Rendering Participating Media Using Path Graphs

BECKY HU, Cornell University, USA
XI DENG, Cornell University, USA
FUJUN LUAN, Adobe Research, USA
MILOŠ HAŠAN, Adobe Research, USA
STEVE MARSCHNER, Cornell University, USA

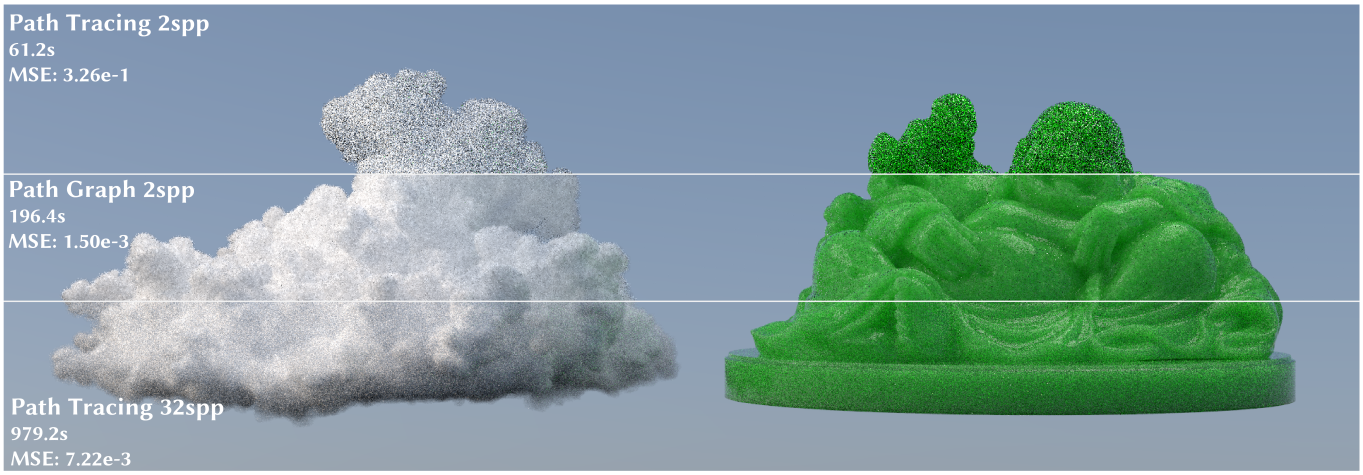

Fig. 1. A heterogeneous cloud, left, and a homogeneous buddha statue with a dielectric interface, right, both illuminated by a sun-sky model and an area light. At the top we show an equal-sample comparison between path tracing and our method (path tracing + path graph); our method reduces variance significantly. At the bottom we show that using our method with a very small number of samples can both take less time and produce a more accurate result than using a larger number of samples with path tracing.

Rendering volumetric scattering media, including clouds, fog, smoke, and other complex materials, is crucial for realism in computer graphics. Traditional path tracing, while unbiased, requires many long path samples to converge in scenes with scattering media, and a lot of work is wasted by paths that make a negligible contribution to the image. Methods to make better use of the information learned during path tracing range from photon mapping to radiance caching, but struggle to support the full range of heterogeneous scattering media. This paper introduces a new volumetric rendering algorithm that extends and adapts the previous *path graph* surface rendering algorithm. Our method leverages the information collected through multiple-scattering transport paths to compute lower-noise estimates, increasing computational efficiency by reducing the required sample count. Our key contributions include an extended path graph for participating media and new aggregation and propagation operators for efficient path reuse in volumes. Compared to previous methods, our approach significantly boosts convergence in scenes with challenging volumetric light transport, including heterogeneous media with high scattering albedos and dense, forward-scattering translucent materials, under complex lighting conditions.

CCS Concepts: • **Computing methodologies** → **Ray tracing**.

Additional Key Words and Phrases: Rendering, volume rendering, raytracing, global illumination

ACM Reference Format:
Becky Hu, Xi Deng, Fujun Luan, Miloš Hašan, and Steve Marschner. 2024. Rendering Participating Media Using Path Graphs. 1, 1 (April 2024), 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Authors' addresses: Becky Hu, Cornell University, USA; Xi Deng, Cornell University, USA; Fujun Luan, Adobe Research, USA; Miloš Hašan, Adobe Research, USA; Steve Marschner, Cornell University, USA.

XXXX-XXXX/2024/4-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Volumetric scattering media (such as clouds, fog, smoke, juice, soap, or marble) are common in real-world scenes, and accurate rendering of such materials is key to realism in scenes with scattering atmospheres such as clouds, fog, or smoke; turbid fluids like juice, milk, or seawater; or translucent solids like marble. However, simulating light transport in participating media presents well-known challenges: compared to scenes with only surfaces there is an increased number of scattering events, especially in media with anisotropic phase functions, and it is challenging to sample fast-varying optical properties, especially when optical density varies spectrally.

Monte Carlo path tracing is widely used for rendering scenes with participating media; while unbiased, it requires a large number of samples to converge. High sample counts become particularly problematic when artists need fast turnaround during the scene design process. Denoisers improve convergence to some extent; however, better sampling methods with lower variance are desirable independent of whether denoising is used. The high-level goal of this paper is to extract as much information as possible from a very small number of volumetric scattering paths so that we can obtain useful images quickly and ultimately achieve faster convergence.

Participating media are characterized by long multiple-scattering transport paths. By tracing these paths through the volume, a path tracer collects a lot of valuable information about the scene, but then computes a single pixel contribution and discards all the other information. While this has the benefit of requiring minimal storage, it clearly feels suboptimal in terms of computational efficiency, and many methods have been proposed to make better use of the traced paths. This path information can be stored in the scene and then gathered by density estimators, including photon points [Jensen 1996; Jensen and Christensen 1995], photon beams [Jarosz et al. 2008a], and photon primitives [Bitterli and Jarosz 2017; Deng et al. 2019]; weighted by multiple importance sampling [Veach and Guibas 1995] to combine different sampling techniques [Georgiev et al. 2012b; Hachisuka et al. 2012; Křivánek et al. 2014]; or cached to guide smarter future sampling [Bitterli et al. 2020; Herholz et al. 2019; Lin et al. 2021]. However, the reuse of information in these methods is constrained to be relatively local, among spatially close paths or paths from neighboring pixels. We would prefer more global information sharing, especially in scenes with complex, long paths where the information is hidden behind multiple bounces or in scenes dominated by indirect illumination where the majority of the paths fail to find the light source. In fact, such cases are ubiquitous in volumetric scenes, such as subsurface-scattering materials with dielectric boundaries, or clouds and smoke with challenging lighting conditions and high single-scattering albedos.

Recent work on path graphs [Deng et al. 2021] has shown the benefits of sharing information globally across all traced paths in scenes dominated by indirect illumination. As illustrated in Fig. 2, the path graph framework records the information of all sampled paths during the path sampling process and builds a graph over these paths. Iterative refinement (through aggregation and propagation) is then applied to the path graph to refine the radiance returned from each path. This approach is beneficial because it not only shares radiance among spatially nearby shading points, but also iteratively propagates the updated radiance along paths to refine the estimates at distant shading points, making extensive use of the initial sampled paths. However, this technique is limited to surfaces.

Our goal is to develop a new method based on the path graph framework that applies to volumetric scattering. This requires several changes to the framework, since the original path graph method relies on surface shading points and related concepts such as normals, BRDFs, and their importance sampling pdfs. Furthermore, light extinction (transmittance) is unique to volumes and introduces important subtleties when extinction varies spectrally. We show how the framework can be extended to volumetric rendering and demonstrate that the resulting method significantly improves convergence across a variety of scenes with challenging volumetric transport, including heterogeneous media with high scattering albedos, subsurface scattering materials, and challenging lighting conditions.

2 RELATED WORK AND BACKGROUND

2.1 Volumetric rendering

Volumetric rendering is typically accomplished by solving the radiative transfer equation (RTE) [Kajiya and Kay 1989] using Monte Carlo estimation. Naive path tracing is the most popular rendering algorithm for participating media due to its unbiased nature and applicability to a wide variety of volumes. However, naive path tracing suffers from a low convergence rate in challenging lighting conditions, anisotropic scattering properties, and heterogeneous optical properties.

To address the low convergence rate issue, path samples can be reused. Bidirectional path tracing [Lafortune and Willems 1993; Veach 1997; Veach and Guibas 1994] (BPT) samples paths from both the camera and light sources, then connects the sub-paths and combines them using multiple importance sampling [Veach and Guibas 1995]. Paths can be also reused to improve sampling techniques. Volumetric path guiding [Herholz et al. 2019] caches paths to estimate adjacent transport solutions and uses them to navigate future sampling. The idea of spatiotemporal reservoir resampling [Bitterli et al. 2020] has also been applied in fast volume rendering [Lin et al. 2021].

Methods that reuse path samples by connecting paths sampled from sensors and from sources using photon density estimators, such as photon mapping [Jensen 1996; Jensen and Christensen 1998], are more efficient, although they introduce bias from blurring kernels. To mitigate this bias, researchers have proposed techniques like photon beams, sensor beams [Jarosz et al. 2008a,b]. Similarly to BPT, research has sought to combine these density estimators with each other and with unbiased techniques [Georgiev et al. 2012a; Hachisuka et al. 2012; Křivánek et al. 2014]. Over time, unbiased photon density estimators like photon planes, photon volumes [Bitterli and Jarosz 2017], and even photon surfaces [Deng et al. 2019] have been developed. However, the effectiveness of these unbiased density estimators is limited to homogeneous volumes.

Neural networks have been used for fast prediction in participating media. For instance Hu et al. [Hu et al. 2023] applied radiance-predicting neural networks to store directional light transmittance. However, this approach is limited to specific lighting conditions.

2.2 Path Reuse Techniques

Beyond volumetric rendering, many path reuse technique has been applied in surface rendering scenario. Early work in irradiance caching [Ward et al. 1988] stores the irradiance estimates for future interpolation of efficient indirect illumination. [Křivánek et al. 2008] extends to cache radiance allowing indirect lighting computation with the presence of glossy surface. [Keller et al. 2014] directly computes weighted average of from noisy radiance estimates on the path in Monte Carlo path tracing. [West et al. 2020] later extends the work by applying continuous multiple important sampling in combining the noise irradiance estimates, [Deng et al. 2021] further improved the efficiency of filtering using clusters for aggregation and iteratively propagate the updated refinement from indirect bounce to pixel.

3 METHOD

In this section, we will briefly review the key ideas and concepts of the path graph framework (Sec. 3.1), and derive our path graph operators for participating media starting from standard radiative transfer equations (Sec. 3.2-Sec. 3.6).

3.1 Path Graph Framework

In the rendering framework of path graphs (Fig. 2), the path tracing phase is instrumented to record enough intermediate results to recompute the pixel value given a change to any outgoing radiance value along the path. The key idea is to update these intermediate results in an iterative process where nearby shading points exchange information and then recompute the radiance values from the updated values. We think of these intermediate values as variables attached to the vertices and edges of a graph in which every shading point is a vertex and the edges record all the relevant relationships between vertices including which values are used to update which other values and which points are "nearby" so that they exchange information.

In the following context, we refer to a path graph $G = \langle V, E\rangle$ as an information sharing data structure built upon a collection of complete light paths which are sampled by path tracing of a full resolution image at one sample per pixel. V is the union of all the light points V_y and shading points V_x. The vertices are connected to each other by three different types of edges in E: continuation edges E_C, light edges E_L and neighbor edges E_N. A *continuation edge* $\mathbf{e}_C \in E_C$ stands for the propagation of a radiance value from one shading point to another; the outgoing value on a point becomes the incoming value on another. A *light edge* $\mathbf{e}_L \in E_L$ connects a light point with a shading point. It propagates emitted light from source points to surface points. A *neighbor edge* $\mathbf{e}_N \in E_N$ connects shading points with other shading points in their spatial neighborhood (cluster).

The construction of light edges and continuation edges is done simultaneously with path tracing. The neighbor edges can be constructed by a simple nearest neighbor clustering of all shading points. Given N shading points and a desired number of shading points per a cluster, K, approximately N/K shading points are chosen as cluster centers, and all other shading points are assigned to the clusters with the nearest cluster centers. This process is repeated for large clusters until the number of shading points per cluster is relatively even.

Shading points within a cluster are assumed to share the same incoming radiance distribution, and the neighbor edges serve as bridges for aggregation across those shading points. We treat the neighbors' incoming radiance samples as samples of the point's incoming radiance. Therefore, intuitively, we can treat each of the N points in the cluster as a sampling technique with its own pdf, and combine the N techniques using multiple importance sampling. The resulting improved estimates can further be used to improve estimates in other clusters that depend on them, until convergence

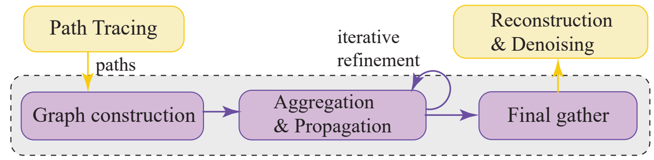

Fig. 2. Our method fits between the traditional steps of path tracing and denoising. This means that it can provide additional benefits on top of (rather than replacing) techniques like neural denoisers or advanced path sampling. Unlike the surface Path Graph, the final gather is optional in the volume case. The final gather serves to remove correlations between neighboring pixels, which are very subtle in volumes since path vertices are scattered in the volume.

is reached. In the following sections, we turn these intuitions into a precise estimator for volume scattering media.

3.2 Radiative Transfer Equation in a Medium

We will start with the radiative transfer equation (RTE) in a medium (Sec. 3.2) and review the Monte Carlo solution to the RTE (Sec. 3.3). We then derive the path graph propagation and aggregation operators that apply to participating media.

The radiance propagating in direction ω at a point $\mathbf{x}$ in the medium can be written as

$$L(\mathbf{x},\omega) = \tau(\mathbf{x},\mathbf{y})L_e(\mathbf{y},\omega) + \int \tau(\mathbf{x},\mathbf{x}')\left(L_v(\mathbf{x}',\omega) + \int_{S^2} f_\rho(\mathbf{x}',\omega,\omega')L(\mathbf{x}',\omega')\mathrm{d}\omega'\right)\mathrm{d}\mathbf{x}', \quad (1)$$

where $f_\rho(\mathbf{x}',\omega,\omega') = \sigma_s(\mathbf{x}')\rho(\mathbf{x}',\omega,\omega')$, $\mathbf{y}$ is the visible surface point in the direction $-\omega$ from $\mathbf{x}$ and the outer integral is along the line segment joining $\mathbf{x}$ and $\mathbf{y}$. $L_e(\mathbf{y},\omega)$ is the surface emitted radiance at $\mathbf{y}$, $L_v(\mathbf{x}',\omega)$ is the volume emitted radiance per unit length at $\mathbf{x}'$, $\rho(\mathbf{x}',\omega,\omega')$ is the phase function of the medium at $\mathbf{x}'$, $\tau(\mathbf{x},\mathbf{y}) = e^{-\int_{\mathbf{x}}^{\mathbf{y}}\sigma_t(\mathbf{x}')\mathrm{d}\mathbf{x}'}$ is the transmittance between two points, and $\sigma_t(\mathbf{x})$ and $\sigma_s(\mathbf{x})$ are the attenuation and scattering coefficients of the medium.

For compactness of notation we encapsulate the two nested integrals in Eq. (1) inside linear operators $\mathcal{K}$ and $\mathcal{G}$:

$$[\mathcal{K}f]\,(\mathbf{x},\omega) = \int_{S^2} f_\rho(\mathbf{x},\omega,\omega')f(\mathbf{x},\omega')\mathrm{d}\omega' \quad (2)$$

$$[\mathcal{G}f]\,(\mathbf{x},\omega) = \int \tau(\mathbf{x},\mathbf{x}')f(\mathbf{x},\omega)\mathrm{d}\mathbf{x}'. \quad (3)$$

so that Eq. (1) reads

$$L = Q + \mathcal{G}(L_v + \mathcal{K}L). \quad (4)$$

where $Q(\mathbf{x},\omega) = \tau(\mathbf{x},\mathbf{y})L_e(\mathbf{y},\omega)$. This operator notation is inspired by Arvo [1993].

To align our transport equations with the quantities stored in the path graph, we rearrange the RTE in two ways. First, we introduce scattered radiance $\bar{L}$ and write Eq. (4) in two steps as:

$$\begin{cases} \bar{L} = \mathcal{K}L \\ L = Q + \mathcal{G}(L_v + \bar{L}). \end{cases} \quad (5)$$

Like volume emission, $\overline{L}$ is a radiance per unit length with units $W/(m^3 \cdot sr)$, whereas L has units $W/(m^2 \cdot sr)$.

Second, we write the radiance L as a sum of $L^{\mathbf{o}} = Q + \mathcal{G}L_v$, which is incoming radiance due to direct illumination and volume emission, and $L^+ = \mathcal{G}\overline{L}$, the radiance due to multiple scattering. We also introduce corresponding scattered quantities $\overline{L^{\mathbf{o}}} = \mathcal{K}L^{\mathbf{o}}$ and $\overline{L^+} = \mathcal{K}L^+$. The RTE then reads

$$\begin{cases} \overline{L^+} = \mathcal{K}L^+ \\ L^+ = \mathcal{G}\overline{L^{\mathbf{o}}} + \mathcal{G}\overline{L^+} \end{cases} \tag{6}$$

In this pair of equations $\overline{L^{\mathbf{o}}}$ is known (it is an integral of known quantities) and L^+ is the unknown to solve for, after which the complete solution is simply $L = L^{\mathbf{o}} + L^+$.

The steps of the path graph algorithm will correspond to these operators; Eq. (2) corresponds to the aggregation step and Eq. (3) corresponds to the propagation step. One aggregation step is used to compute $\overline{L^{\mathbf{o}}}$; a series of alternating aggregations and propagations serve to solve for L^+; and a final addition computes L for the pixels of the image.

In the next subsection, we will review the general form of Monte Carlo estimators for $\mathcal{K}$ and $\mathcal{G}$, then show how the path graph algorithm efficiently approximates these operators from a limited number of sampled paths.

3.3 Monte Carlo Estimator for RTE

Consider a general integral $I = \int_\Omega f(z)\mathrm{d}z$ on the space Ω. Let $z_1, z_2 \cdots z_N$ be random sample on Ω drawn from a probability density function $p(z)$, and assume $p(z)$ is nonzero whenever $f(z)$ is nonzero. One can prove that $\xi(z) = \frac{f(z)}{p(z)}$ is an unbiased estimator of I, because

$$E[\xi(z)] = \int_\Omega \frac{f(z)}{p(z)} p(z)\mathrm{d}z = \int_\Omega f(z)\mathrm{d}z = I. \tag{7}$$

Applying the Monte Carlo estimator Eq. (7) to the operator $\mathcal{K}$ defined in Sec. 3.2, we find that the scattered quantities $\overline{L^{\mathbf{o}}}$ and $\overline{L^+}$ can be estimated by a sample drawn from the solid angle space with probability density $p_w(\omega)$, ideally equal to the phase function (which causes cancellation):

$$\overline{L^{\mathbf{o}}}(\mathbf{x}', \omega) \approx \frac{f_\rho(\mathbf{x}', \omega, \omega')L^{\mathbf{o}}(\mathbf{x}', \omega')}{p_w(\omega')}, \tag{8}$$

$$\overline{L^+}(\mathbf{x}', \omega) \approx \frac{f_\rho(\mathbf{x}', \omega, \omega')L^+(\mathbf{x}', \omega')}{p_w(\omega')}. \tag{9}$$

The right sides are unbiased estimators, not just approximations. Similarly, to apply the Monte Carlo estimator to the operator $\mathcal{G}$, we take a sample $\mathbf{x}'$ along the direction ω from $\mathbf{x}$:

$$L^+(\mathbf{x}, \omega) \approx \frac{\tau(\mathbf{x}, \mathbf{x}')\left(\overline{L^+}(\mathbf{x}', \omega) + \overline{L^{\mathbf{o}}}(\mathbf{x}', \omega)\right)}{p_t(\mathbf{x}'|\mathbf{x})}, \tag{10}$$

where again the right side is an unbiased estimator, $p_t(\mathbf{x}'|\mathbf{x})$ is the probability density function of sampling distance $||\mathbf{x} - \mathbf{x}'||$ from $\mathbf{x}$, ideally proportional to $\tau(\mathbf{x}, \mathbf{x}')$. The Monte Carlo estimators turn the continuous operators from Sec. 3.2 to discrete samples from continuous distributions. In the following section, we will briefly review volumetric path tracing, which is nothing but recursive application of the above estimators, and introduce the construction of a path graph from the resulting path samples.

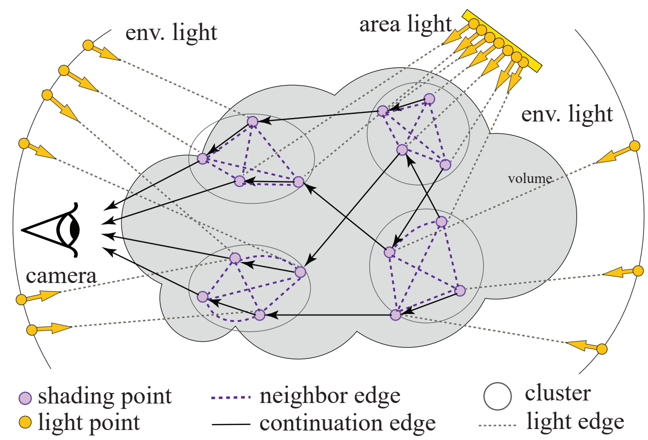

Fig. 3. Illustration of the volume path graph construction from paths sampled during a standard volumetric path tracing pass with next event estimation in participating media.

3.4 Volume Path Tracing

With Eq. (8), Eq. (9) and Eq. (10) one can estimate the integral by taking discrete random samples and continue by recursively expanding the indirect incoming radiance term as another Monte Carlo estimate until the emission is queried. A consecutive chain of shading points $\mathbf{x}_0 \cdots \mathbf{x}_k$ are sampled, with each point representing a scattering event or a surface event. Typically, each shading point $\mathbf{x}_i$ has a connection to an emission point $\mathbf{y}_i$ due to next event estimation, completing a light path. When a direction is sampled in proportion to the local phase function, it may also create a connection to light source if there is an unoccluded emission along the direction. The pixel value is computed from the contribution of those complete light paths. In standard volume path tracing, all intermediate path samples are dropped once the contribution is computed, but in path graph framework, we record enough information to reconstruct these complete light paths and build the information sharing graph.

3.5 Aggregation and Propagation

After a path graph is constructed, the next step is the aggregation of information across neighbor edges and propagation along continuation edges. We use $i = 0 \cdots D$ to index shading points on a complete light path $\bar{\mathbf{z}}$ with D being max depth of the path, and $j = 1 \cdots K$ to index the shading points within a cluster, with K being the number of shading points in a cluster.

For a shading point $\mathbf{x}_i$ in a cluster C, we aggregate direct and indirect radiance estimates from all neighbors. One is the incoming indirect radiance $L^+(\mathbf{x}_j, \omega_j)$, which updates the scattered indirect radiance (noted as $\overline{L^+}(\mathbf{x}_i, \omega_i)$) at point $\mathbf{x}_i$. Another quantity we aggregate is the direct radiance $L^{\mathbf{o}}(\mathbf{x}_j, \omega_j)$, updating refined estimates of scattered direct radiance $\overline{L^{\mathbf{o}}}(\mathbf{x}_i, \omega_i)$. Combining those two aggregations gives an updated estimate of total scattered radiance:

$$\overline{L}(\mathbf{x}_i, \omega_i) = \overline{L^+}(\mathbf{x}_i, \omega_i) + \overline{L^\mathbf{o}}(\mathbf{x}_i, \omega_i) \tag{11}$$

This updated outgoing radiance at $\mathbf{x}_i$ corresponds to the incoming indirect radiance at the previous shading point on the path (if any). It is propagated along its outgoing continuation edge to update the indirect radiance $L^+(\mathbf{x}_{i-1}, \omega_{i-1})$ at the $(i-1)$-th shading point $\mathbf{x}_{i-1}$ on $\overline{\mathbf{z}}$. In the surface case, [Deng et al. 2021] assumes light transport in vacuum between surfaces, so $L^+(\mathbf{x}_{i-1}, \omega_{i-1}) = \overline{L}(\mathbf{x}_i, \omega_i)$. Unlike the simple copying operation used in the surface case, propagation for the volume path graph should consider the transmittance term and the probability of sampling distance $||\mathbf{x}_i - \mathbf{x}_{i-1}||$ from $\mathbf{x}_{i-1}$ in Eq. (10), where

$$L^+(\mathbf{x}_{i-1}, \omega_{i-1}) = \frac{\tau(\mathbf{x}_i, \mathbf{x}_{i-1})}{p_t(\mathbf{x}_i|\mathbf{x}_{i-1})}\left(\overline{L^+}(\mathbf{x}_i, \omega_i) + \overline{L^\mathbf{o}}(\mathbf{x}_i, \omega_i)\right). \tag{12}$$

3.6 Path Graph Operators

In path graph, we use an **aggregation operator** to gather incoming radiance from nearby shading points and recompute the outgoing radiance at the points. Then we use a **propagation operator** to propagate the updated quantity to previous shading points on their path. Finally after a few iterations of alternating aggregation and propagation, we propagate updated radiance to the pixels.

The outgoing radiance of a point is updated by a weighted average of the contribution from nearby shading points in the current cluster. More precisely, this is done by treating other points in the cluster as independent sampling techniques, and combining them by multiple importance sampling. Recall that multiple importance sampling with the balance heuristic [Veach and Guibas 1995] provides weights for combining m sampling strategies as follows:

$$w(z_i) = \frac{p_i(z_i)}{\sum_{l=1}^m p_l(z_i)}, \tag{13}$$

where $p_i(z)$ is the probability of sampling z with the i-th strategy. The estimate of $f(z)$ from m strategies is then

$$E[f(z)] \approx \sum_{j=1}^m w(z_j)\frac{f(z_j)}{p_j(z_j)}. \tag{14}$$

We apply MIS to aggregate the indirect radiance from Eq. (9) for point $\mathbf{x}$ in a cluster C of size K. Given the sampling strategy $\hat{\mathbf{x}_j}$ at point $\mathbf{x}_j$ with its sampled direction ω_j,

$$\overline{L^+}(\mathbf{x}, \omega) = \sum_{j=1}^K w(\hat{\mathbf{x}}, \omega_j)\frac{f_\rho(\mathbf{x},\ \omega,\ \omega_j)L^+(\mathbf{x}_j, \omega_j)}{p(\hat{\mathbf{x}}, \omega_j)}. \tag{15}$$

Applying Eq. (13) to weigh the contribution from direction ω_j in cluster C with K points, we get the following weight

$$w(\hat{\mathbf{x}}, \omega_j) = \frac{p(\hat{\mathbf{x}}, \omega_j)}{\sum_{l=1}^K p(\hat{\mathbf{x}}_l, \omega_j)}, \tag{16}$$

where $\{\mathbf{x}_l \in C\}$ and $p(\hat{\mathbf{x}}_l, \omega_j)$ is the probability of choosing ω_j at $\mathbf{x}_l$ using the sampling strategy $\hat{\mathbf{x}}_l$. We call $p_m(\omega_j) = \sum_{l=1}^K p(\hat{\mathbf{x}}_l, \omega_j)$ the marginal density of directional sample ω_j in cluster C. Combining Eq. (16) and Eq. (15) yields:

$$\overline{L^+}(\mathbf{x}, \omega) = \sum_{j=1}^K \frac{f_\rho(\mathbf{x}, \omega, \omega_j)}{p_m(\omega_j)}L^+(\mathbf{x}_j, \omega_j). \tag{17}$$

The above can be computed for each shading point in time linear in K, computing the marginal densities in a first pass and Eq. (17) in a second pass over the cluster.

We define $\overline{\mathbf{L}^+}$ as a vector that holds the scattered indirect radiance of all shading points V_x along all the outgoing continuation edges in E_C and $\mathbf{L}^+$ as a vector that holds incoming indirect radiance at all the shading points V_x from all the incoming continuation edges in E_C. Then Eq. (17) can be expressed in matrix form as follow

$$\overline{\mathbf{L}^+} = \mathbf{A}^+\mathbf{L}^+, \tag{18}$$

where $\mathbf{A}^+$ is a $|V_x| \times |V_x|$ matrix with its element on row r column c being

$$\mathbf{A}^+_{rc} = \begin{cases} \frac{f_\rho(\mathbf{x}_r, \omega_r, \omega_c)}{p_m(\omega_c)}, & \mathbf{x}_r, \mathbf{x}_c \in \text{the same cluster} \\ 0, & \mathbf{x}_r, \mathbf{x}_c \text{ in different clusters.} \end{cases} \tag{19}$$

Aggregation of the direct radiance is slightly different from the indirect aggregation. For each shading point $\mathbf{x}$ in the cluster, it has two directions sampled for the incoming direct radiance, one from emitters and one from the local phase function. Therefore, when combining the estimations from all the direct radiance samples in a cluster, we consider $2K$ samples from $K+1$ sampling strategies: K samples from emitter sampling (here we view all emitter sampling as one strategy), and K local phase function strategies for each sample. Applying this multi-sample MIS on the $2K$ direct light samples for Eq. (8) and cancelling some terms gives us

$$\overline{L^\mathbf{o}}(\mathbf{x}, \omega) = \frac{1}{K}\sum_{j=1}^K \frac{Kf_\rho(\mathbf{x}, \omega, \omega_j^e)L^\mathbf{o}(\mathbf{x}_j, \omega_j^e)}{\sum_{l=1}^K p_\rho(\hat{\mathbf{x}}_l, \omega_j^e) + Kp_e(\omega_j^e)} \tag{20}$$

$$+ \sum_{j=1}^K \frac{f_\rho(\mathbf{x}, \omega, \omega_j)L^\mathbf{o}(\mathbf{x}_j, \omega_j)}{\sum_{l=1}^K p_\rho(\hat{\mathbf{x}}_l, \omega_j) + Kp_e(\omega_j)}, \tag{21}$$

where ω^e stands for the directional sample taken from emitter sampling with a PDF $p_e(\omega^e)$, and $p_\rho(\hat{\mathbf{x}}_l, \omega)$ is the PDF of sampling ω from phase function at a shading point $\mathbf{x}_l$. Similar to the indirect radiance case, we define $p_m(\omega) = \sum_{l=1}^K p_\rho(\hat{\mathbf{x}}_l, \omega) + Kp_e(\omega_j)$ as the marginal density for direction ω in its cluster, which gives us

$$\overline{L^\mathbf{o}}(\mathbf{x}, \omega) = \sum_{j=1}^K \frac{f_\rho(\mathbf{x}, \omega, \omega_j^e)}{p_m(\omega_j^e)}L^\mathbf{o}(\mathbf{x}_j, \omega_j^e) + \sum_{j=1}^K \frac{f_\rho(\mathbf{x}, \omega, \omega_j)}{p_m(\omega_j)}L^\mathbf{o}(\mathbf{x}_j, \omega_j). \tag{22}$$

To turn Eq. (22) into a matrix form, we define $\mathbf{L^o}$ as a vector of length $|V_y| = 2|V_x|$, with interleaved entries being incoming direct radiance from the light edges on the path graph which are sampled from the emitter distribution (entries with even indices) or in proportion to the local phase function (entries with odd indices); then $\overline{\mathbf{L^o}}$ is a vector of size $|V_x|$ holding the scattered direct radiance on all the shading points along all continuation edges. Writing Eq. (22) in a matrix form, we have

$$\overline{\mathbf{L^o}} = \mathbf{A}^o\mathbf{L^o}, \tag{23}$$

where the direct aggregation matrix $\mathbf{A}^o$ has a size of $|V_x| \times |V_y|$; its entry on row r and column c is

$$\mathbf{A}^o_{rc} = \begin{cases} \frac{f_\rho(\mathbf{x}_r, \omega_r, \omega^e_{c/2})}{p_m(\omega^e_{c/2})}, & \mathbf{x}_r, \mathbf{x}_{c/2} \text{ in same cluster and c is even} \\ \frac{f_\rho(\mathbf{x}_r, \omega_r, \omega_{c/2})}{p_m(\omega_{c/2})}, & \mathbf{x}_r, \mathbf{x}_{c/2} \text{ in same cluster and c is odd} \\ 0, & \text{otherwise.} \end{cases} \tag{24}$$

Putting Eq. (23) and Eq. (18) into Eq. (11) we get

$$\overline{\mathbf{L}} = \overline{\mathbf{L}^+} + \overline{\mathbf{L}^\mathbf{o}} = \mathbf{A}^+\mathbf{L}^+ + \mathbf{A}^o\mathbf{L}^\mathbf{o}. \tag{25}$$

Eq. (25) is the aggregation operation; it gives us the equations for recomputing the outgoing radiance at each shading point from the recorded quantities on the path graph.

Once the outgoing radiance $\overline{\mathbf{L}}$ of a shading point is refined, we need to propagate this refinement towards the previous shading point through the continuations edges. This propagation operation is described in Eq. (12), and we also write it in matrix multiplication form as

$$\mathbf{L}^+ = \mathbf{P}\overline{\mathbf{L}} \tag{26}$$

with a propagation matrix $\mathbf{P}$ of size $|V_x|^2$ whose entry at row r and column c is

$$\mathbf{P}_{rc} = \begin{cases} \frac{\tau(\mathbf{x}_r, \mathbf{x}_c)}{p_t(\mathbf{x}_r|\mathbf{x}_c)}, & \mathbf{x}_r \text{is the previous shading point of } \mathbf{x}_c \\ 0.0, & \text{otherwise.} \end{cases} \tag{27}$$

Expanding Eq. (26) by Eq. (25) yields

$$\mathbf{L}^+ = \mathbf{P}\mathbf{A}^+\mathbf{L}^+ + \mathbf{P}\mathbf{A}^o\mathbf{L}^\mathbf{o}, \tag{28}$$

Now that we have introduced the aggregation operators $(\mathbf{A}^o, \mathbf{A}^+)$ and the propagation operator $(\mathbf{P})$ over the path graph, given a path graph $G = \langle V, E\rangle$ and the recorded variables on the path graph, we are solving for a fixed point $\mathbf{L}^+$ for the updating rule in Eq. (28), and each image pixel value can be reconstructed by selecting the entries of $\mathbf{L}^+$ that tie to that pixel. Since the direct illumination vector $\mathbf{P}\mathbf{A}^o L^\mathbf{o}$ is independent of the variable $\mathbf{L}^+$ that is being updated, it is computed once and remains constant throughout the iteration. With $\mathbf{L}^+$ initialized to the indirect radiance estimates from path tracing, the linear system converges within a few iterations.

4 RENDERING SYSTEM

We implemented an instrumented volumetric path tracer for path graph recording based on the open source renderer Mitsuba v0.6 [Jakob 2013] and added our volumetric path graph solver as a CUDA plugin.

Instrumented path tracer. Path tracing is performed on the CPU. During path tracing, we record data for the emitter sampling and phase function sampling at each shading point. Additionally, the incoming indirect radiance for each shading point is recorded. Once a path is traced, all of its shading points are stored into a global array $\mathbf{x}$ that preserves their order on the path. $\mathbf{x}$ is stored in pinned memory to minimize data transfer time onto the GPU.

Path graph solver. Once the shading points are on the GPU, m cluster centers are randomly selected such that $mK = n$ where K is the expected cluster size and n is the total number of shading points. A hash-grid on the shading points' positions is built, and a nearest-neighbor clustering is performed by searching through neighboring bins in the hash grid. After forming the clusters, we use data from the recorded light points to perform one pass of direct radiance aggregation following Eq. (22). We also use the recorded incoming indirect radiance for each shading point to compute the aggregated indirect radiance by following Eq. (17). The sum of aggregated direct and indirect light is then propagated across each continuation edge. We then perform a few iterations of alternating indirect radiance aggregation and propagation to get to the final result. Note that we could have initialized a zero-valued incoming indirect radiance at each shading point, but it would take q iterations of aggregation and propagation to correctly compute the radiance for a path of length q; that is not ideal for volumetric rendering, which can have especially long paths. Starting from path-tracing estimates of indirect incoming radiance puts the image at the correct energy level from the first iteration, and the iterative refinements are only used to improve the image smoothness. It normally takes a small number of iterations (less than 10) to obtain a converged output, even if many paths are much longer than 10 bounces.

Handling direct Light. Direct radiance aggregation is only performed to better optimize indirect radiance (through propagation). In order to remove significant correlations between neighboring points, we use the non-aggregated direct radiance from path tracing instead of the aggregated direct radiance when writing the final output. This means that our method can be expected mainly to improve on scenes dominated by indirect radiance. However, for certain scenes that are dominated by participating media but have a few direct radiance components (e.g. emitters in a medium or media encapsulated by dielectric surfaces), Path Graph can still show a benefit, but only if the direct radiance is handled well. To this end we optionally record a few extra emitter samples for the first bounce on each path. In scenes where direct radiance needs more improvement, we apply Path Graph as normal by using the aggregated direct radiance in propagation, but then use the extra direct light samples, instead of the non-aggregated direct radiance, to compute the final pixel values.

Final output. The radiance on the continuation edge for the first shading point of each path represents the radiance propagated into the camera. These radiance values will be the output for the Path Graph after a few iterations of aggregation and propagation. The output values for each path are written into the corresponding pixels of the output image, which can then be optionally fed into a denoiser for the final output.

5 EXPERIMENTS

We study the behavior of path graph on a variety of volumetric scattering scenes under challenging lighting conditions. Our tests contain both homogeneous (Buddha, Traffic Light, Foggy Forest) and heterogeneous volumes (Bunny Cloud, Disney Cloud, Dust Shockwave, Colored Smoke, Industry Smoke, and Golden

Table 1. Detailed timing of our pipeline steps. The 2nd to 5th columns are the average render time of 1 sample per pixel for both path tracing, revised path tracing with data recording, recording of extra direct light samples (only applicable to BUDDHA and TRAFFIC LIGHT), and the total execution time of our path graph iteration. The ratio is the number of samples that a path tracer could compute during the same time of 1spp using our method, with path sample recording overhead. The number of PT/PG samples used in 5 min equal time comparison is computed from the time for a single sample run, with an additional note that if extra direct light samples are recorded, this operation will only be performed once and thus its time would not multiply with the number of Path Graph samples.

Scene Name	time (s)				ratio	MSE for 1SPP		pathtr:ours SPP for equal time(300s)	Equal time (300s) MSE ratio path tr: ours
	path tr	ours				path tr	ours		
		record path tr	record extra direct samples	path graph					
BUDDHA	18.9	24.2	2.8 (15spp)	16.92	2.32	0.0184	0.000837	15:7	1.24e-3 : 2.68e-4 = 4.62
TRAFFIC-LIGHT	4.3	6.0	31.1 (69spp)	9.426	10.82	0.0263	0.000999	69:17	8.98e-4 : 2.12e-4 = 4.24
FOGGY-FOREST	6.8	10.4	N/A	13.74	3.55	0.0708	0.00515	44:12	1.61e-3 : 4.34e-4 = 3.70
DISNEY-CLOUD	3.0	15.8	N/A	5.148	6.98	0.0786	0.00391	100:14	7.87e-4 : 3.07e-4 = 2.57
INDUSTRY-SMOKE	12.5	13.1	N/A	1.826	1.25	0.0432	0.00721	24:19	1.80e-3 : 3.84e-4 = 4.69
MITSUBA-COLORED-SMOKE	6.2	6.9	N/A	6.882	2.22	0.212	0.0238	48:21	4.31e-3 : 3.95e-4 = 10.91
GOLDEN-GATE	43.8	50.4	N/A	17.36	1.55	0.128	0.0135	7:4	1.85e-2 : 3.48e-3 = 5.32

GATE). The resolution of all renderings is 1440x960 pixels. The CPU we use is an Intel Xeon Silver 4214 with 8 cores, and the GPU is an Nvidia RTX 3090.

Equal sample comparison. In Fig. 4 we show an 1 spp comparison between our method and path tracing on scenes with heterogeneous media. Our method is much closer to the reference and produces a much smaller MSE than path tracing.

Equal time comparison. Our method necessarily has some overhead per sample. Therefore, we also render some more challenging scenes using both path tracing and our method in 5 minutes. In all tested cases, our method still visually gives a smoother result than path tracing and numerically has a smaller MSE, as we can see in Fig. 5 and Fig. 6. The detailed breakdown of path tracing and our method's time and MSE is shown in Table 1. Our method produces even more variance reduction in scenes that have more indirect light (e.g. COLORED SMOKE) and/or media with a high density and high albedo (e.g. INDUSTRY SMOKE).

Note that despite being dominated by indirect light, BUDDHA and TRAFFIC LIGHT both have some noticeable direct light components (specular reflections on the dielectric surface in BUDDHA and the emitters in TRAFFIC LIGHT), and as mentioned in Sec. 4, here we used additional direct light samples to render the final result. Again these 1-bounce samples are not used in the path graph itself but only when writing the final output to make up for the path graph's non-aggregated direct radiance. We make it so that path graph has an equal number of direct illumination samples as path tracing, and thus any variance reduction comes from path graph itself. The time for the additional samples is accounted for in the equal time comparison. Since the Path Graph already handles the more difficult indirect illumination, the direct illumination samples do not take long to render and won't significantly reduce the number of path graph samples that could be run. We can see that path graph still gives more desirable results in both scenes.

Photon mapping and BDPT. Photon mapping and Bidirectional Path Tracing (BDPT) are two well-known methods for rendering homogeneous media. We include them in our equal time comparison shown in Fig. 5. These two methods show certain strengths in some types of scenes but both have significant limitations. Photon mapping gives a reasonable result in a small enclosed medium (BUDDHA), but it struggles particularly in outdoor scenes (TRAFFIC LIGHT and FOGGY FOREST) where targeting the photons into the camera view is non-trivial. BDPT, though theoretically unbiased, converges slowly and thus has a larger MSE when rendered in a small amount of time like here. Our method adapts well to the various lighting conditions and has barely noticeable bias.

Log-log error curves. In Fig. 6, we further show a log-log convergence curve up to 2000 iterations of path tracing and our method. Due to the bias we introduce during aggregation, the curve for our method will eventually intersect with path tracing's curve when it is more preferable to use path tracing. Nevertheless, this will happen well beyond 2000 spp, showing that our method can improve convergence across a reasonable range of spp settings.

Study of Path Graph iterations. As mentioned in Sec. 4, it often takes less than 10 iterations of aggregation and propagation for Path Graph to converge. To support our claim, we show in Fig. 7 how Path Graph's output changes as the number of iterations varies. It can be seen that even though the two scenes differ in the amount of direct illumination (the bunny scene has more indirect light and thus takes a few more iterations to converge), they both converge within 10 iterations.

6 CONCLUSION

In this work, we extended the path graph framework to scenes involving volumetric scattering media like clouds and fog. Our method focuses on more effective reuse of multiple-scattering paths and offers a notable improvement in sampling efficiency and speed over traditional path tracing techniques, particularly in challenging environments with heterogeneous, highly scattering media and complex lighting. While our approach shows significant potential in enhancing volumetric rendering, it represents just one step in addressing the broader challenges of the field. Future explorations could include deeper integration of volume and surface path graphs.

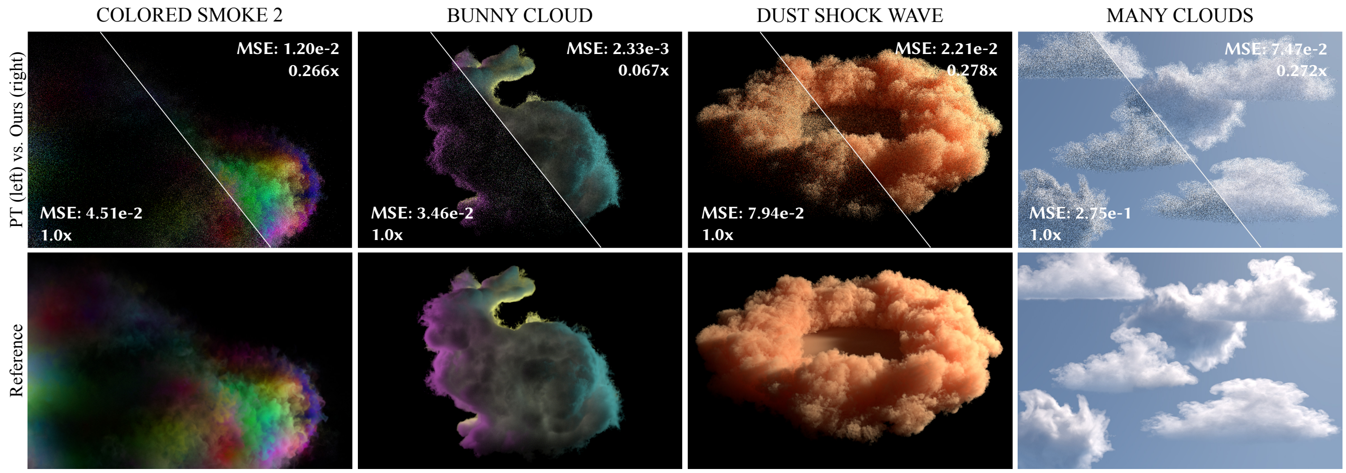

Fig. 4. We show an **equal sample (at 1 sample per pixel) comparison** between ours and path tracing. Our method extensively reuses the paths' information even with only one sample per pixel, significantly improving rendering efficiency.

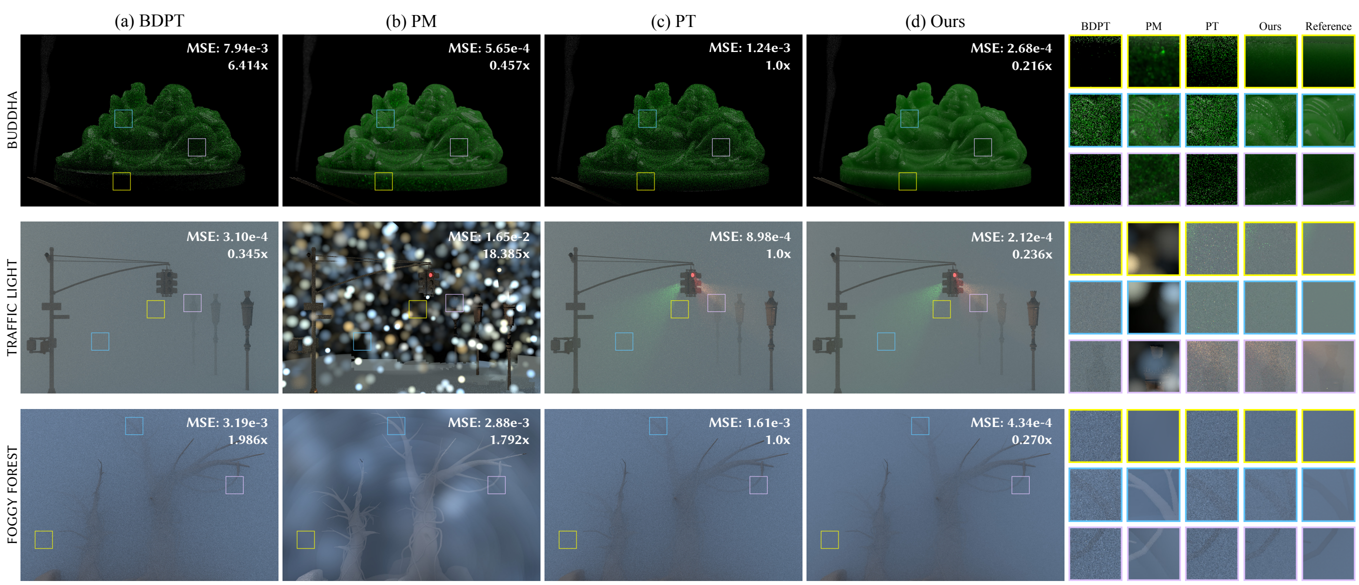

Fig. 5. Comparison between (a) bidirectional path tracing, (b) photon mapping, (c) path tracing and (d) our method (path tracing + path graphs) on an **equal time (5 min)** in scenes with the presence of homogeneous volume. We show full light transport in all the scenes and our method provides significant variance reduction over previous methods on rendering the participating media. The 2nd and 3rd scenes contain large volume of outdoor medium (e.g. fog), where (unguided) photon mapping suffers since only a small amount of photons arrive at the region where the camera is looking towards due to strong multiple scattering. The 2nd scene has a skydom as environment lighting as well as the traffic lights, where the bidirectional method failed to sample the small light sources.

REFERENCES

James Arvo. 1993. Linear Operators and Integral Equations in Global Illumination. In *Global Illumination*. Vol. 42. Chapter 2, 1–21.

Benedikt Bitterli and Wojciech Jarosz. 2017. Beyond Points and Beams: Higher-Dimensional Photon Samples for Volumetric Light Transport. 36, 4 (July 2017), 1–12. https://doi.org/10/gfznbr

Benedikt Bitterli, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. 2020. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 39, 4 (July 2020). https://doi.org/10/gg8xc7

Xi Deng, Miloš Hašan, Nathan Carr, Zexiang Xu, and Steve Marschner. 2021. Path graphs: iterative path space filtering. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 40, 6 (dec 2021). https://doi.org/10.1145/3478513.3480547

Xi Deng, Shaojie Jiao, Benedikt Bitterli, and Wojciech Jarosz. 2019. Photon surfaces for robust, unbiased volumetric density estimation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 38, 4 (jul 2019). https://doi.org/10.1145/3306346.3323041

Iliyan Georgiev, Jaroslav Křivánek, Tomáš Davidovič, and Philipp Slusallek. 2012a. Light Transport Simulation with Vertex Connection and Merging. 31, 6 (Nov. 2012), 192:1–192:10. https://doi.org/10/gbb6q7

Iliyan Georgiev, Jaroslav Křivánek, Stefan Popov, and Philipp Slusallek. 2012b. Importance Caching for Complex Illumination. 31, 2 (June 2012), 701–710. https:

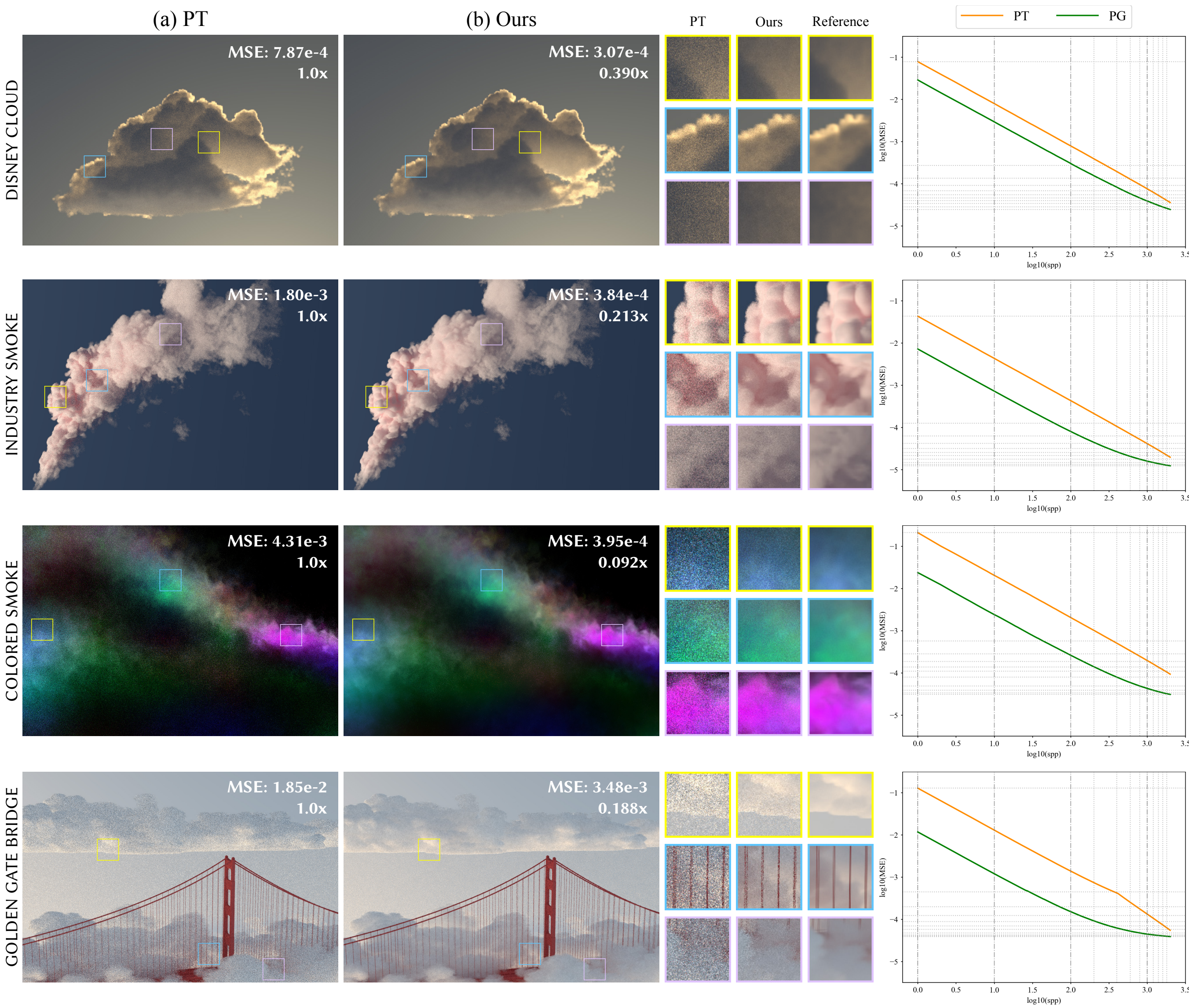

Fig. 6. Comparison between (a) path tracing and (b) our method (path tracing + path graphs) on an **equal time (5 min)** in scenes with the presence of heterogeneous volume. We show full light transport in all the image and our method out perform the path tracing especially around the area where the pixel value is dominated by multi-bounces.

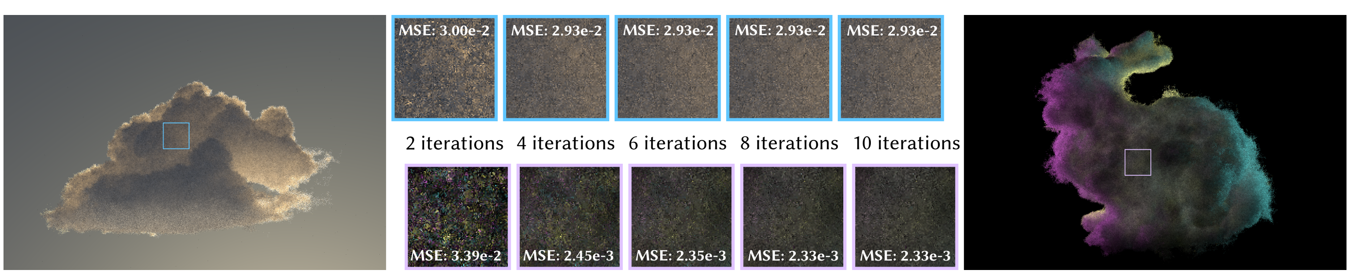

Fig. 7. Two 1spp scenes with a varying number of Path Graph iterations. Path Graph converges well within 10 iterations in both cases.

//doi.org/10/gbbdcc
Toshiya Hachisuka, Jacopo Pantaleoni, and Henrik Wann Jensen. 2012. A Path Space Extension for Robust Light Transport Simulation. 31, 6 (Jan. 2012), 191:1–191:10. https://doi.org/10/gbb6n3
Sebastian Herholz, Yangyang Zhao, Oskar Elek, Derek Nowrouzezahrai, Hendrik P. A. Lensch, and Jaroslav Křivánek. 2019. Volume Path Guiding Based on Zero-Variance Random Walk Theory. *ACM Trans. Graph.* 38, 3, Article 25 (jun 2019), 19 pages. https://doi.org/10.1145/3230635
Jinkai Hu, Chengzhong Yu, Hongli Liu, Lingqi Yan, Yiqian Wu, and Xiaogang Jin. 2023. Deep Real-Time Volumetric Rendering Using Multi-Feature Fusion. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) *(SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, Article 61, 10 pages. https://doi.org/10.1145/3588432.3591493
Wenzel Jakob. 2013. Mitsuba Renderer. http://www.mitsuba-renderer.org.
Wojciech Jarosz, Matthias Zwicker, and Henrik Wann Jensen. 2008a. *The Beam Radiance Estimate for Volumetric Photon Mapping*. Technical Report CS2008-0914. University of California, San Diego.
Wojciech Jarosz, Matthias Zwicker, and Henrik Wann Jensen. 2008b. The Beam Radiance Estimate for Volumetric Photon Mapping. 27, 2 (April 2008), 557–566. https://doi.org/10/bjsfsx
Henrik Wann Jensen. 1996. *The Photon Map in Global Illumination*. Ph.D. Thesis. Technical University of Denmark.
Henrik Wann Jensen and Niels Jørgen Christensen. 1995. Photon Maps in Bidirectional Monte Carlo Ray Tracing of Complex Objects. 19, 2 (March 1995), 215–224. https://doi.org/10/d9xr6q
Henrik Wann Jensen and Per H. Christensen. 1998. Efficient Simulation of Light Transport in Scenes with Participating Media Using Photon Maps, Vol. 32. ACM, 311–320. https://doi.org/10/b64p36
James T. Kajiya and Timothy L. Kay. 1989. Rendering Fur with Three Dimensional Textures. 23, 3 (July 1989), 271–280.
Alexander Keller, Ken Dahm, and Nikolaus Binder. 2014. Path Space Filtering *(SIGGRAPH '14)*. ACM, Article 68, 1 pages. https://doi.org/10/gfz6mr
Jaroslav Křivánek, Pascal Gautron, Greg Ward, Henrik Wann Jensen, Eric Tabellion, and Per Christensen. 2008. Practical Global Illumination with Irradiance Caching. https://doi.org/10/bhnsjz
Jaroslav Křivánek, Iliyan Georgiev, Toshiya Hachisuka, Petr Vévoda, Martin Šik, Derek Nowrouzezahrai, and Wojciech Jarosz. 2014. Unifying Points, Beams, and Paths in Volumetric Light Transport Simulation. 33, 4 (July 2014), 103:1–103:13. https://doi.org/10/f6cz72
Eric P. Lafortune and Yves D. Willems. 1993. Bi-Directional Path Tracing, H. P. Santo (Ed.), Vol. 93. Alvor, Portugal, 145–153.
Daqi Lin, Chris Wyman, and Cem Yuksel. 2021. Fast Volume Rendering with Spatiotemporal Reservoir Resampling. *ACM Trans. Graph.* 40, 6, Article 279 (dec 2021), 18 pages. https://doi.org/10.1145/3478513.3480499
Eric Veach. 1997. *Robust Monte Carlo Methods for Light Transport Simulation*. Ph.D. Thesis. Stanford University, United States – California.
Eric Veach and Leonidas J. Guibas. 1994. Bidirectional Estimators for Light Transport, Georgios Sakas, Peter Shirley, and Stefan Müller (Eds.). 145–167. https://doi.org/10/gfznbh
Eric Veach and Leonidas J. Guibas. 1995. Optimally Combining Sampling Techniques for Monte Carlo Rendering, Vol. 29. 419–428. https://doi.org/10/d7b6n4
Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear. 1988. A Ray Tracing Solution for Diffuse Interreflection. 22, 4 (Aug. 1988), 85–92. https://doi.org/10/dk6rt5
Rex West, Iliyan Georgiev, Adrien Gruson, and Toshiya Hachisuka. 2020. Continuous Multiple Importance Sampling. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 39, 4, Article 136 (July 2020), 12 pages. https://doi.org/10.1145/3386569.3392436