CSE 272 Final Project: ReSTIR and ReSTIR PT for Lajolla Renderer

Long-Giang, Vu [1]

Abstract

This project implements ReSTIR (Bitterli et al., 2020) and ReSTIR PT (Lin et al., 2022) for the Lajolla Renderer. ReSTIR focuses on efficiently capturing direct lighting, making it particularly effective for rendering many-light scenes. In contrast, ReSTIR PT generalizes the algorithm to ensure convergence of the renderer while also capturing global illumination. Additionally, ReSTIR is well-suited for real-time rendering due to its optimizability, whereas ReSTIR PT demands more computational resources. However, given equivalent parameter settings, ReSTIR PT produces higher-quality rendered images than ReSTIR, and both outperform the standard path-tracing algorithm implemented in Lajolla.

1. Introduction

In scenes with multiple light sources, direct lighting poses significant challenges when modeled using Monte Carlo integration. The ReSTIR algorithm addresses these issues by introducing a novel approach to efficiently sample lights, resulting in less noisy images. Additionally, ReSTIR operates in a streaming manner and requires constant memory space, making it ideal for both online and offline rendering applications.

While empirical results demonstrate that ReSTIR improves image quality in many lighting scenarios, it theoretically lacks rigorous convergence guarantees due to the practice of reusing samples, which introduces correlation. To address these limitations, ReSTIR PT is introduced as a variant with specific implementation choices designed to extend the work to path tracing renderers, showcasing impressive improvements in image quality.

Below, I briefly present the core ideas behind ReSTIR and ReSTIR PT. The following section includes notations and parameters for easier understanding.

Rendering equation:

$$L = \int_{\Omega} f(x)dx \approx \mathbb{E}[f(Y)W_Y]$$

Where Ω is the path space of a vertex, Y is the output of a sampling algorithm and W_Y is called unbiased contribution weight of Y, estimate reciprocal PDFs.

In Monte Carlo integration, Y is sampled according to a known distribution $p(Y)$ (e.g. BSDF, NEE) and $W_Y = \frac{1}{p(Y)}$.

1.1. ReSTIR

ReSTIR tries to present a novel approach to efficiently resample Y from a set of candidates $X = \{X_1, X_2, ..., X_M\}$ which are sampled by a known distribution (e.g BSDF, NEE). The ultimate goals is to make the sample pdf $p(Y) \propto f$ and make the integral converges quickly and exactly. To do so, it exploits the resampling importance sampling from (Talbot, 2005) by introducing weight term:

$$w_i = \frac{1}{M}\hat{p}(X_i)W_i \qquad W_i = \frac{1}{p(X_i)} \tag{1}$$

where $\hat{p}(X_i)$ is the unnormalized target distribution, e.g $\hat{p} \propto f$. Output Y is selected from set X according to the weight w and has the unbiased contribution weight defined as:

$$W_Y = \frac{1}{\hat{p}(Y)} \sum_{i=1}^{M} w_i \tag{2}$$

This sampling algorithm can be performed in streaming manner by Weighted Reservoir Sampling algorithms (WRS) (Chao, 1982). In details, let M be the current number of item in the reservoir, the next X_{M+1} candidate added will be selected with the probability

$$Pr[Y = X_{M+1}] = \frac{(M+1)w_{M+1}}{(M+1)w_{M+1} + M\sum_{i=1}^{M} w_i}$$

where $\sum_{i=1}^{M} w_i$ can be easily retrieved from equation 2 without recomputing all the weights.

ReSTIR algorithm is performed by the following steps:

- For each pixel in image, initialize it's reservoir with some candidates (sampled by NEE for direct lighting).
- Add candidates from previous frame to the reservoirs (temporal reuse).

- Add candidates from neighbor pixels to current pixel (spatial reuse).
- Compute radiance for pixels.

By adding candidates from temporal frame and neighbor pixels, ReSTIR can reduce time for sampling new candidates and still having sufficient number of samples, thus render high fidelity images. Details about hyperparameters and implementation choices are described in Section 3.

1.2. ReSTIR PT

ReSTIR is built based on the assumption that independent samples X_i are in a shared domain Ω and therefore it may not retain any theoretical convergence guarantees. ReSTIR PT introduces the generalized resampled importance sampling algorithm (GRIS) allowing samples X_i from different domains and provides the theoretical analysis for the convergence.

In details, ReSTIR uses shift mapping to map every samples X_i from domains other than Ω to Ω: $T_i : \Omega_i \to \Omega$, the weight for samples X_i are now modified as bellow:

$$w_i = m_i(T_i(X_i))\hat{p}(T_i(X_i))W_i \left|\frac{\partial T_i}{\partial X_i}\right|$$

where m_i is the multiple importance sampling weight depicting the fact that each sample X_i can come from the same domain but different distributions: $\sum_i^M m_i(y) = 1$, the jacobian determinant of the shift mapping is added to transform the weight from Ω_i to Ω. W_i and W_Y are calculated same as in ReSTIR. Every terms in this new formula are in hand from ReSTIR except m_i and $\left|\frac{\partial T_i}{\partial X_i}\right|$. In the original paper (Lin et al., 2022), there are multiple choices for these two terms. However, one specific method for each term is implemented in this project will be presented below.

1.2.1. MIS

A sample $X_i \in \Omega_i$ is called canonical sample if it's domain is Ω, and it use identity shift map $T_i(x) = x$, use $\hat{p}_i = \hat{p}$. In a reservoir with M samples, denote the set of indices of canonical samples by R and their population by $|R|$. To simplify the derivation, $\hat{p}_{\leftarrow i}$ denoting unnormalized target probability of a sample $y \in \Omega$ in other domain Ω_i is defined as:

$$\hat{p}_{\leftarrow i}(y) = \begin{cases} \hat{p}_i(T_i^{-1}(y))|\partial T_i^{-1}/\partial Y_i|, & \text{if } y \in T_i(suppX_i) \\ 0 & \text{otherwise} \end{cases}$$

I use the defensive generalized pairwise MIS for this project. This option is designed eliminate singularities in weight values which can be caused by the unbounded terms in the formula (e.g too large m_i, unbouned unnormalized $\hat{p}$, ...). It also makes the canonical samples having larger weight in comparison with samples from other domains.

$$m_i(y) = \begin{cases} \frac{1}{M} + \frac{1}{M}\sum_{j \notin R} \frac{\hat{p}(y)}{|R|\hat{p}(y) + (M-|R|)p_{\leftarrow j}(y)} & \text{if } i \in R \\ \frac{M-|R|}{M} \frac{p_{\leftarrow i}(y)}{|R|\hat{p}(y)+(M-|R|)p_{\leftarrow i}(y)} & \text{if } i \notin R \end{cases}$$

1.2.2. Shift mapping

There are several choices of shift mapping built on some basic blocks: vertex copy (reconnection) (Lehtinen et al., 2013), half-vector copy (Kettunen et al., 2015), direction copy, random reply, manifold exploration (Lehtinen et al., 2013). ReSTIR PT itself experiments with two different shift mappings:

- The reconnection shift: Set $y_2 = x_2$ to always connect at the first indirect vertex.
- A hybrid of random replay and reconnection. The decision is based on the reconnection conditions.

In this project, I only use reconnection shift due to it's simplicity. The other options will be left for future improvement.

I also use solid angle parametrization (e.g (Kettunen et al., 2015)) for Jacobian computation which is different from the choice of ReSTIR PT (PSS). The Jacobian for the reconnection shift mapping is:

$$\left|\frac{\partial \omega_i^y}{\partial \omega_i^x}\right| = \left|\frac{\cos\theta_2^y}{\cos\theta_2^x}\right| \frac{\|x_{i+1} - x_i\|^2}{\|x_{i+1} - y_i\|^2}$$

where ω_i^x is the unit vector from x_i to x_{i+1}, $\theta_2^\bullet$ the angle between $\omega_i^\bullet$ and the geometric surface normal at $x_{i+1} = y_{y+1}$.

2. Desired goals

This section is to remind the desired goals and progress from the project checkpoint. The goals of this project are:

- Implement ReSTIR to better render many lights scene and make comparison with path tracing in lajolla renderer.
- With better sampling method, generate less noisy image in comparison with path tracing in the same time.
- Implement ReSTIR PT.

2.1. Progress from checkpoint

Done tasks:

- Create a test scene with numerous light sources using Blender.

- Implement ReSTIR.

Problems:

- Correlation aliasing.
- Slow rendering.

Figure 1 shows some images from checkpoint with correlation aliased images generated by ReSTIR.

3. Implementation details

This section will provide implementation details and hyper-parameters choice.

Since ReSTIR and ReSTIR PT focuses on better sampling strategies, the number of samples per pixel is fixed to 1 and 2 for all experiments to show their effectiveness (especially for comparison with vanilla path tracer).

Scenes used in experiments do not contain any raw glass-like object that mean they all reflect but refract. This condition is hold to ensure ReSTIR PT does not need to check for reconnection condition during shifting. The more complicated scenes are expected to be supported after the project deadline.

Only spatial reuse is employed in this project since lajolla does not render animations.

Neighbor pixels for both ReSTIR and ReSTIR are chosen from the neighborhood with predefined radius, sampled from a low-discrepancy sequence (Halton sequence with base 2 for the width and base 3 for the height).

3.1. ReSTIR

Unnormalized target distribution $\hat{p}(x) = f(x)$ is used for all experiments.

The reservoir combination is executed in an unbiased manner as described in (Bitterli et al., 2020).

The value choices for initial reservoir size are [16, 32] and for the number of neighbors is [1, 2, 3].

Radius of surrounding neighborhood is fixed at 30 (pixels).

3.2. ReSTIR PT

ReSTIR PT also introduces M capping technique which provides a more gentle interpretation of M rather than the number of candidates in reservoirs in ReSTIR. For simplification, M capping clamps the value of M to a predefined value whenever it goes beyond.

For all experiments, M capping for ReSTIR is fixed at 36.

Maxdepth for a light path of ReSTIR is set equivalent to it of vanilla path tracer.

Number of neighbors per pixel and neighborhood radius are set equivalent to them of ReSTIR.

4. Empirical results

4.1. Correlation alias

As mentioned in Section 2, ReSTIR implementation at project checkpoint suffers from correlation alias. The final implementation fix that by re-assigning random seed for pixels. Figure 1.

4.2. Many lights scenes with ReSTIR

I run ReSTIR to render two different scenes, both with many small light sources. Figure 1 and 2 shows the output images from both path tracer and ReSTIR for those scenes. In both scenes, ReSTIR turns out brighter and less noisy images. In the scene with the monkey in Figure 1, path tracer fails to capture details of the two monkeys with just 1 sample per pixel and maximum depth of the path equals to 2. In contrast, ReSTIR successfully renders the monkeys's face and also show nicely convergence. The final version also fixed the correlation aliasing mentioned earlier.

For the second scene, Figure 2 demonstrates images generated with different settings. With the same number of samples per pixel, images rendered with ReSTIR are brighter and more detail. In the first two images, it's impossible to locate the yellow cube on the left hand side of the bunny and also a couple of bananas right above it. ReSTIR captures those objects clearly and also add more light to the bunny's back.

Increasing number of spatial resuse (neighbor) enhance image quality of ReSTIR. Similar trend is witnessed when increase the reservoir size (increasing the number of initial lights).

5. ReSTIR PT with diffuse scene

6. Path tracer vs ReSTIR vs ReSTIR PT

References

Bitterli, B., Wyman, C., Pharr, M., Shirley, P., Lefohn, A., and Jarosz, W. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Transactions on Graphics (TOG)*, 39(4):148–1, 2020.

[1]Computer Science and Engineering Department, University of California, San Diego, CA, US.

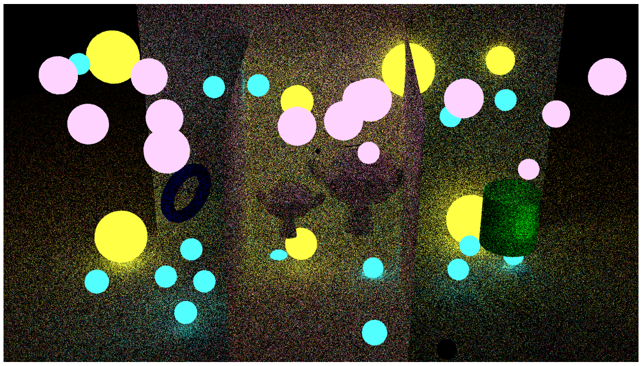
Chao, M.-T. A general purpose unequal probability sampling plan. *Biometrika*, 69(3):653–656, 1982.

Kettunen, M., Manzi, M., Aittala, M., Lehtinen, J., Durand, F., and Zwicker, M. Gradient-domain path tracing. *ACM Transactions on Graphics (TOG)*, 34(4):1–13, 2015.

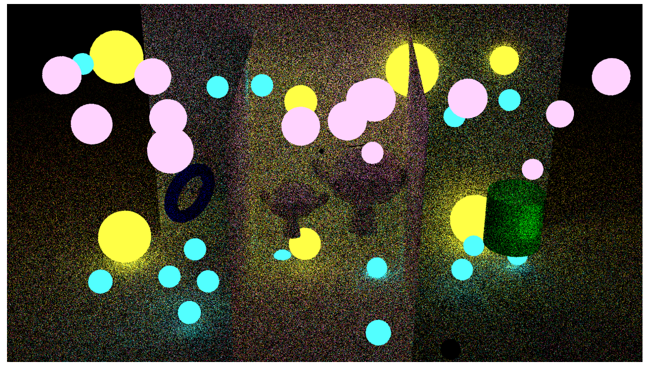
Lehtinen, J., Karras, T., Laine, S., Aittala, M., Durand, F., and Aila, T. Gradient-domain metropolis light transport. *ACM Transactions on Graphics (TOG)*, 32(4):1–12, 2013.

Lin, D., Kettunen, M., Bitterli, B., Pantaleoni, J., Yuksel, C., and Wyman, C. Generalized resampled importance sampling: Foundations of restir. *ACM Transactions on Graphics (TOG)*, 41(4):1–23, 2022.

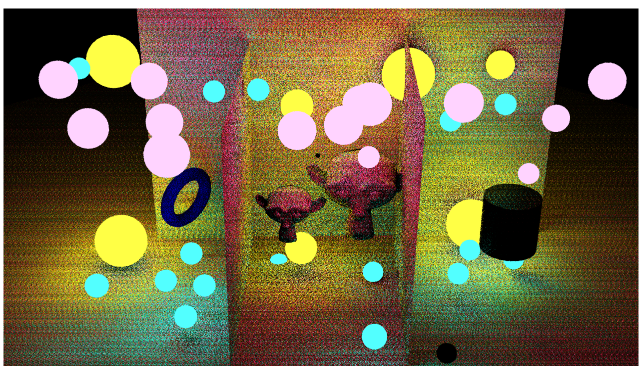
Talbot, J. F. *Importance resampling for global illumination*. Brigham Young University, 2005.



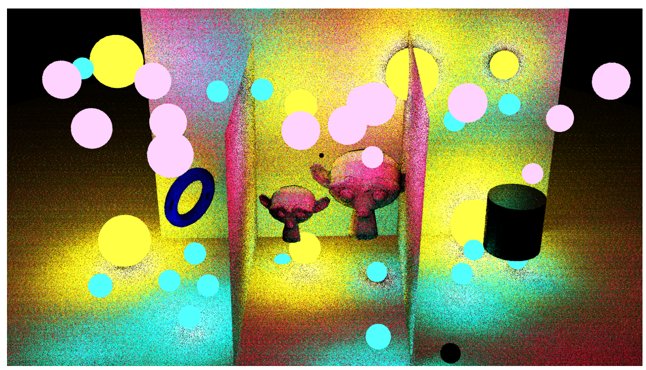
(a) Path tracer (max_depth=2, spp=1)



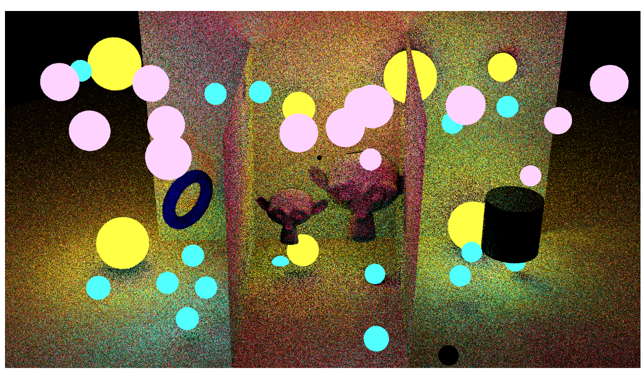
(b) Path tracer (max_depth=10, spp=1)



(c) ReSTIR (neighboor=1, spp=1)

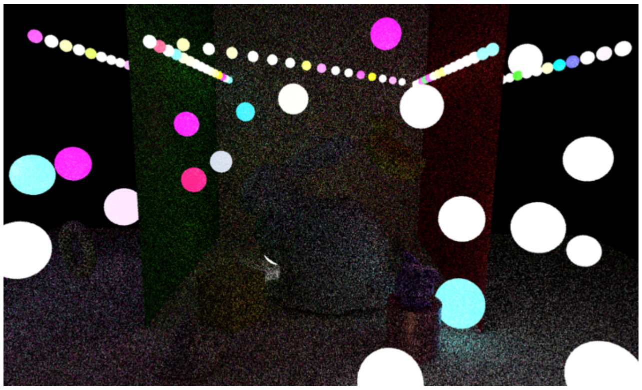


(d) ReSTIR (neighboor=5, spp=1)

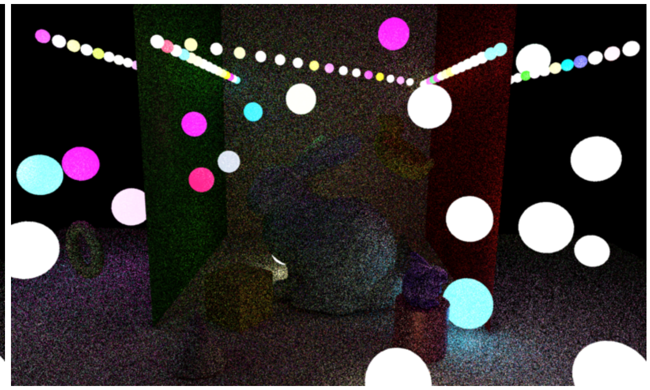


(e) Fixed ReSTIR with neighbor=1, spp=1

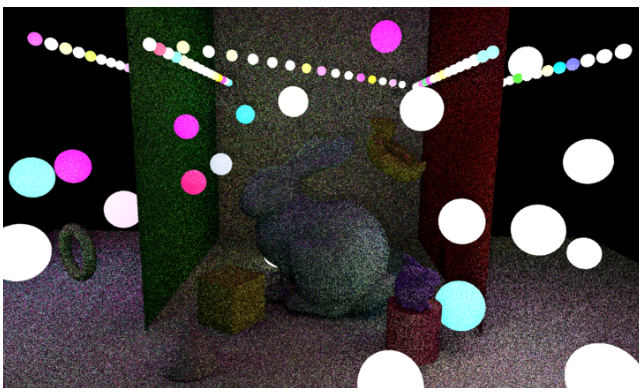
Figure 1. Result from project checkpoint. ReSTIR suffered from correlation aliasing.



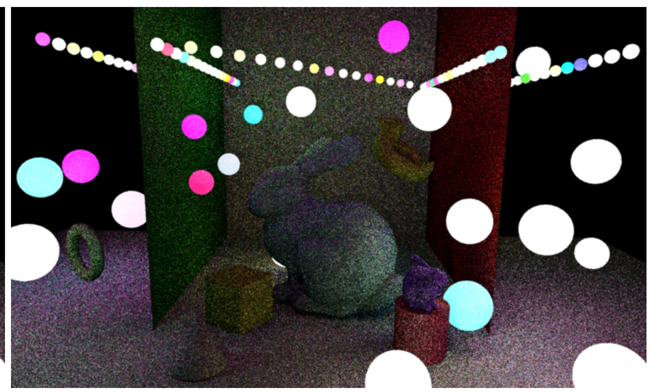
(a) Path tracer (spp=1, depth=1)



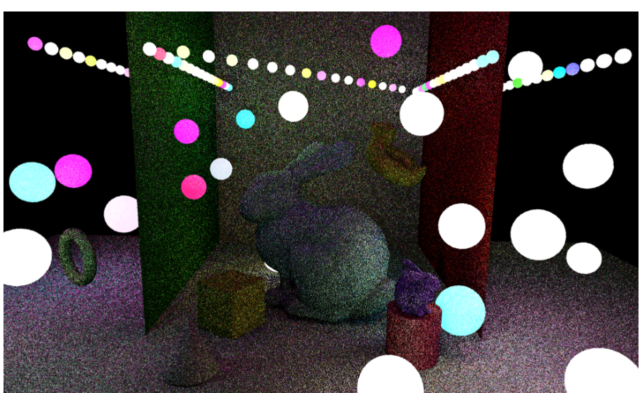
(b) Path tracer (spp=2, depth=1)



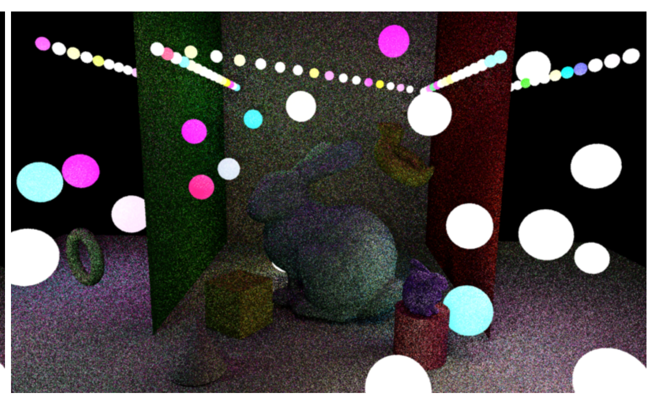
(c) ReSTIR (spp=1, reservoirs_size=16, neighbor=1)



(d) ReSTIR (spp=2, reservoirs_size=16, neighbor=1)

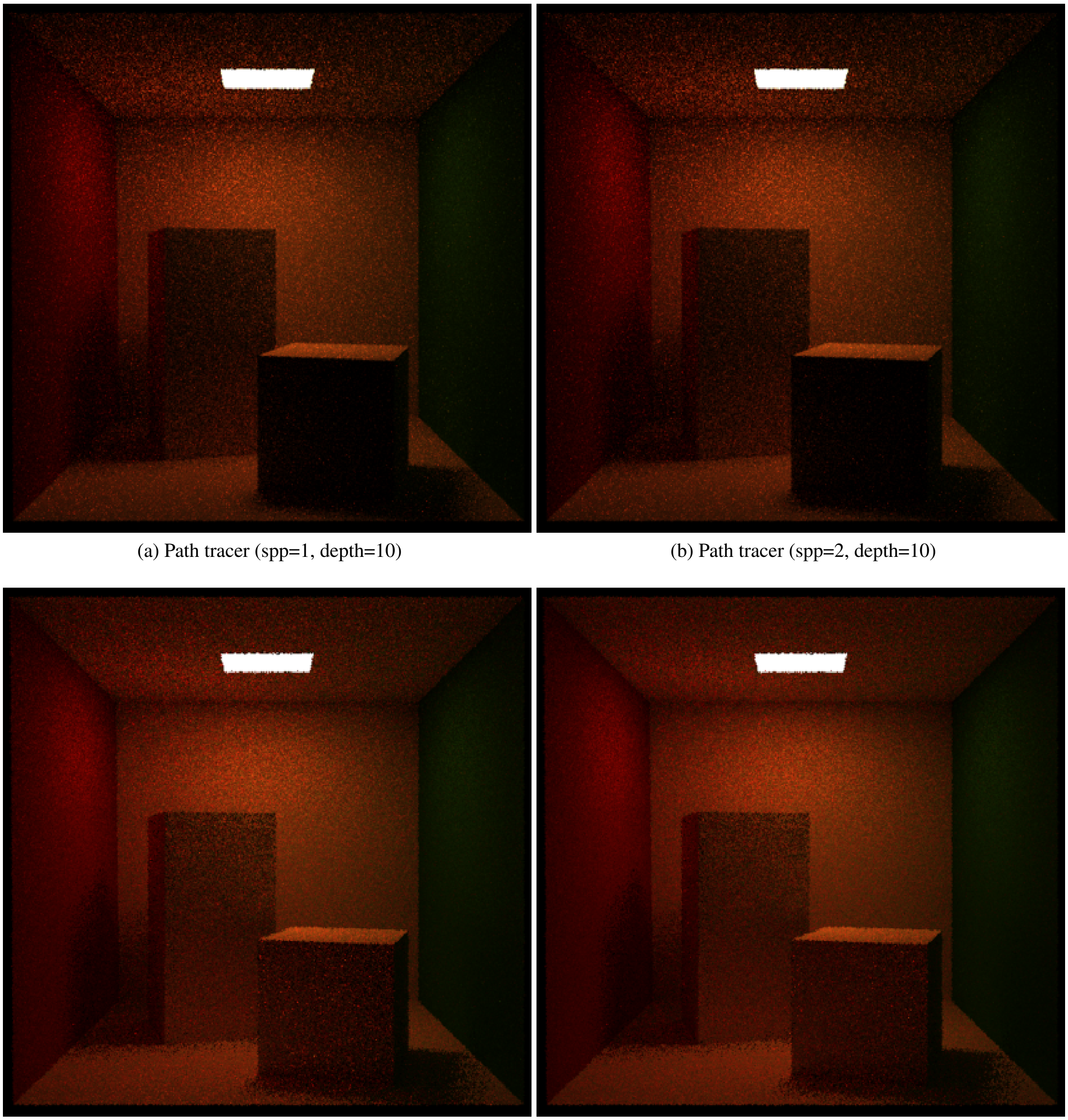


(e) ReSTIR (spp=1, reservoirs_size=32, neighbor=1)



(f) ReSTIR (spp=2, reservoirs_size=32, neighbor=1)

Figure 2. **Images rendered from the same scene by path tracer (first row), ReSTIR algorithm with reservoir_size=16 (second row) and ReSTIR with reservoir_size=32 (last row).**



(a) Path tracer (spp=1, depth=10)

(b) Path tracer (spp=2, depth=10)

(c) ReSTIR PT (spp=1, depth=10, reservoir_size=16, neighbor=1)

(d) Path tracer (spp=2, depth=10, reservoir_size=32, neighbor=1)

Figure 3. **Cbox rendered by path tracer and ReSTIR PT**

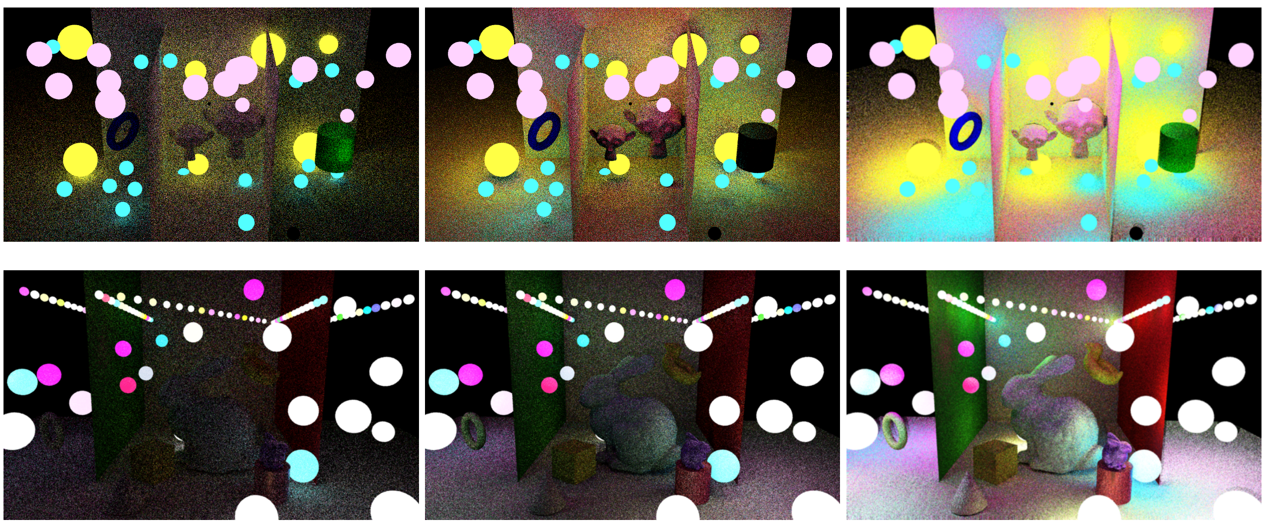


Figure 4. Image rendered from two scenes by **Path Tracer (first column)**, **ReSTIR (second column)** and **ReSTIR PT (last column)**