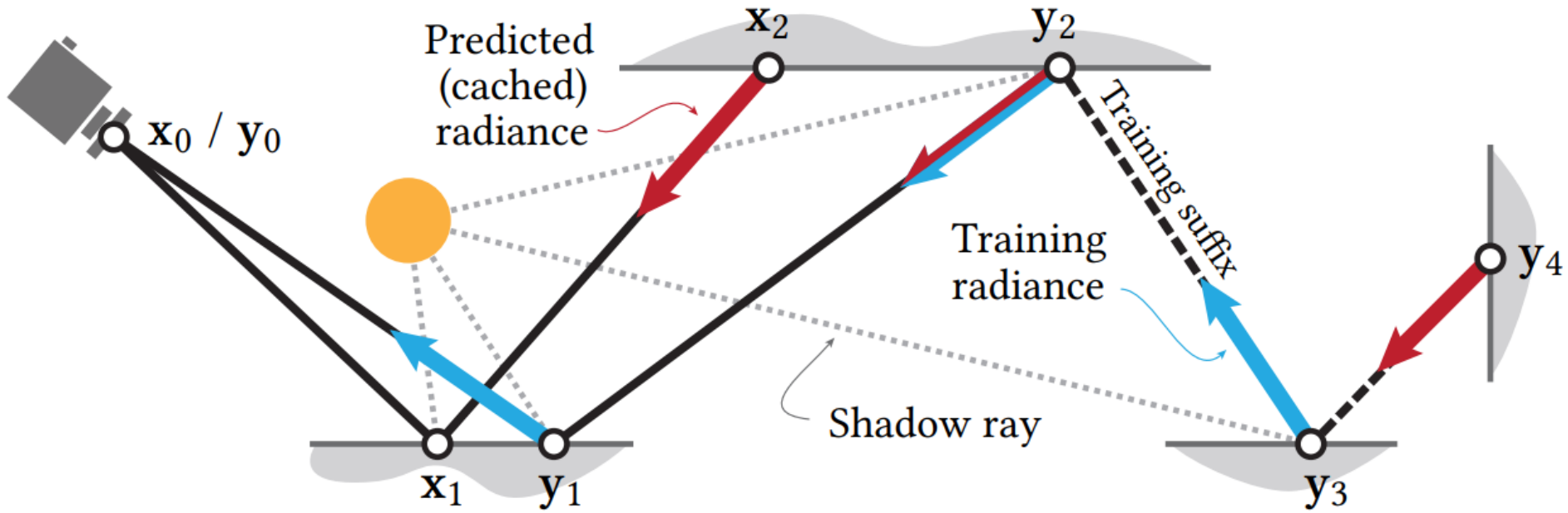
Only aimed for indirect illumination!



Source: ① Volume Path Guiding Based on Zero-Variance Random Walk Theory. Herholz et al. Siggraph 2019. ② PlenOctrees for Real-time Rendering of Neural Radiance Fields. Yu et al. ICCV 2021.

NRC – Algorithm

- Directly query indirect illumination from neural network.

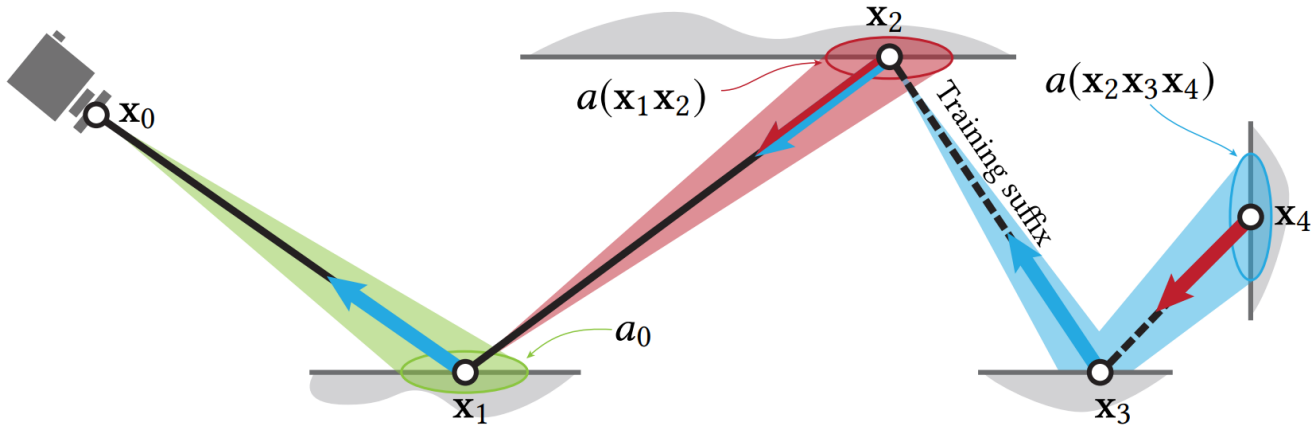


- When to terminate the path?
- Online/Offline training? How is the training strategy?

Source: Real-time Neural Radiance Caching for Path Tracing. Mueller et al. Siggraph 2021.

NRC – Path Termination

- Once the area-spread become large enough to blur away the small scale inaccuracies of the cache.



$$a_0 = \frac{\|\mathbf{x}_0 - \mathbf{x}_1\|^2}{4\pi \cos\theta_1}$$

$$a(\mathbf{x}_1 \cdots \mathbf{x}_n) = \left(\sum_{i=2}^{n} \sqrt{\frac{\|\mathbf{x}_{i-1} - \mathbf{x}_i\|^2}{p(\omega_i|\mathbf{x}_{i-1}, \omega)|\cos\theta_i|}} \right)^2$$

$$a(\mathbf{x}_1 \cdots \mathbf{x}_n) > c \cdot a_0$$

- Not all path are training path, only (2 - 3%).

Source: Real-time Neural Radiance Caching for Path Tracing. Mueller et al. Siggraph 2021.

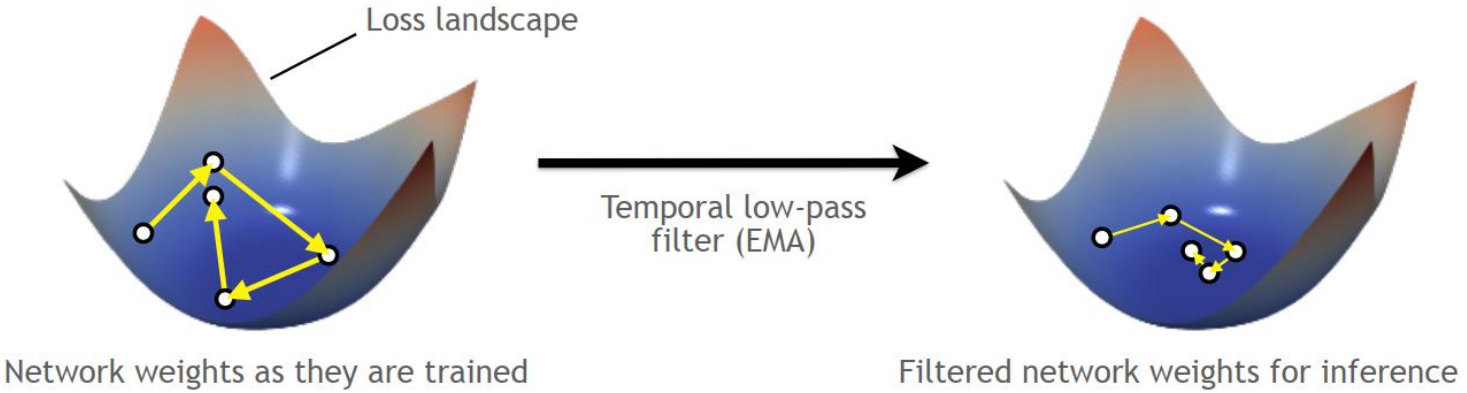
NRC – Training Strategies

- Online training.
- All vertices on training paths are training data.

- Pros for NRC:
  - Dynamic content.
  - Robustness + predictable performance and resource consumption.
  - Bias-variance tradeoff.
  - Volume rendering.
- Cons for NRC:
  - Unseen scene point.
  - Biased! (Solved by Russian Roulette)

NRC – Details

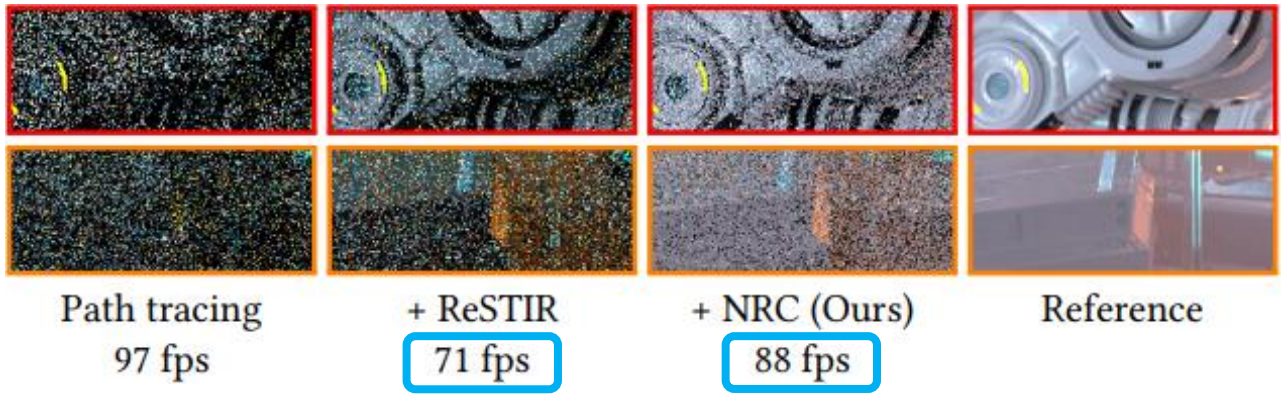
- Input Encoding:
  - $L = f(x, \omega, n, r, \alpha, \beta)$
- Rendering: Flickering for aggressive optimization schedule.
  - Temporal low pass filter - Exponential Moving Average.

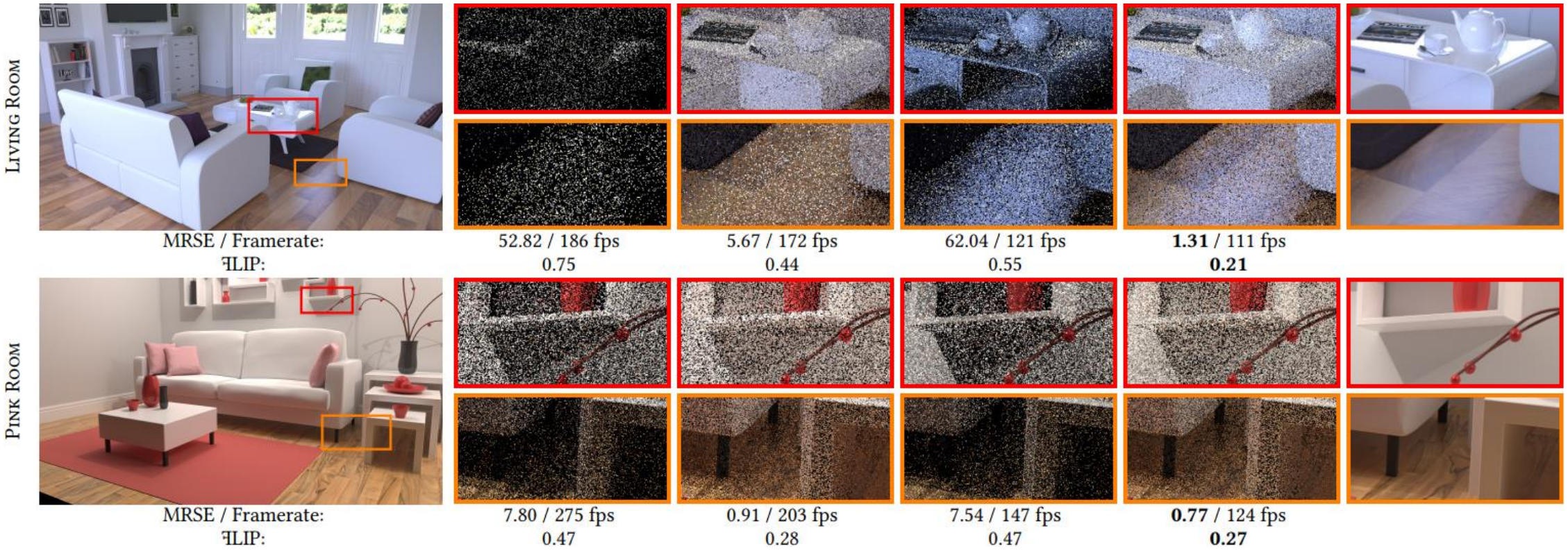


$$\bar{W}_t := \frac{1-\alpha}{\eta_t} \cdot W_t + \alpha \cdot \eta_{t-1} \cdot \bar{W}_{t-1} \quad \eta_t = 1 - \alpha^t$$

Source: Real-time Neural Radiance Caching for Path Tracing. Mueller et al. Siggraph 2021.

NRC – Results



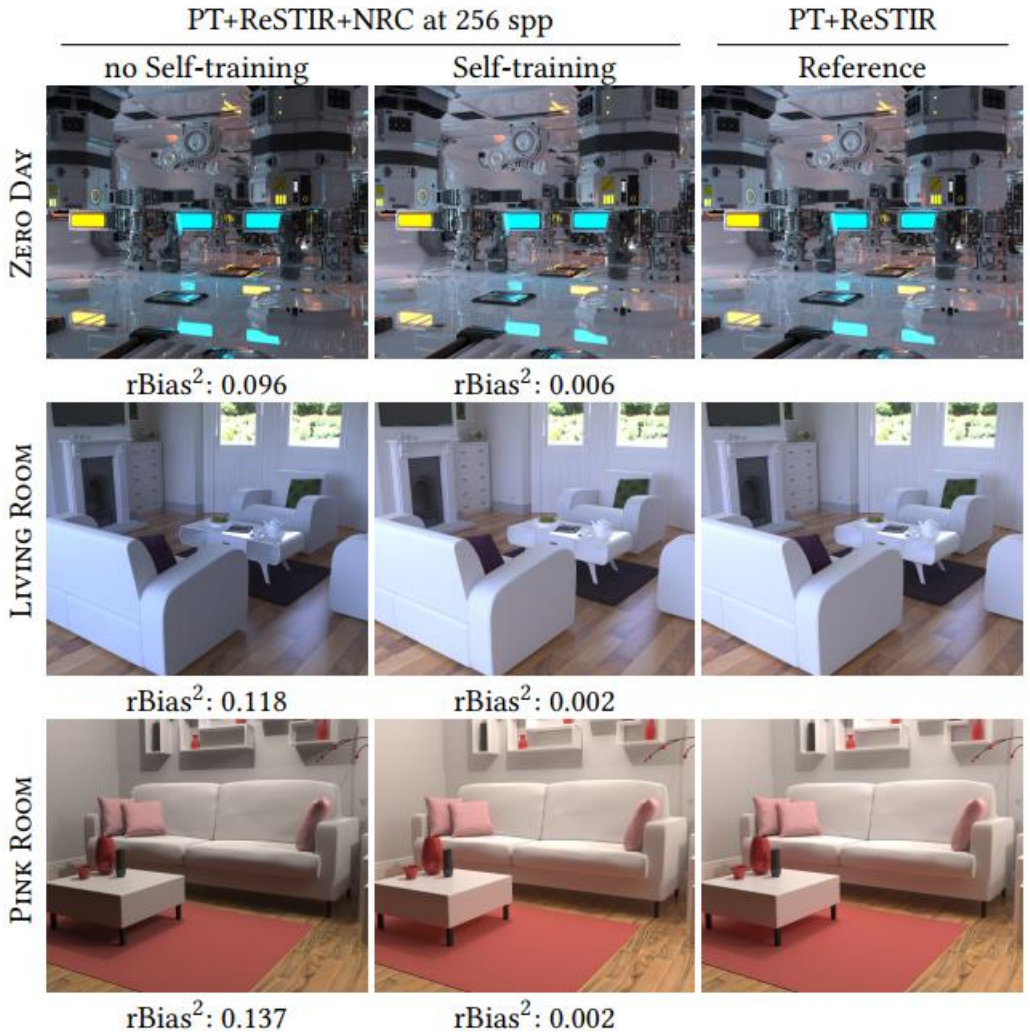


Haizhao Dai

Source: Real-time Neural Radiance Caching for Path Tracing. Mueller et al. Siggraph 2021.

NRC – Results

Scene	Method	Frames	Time	MRSE	Speedup
Attic	PT+ReSTIR	7	92.5 ms	2.739	**6.6×**
	PT+ReSTIR+NRC	1	14.0 ms	2.727	
Bistro	PT+ReSTIR	8	110.7 ms	1.498	**7.6×**
	PT+ReSTIR+NRC	1	14.6 ms	1.407	
Classroom	PT+ReSTIR	145	2625 ms	5.882	**172.7×**
	PT+ReSTIR+NRC	1	15.2 ms	5.847	
Living Room	PT+ReSTIR	53	431.5 ms	1.379	**49.6×**
	PT+ReSTIR+NRC	1	8.7 ms	1.376	
Pink Room	PT+ReSTIR	10	66.9 ms	0.769	**8.4×**
	PT+ReSTIR+NRC	1	8.0 ms	0.765	
Zero Day	PT+ReSTIR	5	69.4 ms	3.799	**6.1×**
	PT+ReSTIR+NRC	1	11.3 ms	3.430	
Average	PT+ReSTIR	16.6	154.1 ms	2.037	**13.6×**
	PT+ReSTIR+NRC	1	11.3 ms	1.941	



Source: Real-time Neural Radiance Caching for Path Tracing. Mueller et al. Siggraph 2021.

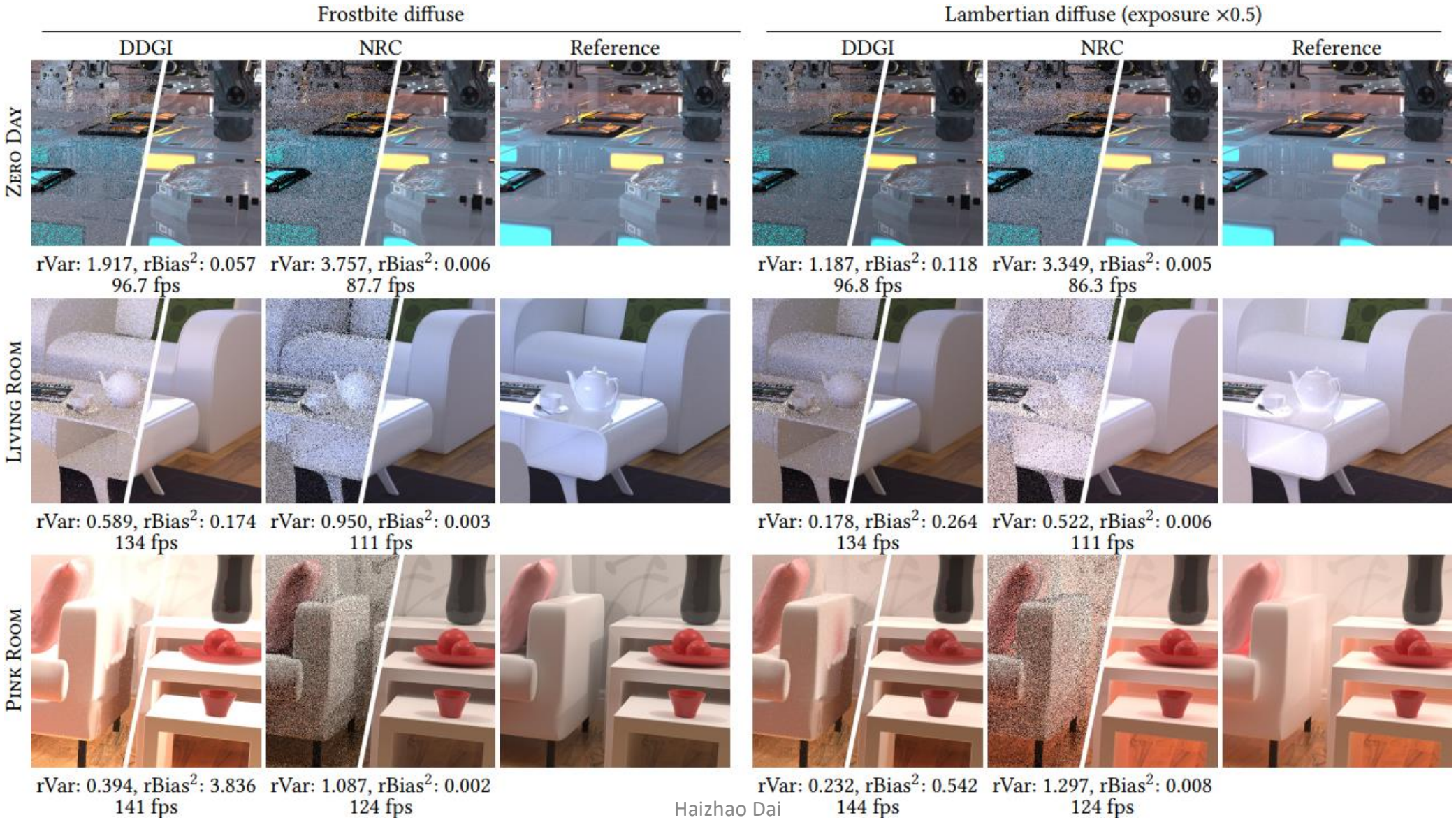
Haizhao Dai

Source: Real-time Neural Radiance Caching for Path Tracing. Mueller et al. Siggraph 2021.

Haizhao Dai

Source: Real-time Neural Radiance Caching for Path Tracing. Mueller et al. Siggraph 2021.

NRC – Results

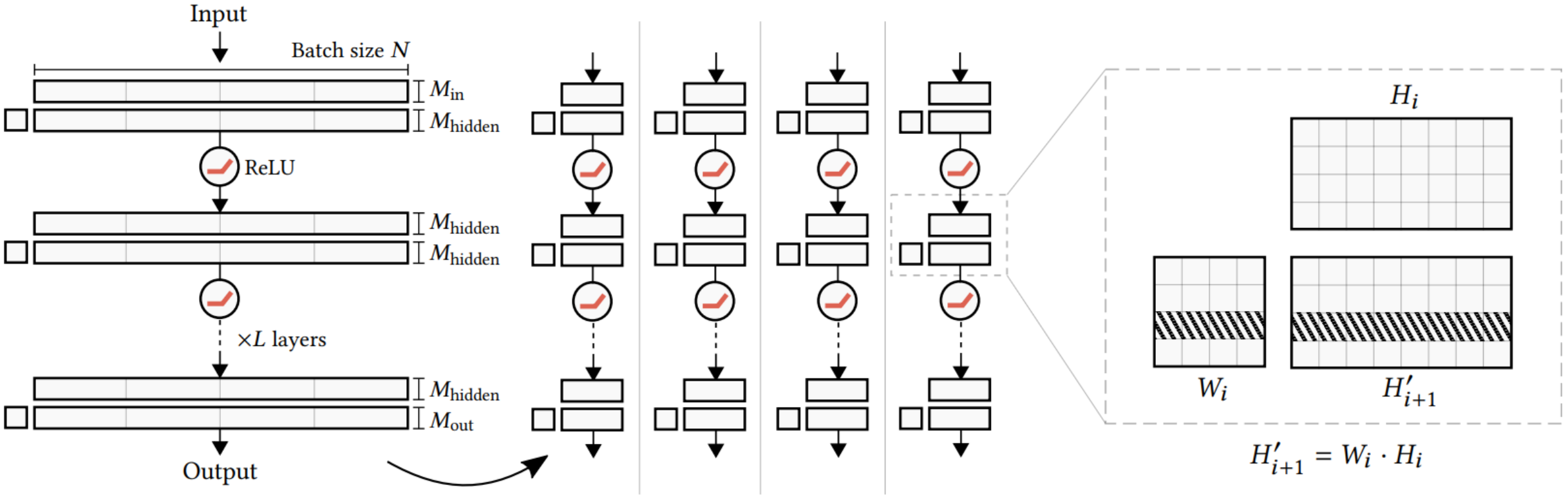


Haizhao Dai

Source: Real-time Neural Radiance Caching for Path Tracing. Mueller et al. Siggraph 2021.

Fully Fused Neural Network

- A.K.A tiny-cuda-nn. Fully fused: implement nn as a GPU kernel.



(a) Batched neural network evaluation **(b)** Distribution of a batch over thread blocks **(c)** Per-thread-block matrix multiplication

Source: Real-time Neural Radiance Caching for Path Tracing. Mueller et al. Siggraph 2021.

References

- Physically Based Rendering: From Theory To Implementation Third Edition. Pharr et al. 2018.
- Volume Path Guiding Based on Zero-Variance Random Walk Theory. Herholz et al. Siggraph 2019.
- PlenOctrees for Real-time Rendering of Neural Radiance Fields. Yu et al. ICCV 2021.
- Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. Bitterli et al. Siggraph 2020.
- Real-time Neural Radiance Caching for Path Tracing. Mueller et al. Siggraph 2021.
- Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. Mueller et al. Siggraph 2022 Best Paper.